A nove l f u z z y l o g i c b a s e d r ever s e e n g i n e e r i n g o f g e n e r e g u l a t o r y n e t w o r k

Raviajot Kaur
Department of Computer Science and Engineering, PEC University of Technology, Chandigarh 160012, India, raviajot25@gmail.com

Abhishek Bhola
Department of Computer Science and Engineering, PEC University of Technology, Chandigarh 160012, India, abhishek_bhola@hotmail.com

Shailendra Singh
Department of Computer Science and Engineering, PEC University of Technology, Chandigarh 160012, India, shailendra_sing@yahoo.com

Follow this and additional works at: https://digitalcommons.aaru.edu.jo/fcij

Part of the Computational Engineering Commons

Recommended Citation

A novel fuzzy logic based reverse engineering of gene regulatory network

Raviajot Kaur, Abhishek Bhola, Shailendra Singh*

Department of Computer Science and Engineering, PEC University of Technology, Chandigarh 160012, India

Abstract

Genes of an organism play a very crucial role in the working of various cellular activities. Genes and other biological molecules like DNA, RNA do not operate alone but they all are correlated. Their relationships are shown with the help of networks commonly known as Gene Regulatory Networks. Gene Regulatory Networks are complex control networks that show the map of interactions among the genes. They provide very useful contribution to the genomic science and increase the understanding of various biological processes. In this paper, fuzzy logic based method is proposed for the reverse engineering of gene regulatory network from microarray gene expression datasets. Pre-processing steps have been introduced to increase the efficiency of the method. Clustering technique is also employed to divide the problem into sub problems to reduce the computational complexity at some extent. Finally, the proposed method is tested on two different time course gene expression datasets of yeast having GEO accession number GDS37 and GDS3030. The results are validated by using Specificity, Sensitivity and F-score as parameters. Results of the proposed method are further compared with other existing method which was proposed by Al-Shobaili in 2014.

© 2017 Faculty of Computers and Information Technology, Future University in Egypt. Production and hosting by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Keywords: Fuzzy logic; Gene regulatory networks; Microarray gene expression datasets; Gene expression levels; Clustering; Filtering

1. Introduction

With availability of advanced technologies like microarray technology, large amount of information related to expression levels of genes of different living beings are available easily. In the last few years, research in the field of reverse engineering of gene regulatory network has received the growing interest due to availability of gene expression data of different living beings. Microarray is the technology which is used to monitor and calculate gene expression levels of thousands of genes simultaneously in single experiment [12]. Gene expression level indicates the magnitude of a gene for defined sample at a certain point of time in changing environment conditions [9]. The gene expression data act as input for reverse engineering of GRN. But microarray gene expression datasets prepared from microarray technology are subjected to noise and experimental errors. This makes the understanding of dynamics of GRNs using correct and representative methods/models a difficult task [18,3].

The large number of biological processes like cell reproduction, metabolism, etc that take place in the living organisms at the cellular level are controlled by the regulation of gene expressions [5]. Gene is the coding region of the DNA that helps in the formation of proteins to regulate the biological activities. The formed proteins further control the rate of formation of proteins by some other genes. Some genes enhance the formation of proteins process that are known as activators and some genes which slows down that process are known as inhibitors. Hence, genes regulate one another and affect each other directly or indirectly which leads to the formation of Gene Regulatory Network (GRN) [4,1,2]. It can also be represented through directed or undirected graphs.

* Corresponding author.
E-mail addresses: raviajot25@gmail.com (R. Kaur), abhishek_bhola@hotmail.com (A. Bhola), shailendra_sing@yahoo.com (S. Singh).

The reverse engineering of GRNs from gene expression data provide many useful applications in different areas, for example, drug discovery, impact of drug on the individual, help in the tracking of development of cancer in the cells, and many more [32].

However, there exist many challenges which make the reverse engineering of GRNs computationally complex. The major problems are: curse of dimensionality (i.e. number of samples are very few as compared to number of genes in the microarray datasets), and incomplete data [19]. The increasing number of genes leads to problem of high time complexity for the methods. As the number of genes increases, the time complexity of the method also increases exponentially.

Several methods have been proposed in the literature for the reverse engineering of GRNs like Boolean Networks [26,34], Probabilistic Boolean Networks (PBN) [30], Bayesian Networks [16,8,6], Dynamic Bayesian Networks (DBN) [16,21,29], Artificial Neural Network based models [25,20,17] and Fuzzy Logic based models [22].

The main motivation for using the fuzzy logic lies in the fact that biological networks are fuzzy in nature and therefore fuzzy logic is an acceptable technique for reverse engineering of GRN. The most basic fuzzy based technique was introduced by Woolf and Wang (Bordon et al. [7]) to find out the relationships between network triplets i.e. repressors, activators and target genes using the *Saccharomyces cerevisiae* (yeast) dataset. The major drawback of this technique is the time complexity. The different fuzzy based methods which are present in literature and perform effectively at some extend are: Quantitative fuzzy logic modelling approach [33], Collateral-Fuzzy Gene Regulatory Network Reconstruction (CF-GeNe) [28], Exhaustive Search Fuzzy based Technique [31] and Fuzzy Model for predicting change in expression levels [23].

The paper is organised as follows: Section 2 explains the background which includes the introduction to fuzzy logic and the traditional fuzzy logic based method for reverse engineering of GRN. Section 3 explains step by step methodology of the proposed method for the reverse engineering of GRN. Section 4 gives detail about dataset employed and their results. Section 5 provides the conclusion.

2. Background

2.1. Fuzzy logic

Fuzzy logic is a fundamental approach of computing which is based on the degrees of truth rather than crisp values i.e. true or false (0 or 1). It uses 0 and 1 for the extreme cases. Mamdani and Tagaki-Sugeno are two well-known fuzzy logic inference techniques. These models work on natural language based if-then-else fuzzy rules. Fuzzy Logic is very simple and easy approach to solve the complex problems with accuracy. Fuzzy logic has various unique features like it is very robust as it does not require exact inputs, it can be modified easily in order to improve the performance, and it can produce smooth output with wide range of inputs [15,10,11].

2.2. Traditional fuzzy logic based method

Woolf and Wang used fuzzy logic to depict the interactions between the genes of yeast data [24,33]. This is one of the basic techniques to find out the relationships between the repressors, activators and target genes. The expression level values of genes are fuzzified according to different qualifiers like High, Low, Medium. The fuzzy rules used in this technique are described in the decision matrix as shown in Fig. 1.

According to these rules, target value is estimated. Finally, the ranking is done on the basis of error between actual target value and estimated target value. Genes with the low error are scored higher i.e. have higher ranks [13].

3. Proposed method

Step 1− pre-processing

High dimensional microarray gene expression datasets comprise of thousands of genes out of which some genes do not show any interesting changes during experimentation. Therefore, these genes can be removed as it will be helpful in effective analysis of gene expression dataset. Fig. 2 represents the filtering steps used in the proposed method for microarray gene expression datasets.

First, the genes are identified with missing values and then indexing commands are used to remove the genes. The reasons for missing values in gene expression data are due to scratches and dust on the microarray glass slide while monitoring gene

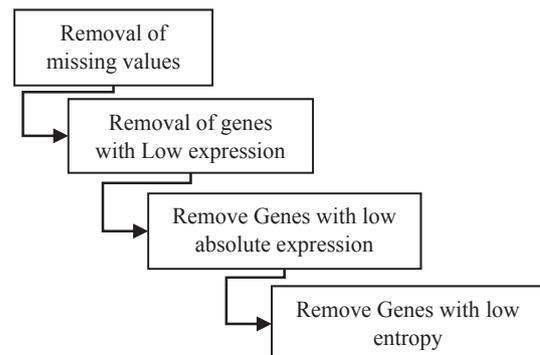Fig. 1. Decision matrix for Woolf and Wang model.

Fig. 2. Filtering steps for microarray gene expression dataset.

expressions. If genes containing missing values are not removed then it will lead us to wrong interpretation. Secondly, the gene expression levels where values differ in very less amount i.e. they have very low variance and flat in nature are filtered out as they do not produce any interesting results. The main aim to introduce this pre-processing step is to ensure that a wide enough dynamic range of measurements exists in different genes profiles. The formulae used to calculate variance of the gene profiles is given below:

$$\text{Variance} = \frac{1}{N-1} \sum_{i=1}^{N} |G_i - \mu|^2 \quad \text{where } \mu = \frac{1}{N} \sum_{i=1}^{N} G_i$$

where, N is total number of genes, G_i is the gene expression for gene i and μ represents the mean. Then, the gene with low absolute expressions levels are filtered out. The low absolute expressions of genes are due to poor spot hybridization and large quantization errors in microarray experiment. The low absolute value filter is implemented as it is believed that gene profiles having low gene expression are less reliable than gene profiles having high gene expression. Lastly the genes having low entropy are removed. The effectiveness of the genes is calculated by using entropy filter method. Low entropy means less effective genes. The motive to remove low entropy genes is to mitigate the spiking behaviour shown by some gene profiles present in a particular gene expression dataset. The formulae used to calculate entropy of the gene profile is given below:

$$\text{Entropy } H(g) = -\sum_{i} p(g_i) \log_2(p(g_i))$$

where p() is the probability function and i stands for number of samples/time point of gene expression dataset.

Step 2 — clustering

After the pre-processing of microarray gene expression dataset, the clustering is performed to divide the problem into sub problems. The two different clustering techniques: hierarchical clustering and k-means clustering are implemented on the input microarray gene expression dataset in order to select the most efficient technique.

K-means clustering outperforms the hierarchical clustering as the performance of hierarchical clustering goes on decreasing as the number of genes increases and it leads to increase in the execution time of the proposed method. The other benefit of using K-mean clustering on top of Hierarchical clustering is that the genes are equally distributed in the clusters produced by k-means clustering while in hierarchical clustering the genes are mostly concentrated in two-three clusters.

K-means clustering produce better results for high dimensional microarray gene expression datasets on the other hand hierarchical clustering produces good results for small datasets. Therefore, k-means clustering is further used in the proposed method for the reverse engineering of GRN.

In this work, clusters are generated using k-means clustering technique. First, the networks of cluster centroids (CCs)

are generated. The cluster centres network is then examined to check which cluster centre fits the method well. Those clusters whose cluster centres do not fit in the network well are discarded and remaining clusters are further considered for examination. This step is helpful in decreasing the computational time of the proposed method to some extent and it also decreases the complexity of network [14].

Step 3 — fuzzy inference system for proposed method

Further, the genes present in the selected clusters are normalized on the scale of 0—1 using the Min-Max technique. Then the normalized values of input gene expression levels are fuzzified using qualifiers High, Medium and Low based on the following fuzzy membership functions as shown in Fig. 3.

The output of proposed fuzzy inference system is classified into five qualifiers {Medium Increase (MI), High Increase (HI), Insignificant (I), Medium Decrease (MD), High Decrease (HD)}. After fuzzification of expression level of genes, genes are further classified into Activator (A), Repressor (R) and Target (T) genes. The target gene is estimated for each pair of activator and repressor for every sample present in the microarray gene expression dataset using the fuzzy rules defined in Fuzzy Decision Matrix as shown in Fig. 4.

Step 4 — ranking of triplets

After successful completion of fuzzification, the target value is defuzzified using the centroid method to get the final target value. After the defuzzification, the Estimated Target Value (ETV) and Actual Target Value (ATV) are compared with each other to calculate the error and the variance. The
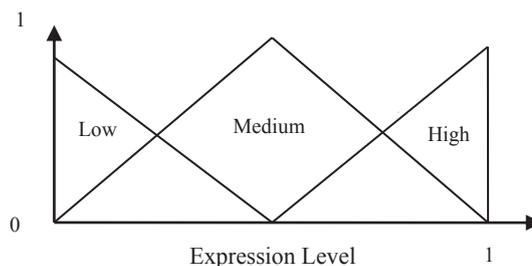

Fig. 3. Fuzzy input membership functions.

Fig. 4. Decision matrix for proposed method.

mathematical formulation for the calculation of Mean Squared Error (MSE) gene is as follows:

$$MSE(X, z) = \frac{1}{N \sum\limits_{i=1}^{N} (y_i - z_i)^2} = \frac{1}{N \sum\limits_{i=1}^{N} (F(X_i) - z_i)^2}$$

where, X is the input matrix containing $x_1, \ldots x_k$ genes and k is the total number of genes. N is the total number of samples in a gene. y is the output target gene i.e. the estimated value of target, $y = F(X)$ and z is the actual target value.

Variance (V) depends upon the number of rules fired for a triplet to find the value of target gene. If all fuzzy rules are fired equally, then the variance is low. Then, Residual Score ($R.S$) is calculated on the basis of V and *MSE*.

$R.S = V*MSE$

To find out the map of interactions in genes, triplets are ranked. Firstly, the threshold limit (α) for residual score is defined to 1%. The values corresponding to those triplets lies below the threshold limit are further considered for ranking. The threshold limit reduces the time complexity to some extent. Selected gene triplets are ranked on the basis of residual score that means low variance and low error means higher rank.

Resulting triplets are added to resultant matrix (M_r) and on the basis of M_r, candidate Gene Regulatory Network (cGRN) i.e. sub-network are generated. The above process is repeated for each cluster and all cGRN are obtained. Then for every cGRN, the gene representative of that network i.e. gene head of clusters is found. Finally, all the cGRNs using their gene representative are merged. A complete network of genes is formed as the output. Fig. 5 shows complete block diagram of the proposed method.

The Pseudo Code for carrying out the reverse engineering of GRN for each cluster is shown below:

For each *Cluster$_i$*
Formation of triplets, *ART*
For each triplet *ART$_j$*
ATV$_j$ = Gene expression of target gene, *T$_j$*
Calculate new target value based on FIS, *ETV$_j$*
MSE$_j$ = *ATV$_j$* − *ETV$_j$*
V$_j$ = Number of Fuzzy rules fired
R.S.$_j$ = *MSE$_j$***V$_j$*
If   *R.S.$_j$* < α
M$_r$ ← *ART$_j$*
cGRN$_j$ ← *M$_r$*
Merge all *cGRN*
Formation of Complete GRN

4. Experimental tests and results

4.1. Dataset

The time course gene expression datasets of yeast (*S. cerevisiae*) are taken into consideration for the validation of the proposed method. The first dataset of yeast is taken during its
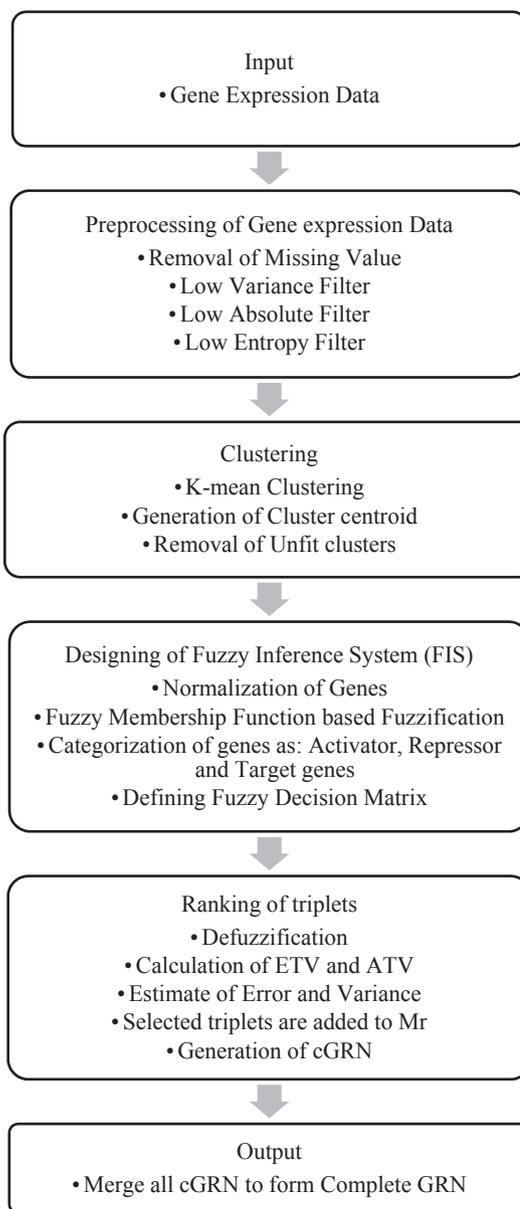

Fig. 5. Block diagram of proposed method.

diamide treatment (GEO Accession Number GDS37) and another is taken while it is shifting from anaerobic to aerobic growth (GEO Accession Number GDS3030). The total number of genes of yeast data present at the time of diamide treatment are 6385 at 7 different experimental conditions while during shifting from anaerobic to aerobic growth are 6307 genes at 6 different experimental condition. The proposed method is validated by employing FunCoup 3.0: a database of genome-wide functional coupling networks [27].

The implementation of the proposed method is done in MATLAB R2015a. Different genes are selected randomly from the both datasets using a random function for the reverse engineering of GRN. The randomly selected genes from GEO Accession Number GDS37 using random function are as: YGR066C, YGR201C, PEX14, DBP1, UGX2, UIP4, NQM1, UPS2, YFL054C, GPX1 and SNA3. The genes selected from
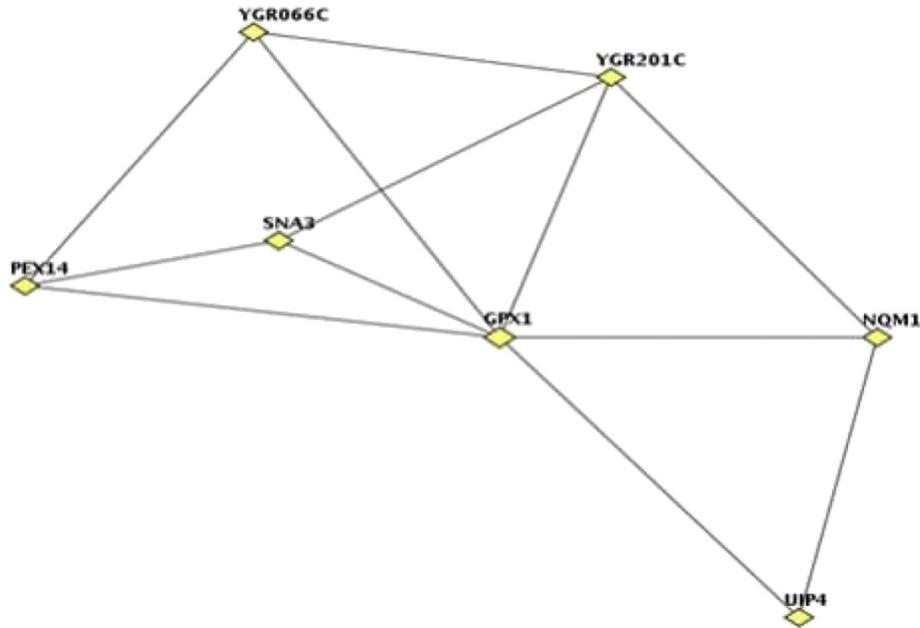
Fig. 6. GRN generated by FunCoup 3.0 for Dataset with GEO Accession Number: GDS37.
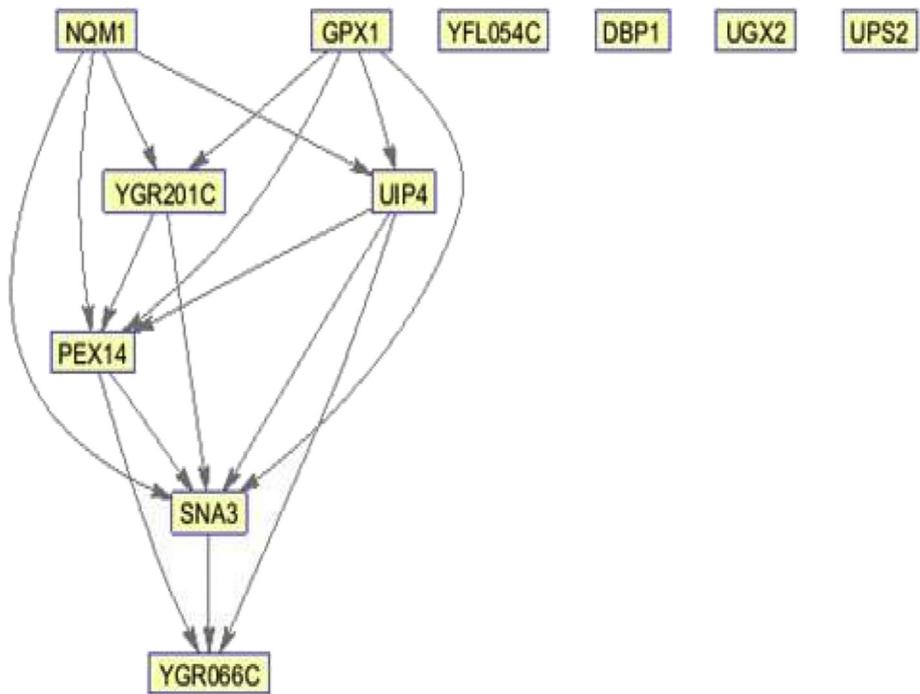

Fig. 7. GRN generated by Proposed Method for Dataset with GEO Accession Number: GDS37.

other dataset i.e. GDS3030 are: YLR108C, ARA1, ADH5, TMT1, ARG3, ARG56, CPA2, ARG4, RTC2, ICY2, INA1.

4.2. Results

To measure the performance of proposed method, Sensitivity, Specificity and F-Score are the measures that are taken into consideration. These parameters are defined as follows:

$$Sensitivity\ (S_n) = \frac{Correctly\ predicted\ edges}{All\ edges\ present\ in\ real\ life\ network}$$

$$Specificity\ (S_p) = \frac{Correctly\ predicted\ edges}{All\ edges\ present\ in\ the\ proposed\ method}$$

$$F - Score = \frac{2*S_p*S_n}{S_p + S_n}$$

Table 1
Performance measures of dataset GDS37.

Number of genes	Specificity	Sensitivity	F-score
7	0.55	0.85	0.67
8	0.55	0.85	0.67
9	0.5	0.75	0.60
10	0.6	0.67	0.63
11	0.56	0.75	0.64

The GRN generated by FunCoup 3.0 for microarray gene expression dataset with GEO accession number GDS37 is shown in Fig. 6 while the network generated by the proposed method for the same dataset is shown in Fig. 7.

Table 1 shows the performance measures for different number of genes in microarray gene expression dataset with GEO Accession Number: GDS37.
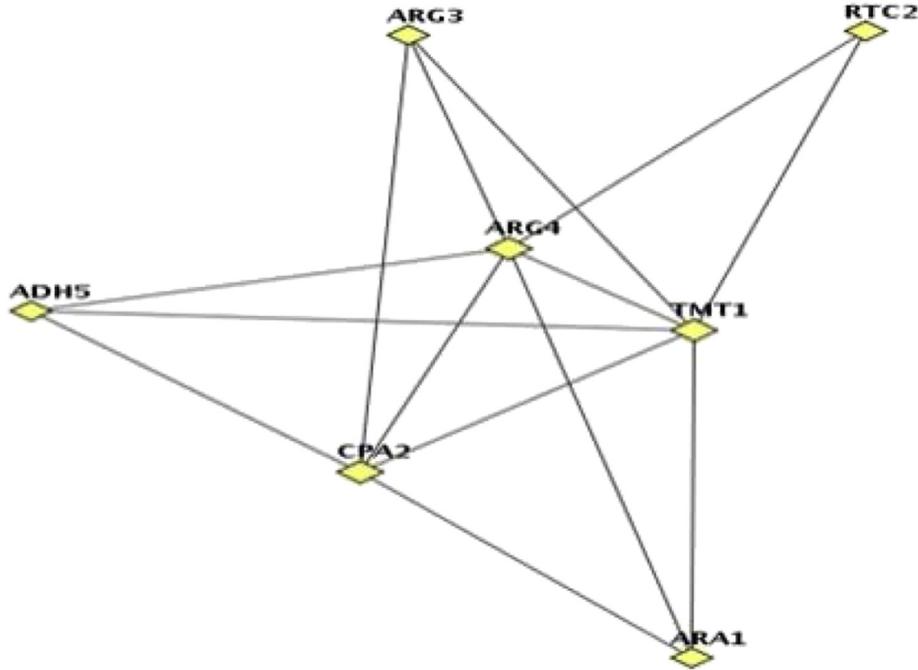

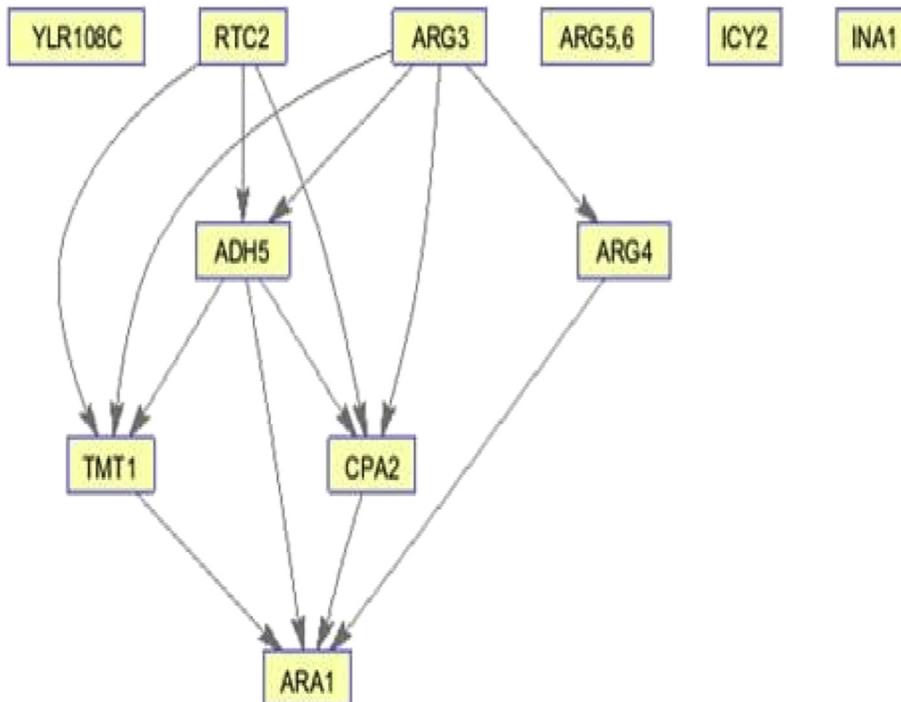Fig. 8. GRN generated by FunCoup 3.0 for Dataset with GEO Accession Number: GDS3030.

Fig. 9. GRN generated by FunCoup for Dataset with GEO Accession Number: GDS3030.

Similarly, the GRN generated by FunCoup 3.0 for other dataset with GEO accession number GDS3030 is shown in Fig. 8 while that of proposed method is shown in Fig. 9.

Table 2 shows the performance measures for different number of genes in microarray gene expression dataset with GEO Accession Number: GDS3030.

The performance measures of both microarray gene expression datasets are compared with existing method which was proposed by Al-Shobaili [19] in 2014. Fig. 10 represents the F-Score Comparison between proposed method and Al-Shobaili 2014 for microarray gene expression dataset with GEO accession number GDS37.

Fig. 11 represents the F-Score Comparison between proposed method for microarray gene expression dataset with GEO accession number GDS3030 with the method proposed by Al-Shobaili 2014 for different number of genes.

5. Conclusion

Gene regulatory networks are complex networks and are generated from the microarray gene expression datasets. Fuzzy logic based technique is robust and can tackle the incomplete and imprecise microarray gene expression datasets. The Woolf and Wang fuzzy logic based model is the most basic technique for the reverse engineering of GRN but have the number of limitations like high time complexity and less

Table 2
Performance measures of GDS3030.

Number of genes	Specificity	Sensitivity	F-score
7	0.75	0.86	0.80
8	0.72	0.67	0.70
9	0.67	0.72	0.70
10	0.67	0.72	0.70
11	0.7	0.64	0.67
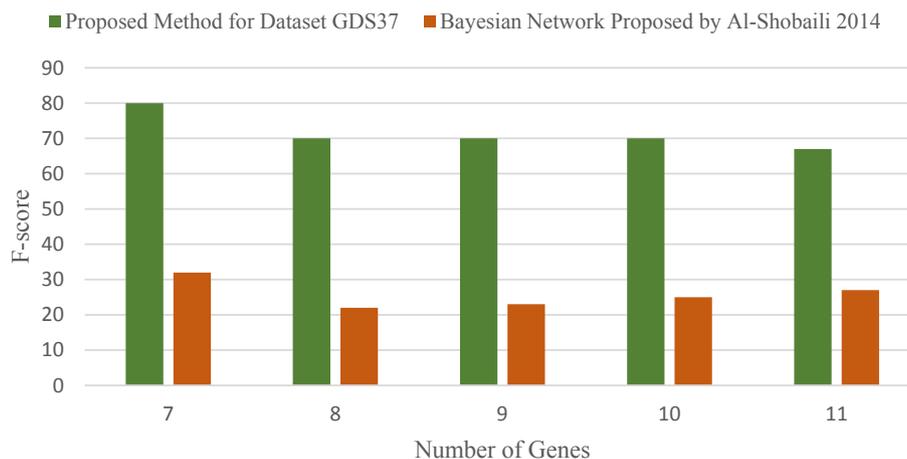
Fig. 10. F-Score Comparison of dataset GDS37 with method by Al-Shobaili 2014.
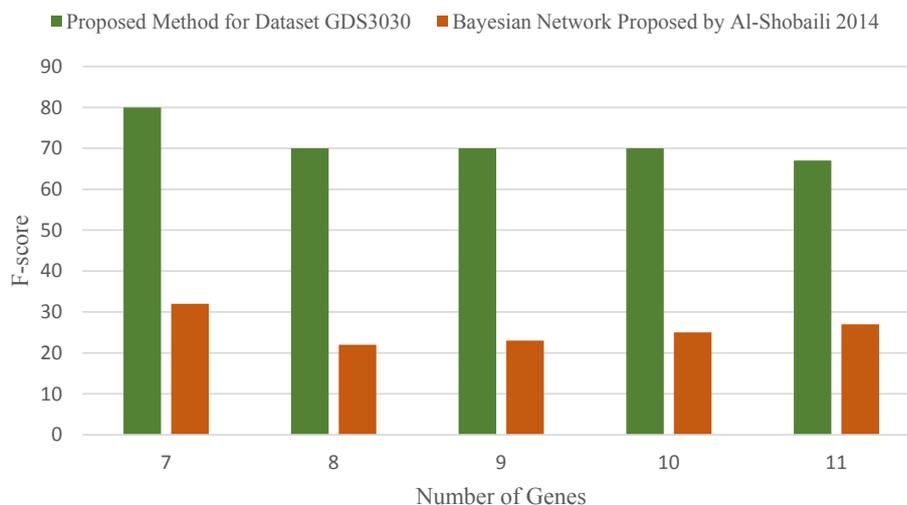

Fig. 11. F-Score Comparison of dataset GDS3030 with method proposed by Al-Shobaili 2014.

efficiency. A novel fuzzy logic based method is presented in this paper which introduces various significant steps like, filtering and clustering, that increase its effectiveness. Threshold limits are defined to decrease the time complexity of the proposed method to some extent. This method is tested on two different datasets of yeast with GEO accession number GDS37 and GDS3030. The results of both datasets are compared with the help of online database FunCoup 3.0. The results are further compared with the existing method Al-Shobaili 2014. The experimental results show that the proposed fuzzy logic based method for reverse engineering of GRN provides comparable results with the other existing method in terms of the F-Score, sensitivity, and specificity.

In future, the method can be tested for the scalability and generated GRN can be validated. In this research work, some novel interactions are found among the genes which need to be clinically tested in the laboratory.

Conflict of interest

The authors confirm that this article has no conflict of interest.

References

[1] Abhishek, Singh S. A gene regulatory network prediction method using particle swarm optimization and genetic algorithm. IJCA 2013;83:32–7.
[2] Abhishek, Singh S. Approaches to gene regulatory network modeling. IJIEASR 2013;2:35–8.
[3] Abhishek, Singh S. Gene selection using high dimensional gene expression data: an appraisal. Curr Bioinform 2016;11:1–9.
[4] Alakwaa FM. Modeling of gene regulatory networks: a literature review. J Comput Sys Biol 2014;1:1.
[5] Al-Quzlan T, Hamdi-Cherif A, Kara-Mohamed C. Big data fuzzy management methods in gene regulatory networks inference: a review. In: Proceedings of the 6th international conference on management of emergent digital ecosystems; 2014. p. 201–3.
[6] Bansal P, Singh S, Abhishek. A review on Bayesian network techniques for inferring gene regulatory networks. IJARSE 2015;4:386–91.
[7] Bordon J, Moskon M, Zimic N, Mraz M. Fuzzy logic as a computational tool for quantitative modelling of biological systems with uncertain kinetic data. In: Proceedings of IEEE computational intelligence society12; 2015. p. 1199–205.
[8] Cooper GF, Herskovits EA. Bayesian method for the induction of probabilistic networks from data. Mach Learn 1992;9:309–47.
[9] ElBakry O, Ahmad MO, Swamy MNS. Inference of gene regulatory networks from time-series microarray data. In: 8th IEEE international NEWCAS conference; 2010. p. 141–4.
[10] Gill Nivit, Singh S. Biological sequence matching using Boolean algebra vs. fuzzy logic. IJCA 2011;26:15–21.
[11] Gill Nivit, Singh S. Multiple sequence alignment using boolean algebra and fuzzy logic: a comparative study. Int J Comp Tech Appl 2011;2:1145–52.
[12] Heller MJ. DNA microarray technology: devices, systems, and applications. Annu Rev Biomed Eng 2002;4:129–53.
[13] Kacprzyk J. Studies in fuzziness and soft computing, vol. 242; 2009.
[14] Kanungo T, Mount DM, Netanyahu NS. An efficient k-means clustering algorithm: analysis and implementation. IEEE Trans Pattern Anal 2002; 24:881–92.
[15] Kaur R, Abhishek, Singh S. Inference of gene regulatory network using fuzzy logic – a review. IJCSE 2016;4:22–9.
[16] Kim H, Gelenbe E. Reconstruction of large-scale gene regulatory networks using bayesian model averaging. In: Proceedings of IEEE international conference on bioinformatics and biomedicine (BIBM); 2011. p. 202–7.
[17] Liu L, Liu M, Ma M. Construction of linear dynamic gene regulatory network based on feedforward neural network. In: Proceedings of 2014 IEEE 10th international conference on natural computation (ICNC); 2014. p. 99–107.
[18] Mahajan S, Abhishek, Singh S. Review on feature selection approaches using gene expression data. IJIR 2016;2:356–64.
[19] Mandai S, Saha G, Pal RK. Neural network based gene regulatory network reconstruction. In: IEEE third international conference on computer, communication, control and information technology (C3IT); 2015.
[20] Mandal S, Saha G, Pal RK. Neural network based gene regulatory network reconstruction. In: Proceedings of 2015 third international conference on computer, communication, control and information technology (C3IT); 2015. p. 1–5.
[21] Murphy KP. Dynamic bayesian networks: representation, inference and learning [PhD diss]. Berkeley: University of California; 2002.
[22] Ram R, Chetty M, Dix T. Fuzzy model for gene regulatory network. In: Proceeding of IEEE congress in evolutionary computation, CEC2006; 2006. p. 1450–5.
[23] Ram R, Chetty M, Dix TI. Fuzzy model for gene regulatory network. In: Proceedings of 2006 IEEE international conference on evolutionary computation; 2006. p. 1450–5.
[24] Ramot D, Friedman M, Langholz G, Kandel A. Complex fuzzy logic. IEEE Trans Fuzzy Syst 2003;11:450–61.
[25] Ressom HW, Zhang Y, Xuan J. Inference of gene regulatory networks from time course gene expression data using neural networks and swarm intelligence. In: Proceedings of 2006 IEEE symposium on computational intelligence and bioinformatics and computational biology CIBCB'06; 2006. p. 1–8.
[26] Ruz GA, Timmermann T, Goles E. Reconstruction of a GRN model of salt stress response in arabidopsis using genetic algorithms. In: IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB); 2015. p. 1–8.
[27] Schmitt T, Ogris C, Sonnhammer EL. FunCoup 3.0: database of genome-wide functional coupling networks. Nucleic Acids Res 2013;42:D380–8.
[28] Sehgal Muhammad, Shoaib B, Gondal I, Dooley LS. CF-GeNe: fuzzy framework for robust gene regulatory network inference. JCP 2006;7:1–8.
[29] Shermin A, Orgun M. A 2-stage approach for inferring gene regulatory networks using dynamic Bayesian networks. In: Proceedings of IEEE international conference on bioinformatics and biomedicine, BIBM'09; 2009. p. 166–9.
[30] Shmulevich I, Dougherty ER, Kim S, Zhang W. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics 2002;18:261–74.
[31] Sokhansansanj BA, Fitch JP, Quong JN, Quong AA. Exhaustive search for fuzzy gene networks from microarray data. In: Proceedings of the 25th annual international conference of the IEEE on engineering in medicine and biology society, 4; 2003. p. 3571–4.
[32] Vineetha S, Bhat C, Idicula SM. Gene regulatory network from microarray data using dynamic neural fuzzy approach. In: Proceedings of the international symposium on biocomputing; 2010.
[33] Woolf PJ, Wang Y. A fuzzy logic approach to analyzing gene expression data. Physiol Genomics 2000:3.
[34] Xiao Y. A tutorial on analysis and simulation of boolean gene regulatory network models. Curr Genomics 2009;10:511–25.