Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.777361

CASCADING MULTIPLE ACTIVATIONS OF AUGMENTED GENE REGULATORY NETWORKS

Saroj Shambharkar
Assistant Professor, Information Technology
K.I.T.S., Ramtek

Yogeshri Choudhari
Assistant Professor, Information Technology
K.D. K. College of Engineering, Nagpur

Rachna Somkunwar
Associate Professor, Computer Engineering
Dr. D. Y. Patil Institute of Technology, Pimri, Pune

Ritu S. Dudhmal
Assistant Professor, Computer Engineering
AISSMS-COE, Pune

Abstract— Gene Regulatory Networks (GReNes) are involved in cell functions and analysis of pathways. GReNes are also useful for inferring the relationships for analyzing and inference there is need of development of good Gene Regulatory Networks (GReNes) approach or models. These models include estimation of molecular interactions, design of simulation systems, estimation of system irregularities via perturbation analysis, biomedical treatment estimation, drug design simulations, etc. Design of GReNes is a complex process, and involves modelling of operations like correlation estimation, activation function design, fuzzy cascade of signals, etc. In order to perform these operations various GReNes are proposed, which include single cell Graph Neural Networks (scGNN), boosting GReNes, ensemble trees based GReNes, reverse engineered Bayesian model based GReNes, and single-cell regulatory network inference and clustering (SCENIC). Each of these models has scalability issues, which limits their deployment capabilities to application specific systems. In order to remove this drawback, the underlying paper proposes an augmented GReNe that uses multiple activations via cascading simpler networks. Due to augmentation and multiple activations, accuracy of the proposed model is observed to be 8% better when compared with state-of-the art models when inferred from single cell transcripts.

Keywords— Augmentation, cascade, Gene Regulatory Network, multiple activation, scalability

Introduction

Gene Regulatory Networks are used for a wide variety of biological applications, which include but are not limited to, simulation of medicine effects on target body, estimation of virus effects

on cells, testing of vaccinations, etc. This is possible due to Gene interactions, which allow for actuation of other Genes [1]. For instance, as observed in figure 1, Gene 1 (G1) consists of a mRNA sequence, which internally consists of a protein sequence (P1). This protein sequence P1 is also present in Gene 2 (G2), which is made up of mRNA 2 and protein sequence P2. Both P1 and P2 interact in order to form Gene 3, which is internally made up of mRNA 3 and protein P3. As a result of these interactions, it can be observed that any changes made to G1 directly affect activation of G2 & G3, similarly any changes made to G2 directly affect activation of G3.

These interactions between different Genes allow for creation of Gene regulatory network, where a cascade activation effect is observed . Various architectures for Gene regulation are designed by researchers, which include single cell Graph Neural Networks (scGNN), boosting GReNes, ensemble trees based GReNes, reverse engineered Bayesian model based GReNes, and single-cell regulatory network inference and clustering (SCENIC). Each of these architectures model the process of Gene regulation in a different manner, for instance, the SCENIC model initially estimates co-expressions from Gene data, and forms different Gene Matrices. Each of these Matrices is given to a motif discovery framework, where common expressions are estimated. These common expressions are then used for grouping Genes with similar activations, thereby forming clusters. All these clusters are ranked according to cell scoring model, which forms a regulon activity matrix that indicates network activities in each cell. Finally, a hierarchical clustering model is designed that groups different network activities in order to segregate working of each Gene expression, and estimate their actuation strength in the network.

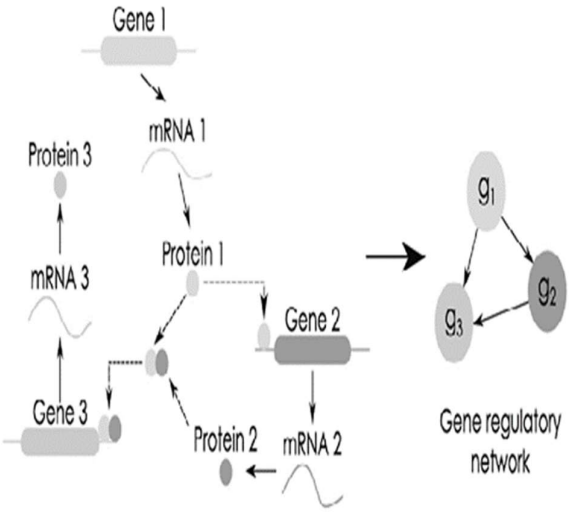

Figure 1: A sample Gene Regulatory Network (GReNe)

Each of these GReNe models have their own advantages and limitations. A survey of these models along with their performance is observed from the next section, where it can be observed that these models have limited scalability when applied to inter-species genomes. In order to improve this scalability, section 3 proposes design of an ensemble GReNe that can adapt to different species types, and model GReNes with high efficiency. This is followed by comparative result analysis and performance evaluation of the proposed model with respect to various state-of-the-art techniques. Finally, this paper concludes with some interesting

observations about the proposed GReNe model, and recommends methods to improve its performance.

Literature Survey

Designing GReNe models is a multidisciplinary task, that requires effective and working knowledge about signal processing, gene bio medics, neural network interconnections, function activations, etc. Thus, GReNe models are highly complex and require large delays for design and deployment. This can be observed by the models proposed in [2, 3, 4], where GReNe design using concepts like minimum set of master regulatory genes, multiple Laplacian & augmented Lagrangian combination models, and dynamic expansion of unstable neurons is used. These models aim at incorporating various mathematic identities to model GReNes with high efficiency. Each of these models require processing of large-scale gene data in order to design their own gene networks for various applications like analysis of bladder cancer [5], and other applications. The efficiency of these models can be improved via incorporation of deep learning methods. Such a method can be observed from [6, 7], where Recurrent Neural Network is combined with Simulated Annealing& Auto encoders are used in order to reconstruct GReNe models. These models have better accuracy and lower error rate when compared with [2, 3, 4], due to which they are highly applicable for real time gene applications. Optimization models for reducing delay of network design can be observed from [8], where time delayed GReNes are designed without the need of SUM regulatory circuit. This reduces computational complexity of the model, thereby making it applicable for high speed and high-performance applications like recognition of diabetic nephropathy [9], where fast activations are needed. Similar models can be observed from [10, 11, 12, 13], where gene trajectories, gene knock-out expression data, cluster assistive models, and reverse engineering models are used for improving the effectiveness of GReNe design & deployment. Application of these models for cancer cell detection can be observed from [14], where an efficiency of over 85% is observed for modelling chemo-resistant cancer cell networks. This efficiency can be improved by use of machine learning models for network design as discussed in [15, 16, 17, 18], where dependency analysis, temporal expression profiles, Bayesian inverse reinforcement learning and Bayesian data fusion models are used. These models aim at reducing error rate via iteratively tuning network performance for better accuracy. A survey of similar models can be observed from [19, 20, 21], where differential regulatory networks, causality inference methods, and limited memory networks are defined. Design practices from these methods can be applied to a wide variety of Gene Networks in order to reduce memory utilization, improve variance based processing and increase network causality.

Some of the most efficient GReNe models utilize gene clustering in order to initially group gene data, and then apply processing on these groups for high efficiency. Such models can be observed from [22, 23, 24], where SCENIC, scGNN, and boosted tree ensemble GReNe are defined. All these models are used for high accuracy applications due to their optimized edge estimation & network design performance. These models can be further optimized via use of data perturbation [25], addition of stability models [26], use of gradient boosted trees [27], and reducing oscillations in mRNA outputs [28]. All these models assist in improving activation response for each of the GReNe nodes, thereby assisting in a better error performance. Soft computing models like random forests [29], differential evolution [30], shortlisted candidate

regulators [31], Boolean network with perturbations [32], and partially-observed Boolean dynamical system [33] are also proposed by researchers. From these designs it can be observed that machine learning and deep learning models have been extensively used by researchers for improving GReNe design. But their application for design of internal sub networks is limited, which is covered by the proposed model that combines machine learning with sub network design for improved GReNe performance as observed from the next section.

Design of the augmented Gene Regulatory Network with multiple activations

As observed from the literature review section, a high accuracy GReNe is a combination of various mRNA and protein activations. As a result of these activations, certain parts of the network are turned ON, while other parts are turned OFF. Such a network is highly useful in modelling complex classification and simulation scenarios. But due to use of a single architecture for modelling GReNes, their efficiency is limited by error performance of the underlying architecture. In order to remove this drawback, this section proposes design of an augmented Gene Regulatory Network that uses multiple activations. Efficiency of the designed network is estimated and improved using various performance parameters, which includes false positive rate (FPR), true positive rate (TPR), positive predicted values (PPV), and matching of top-k edges (MTKE). Each of these parameters are evaluated using the following equations,

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \dots (1)$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \dots (2)$$

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \dots (3)$$

$$\mathrm{MTE_k} = \sum_{\mathrm{i=1}}^{\mathrm{k}} \frac{\left|\mathrm{GT_{E_i}} == \mathrm{D_{E_i}}\right|}{\mathrm{k}} \dots (4)$$

Where, $\mathrm{GT_{E_i}}$&$\mathrm{D_{E_i}}$are edges obtained via ground truth and designed algorithm, while TP,TN,FP,and FN are true positive, true negative, false positive, and false negative instances in the designed GReNe. These instances can be evaluated using the following equations,

$$\mathrm{TP} = \frac{\mathrm{N_{CC}}}{\mathrm{N_T}} \dots (5)$$

$$TN = \frac{N_{CI}}{N_T} \dots (6)$$

$$FP = \frac{N_{IC}}{N_T} \dots (7)$$

$$FN = \frac{N_{II}}{N_T} \dots (8)$$

Where, $\mathrm{N_{CC}}$ is Number of correct edges with correct activations,$\mathrm{N_{CI}}$ is Number of correct edges with incorrect activations, $\mathrm{N_{IC}}$ is Number of incorrect edges with correct activations, $\mathrm{N_{II}}$ is Number of incorrect edges with incorrect activations and$\mathrm{N_T}$ is total number of edges. In order to train the proposed network model, training sets of ground truth networks are input to it, and an iterative process is evaluated. This process is followed by incremental learning model for

improving efficiency of GReNe design, and updating its internal build structure. Both of these operations are described in separate sub-sections.

A) Iterative process for GReNe design

In this sub-section an iterative process for design of a GReNe is described. This process uses sub network division, and parametric optimization for designing a highly accurate GReNe network using multiple sub networks. The process can be described as follows,

- *Input the following parameters,*
  - Number of generations (N_G)
  - Number of networks (N_N)
  - Maximum number of sub networks that can be formed ($N_{SN_{MAX}}$)
  - Gene learning rate (G_{LR})
  - Maximum number of algorithms available for gene network design ($N_{GN_{A_{MAX}}}$)
  - Ground truth networks for training (N_{GT})
- *Output,*
  - Ensemble of algorithms along with their activation positions for highly accurate network design
- *Design Process*
  - For each generation in 1 to N_G
    - For each network in 1 to N_N
    - If the network is marked as 'not to be mutated', then go to the next network.
    - Else, generate a new GReNe using the following process,
      - Generate a random number between 1 to $N_{SN_{MAX}} = N_{sel}$

$$\mathrm{N_{sel}} = \mathrm{RAND}\left(1, \mathrm{N_{SN_{Max}}}\right) \dots (9)$$

      - For each ground truth network in 1 to N_{GT}, split each network into N_{sel} equal parts
      - For each ground truth network, and each part in 1 to N_{sel},
        - Generate a random number between 1 to $N_{GN_{A_{MAX}}}$, which will assist in designing the GReNe for this sub network,

$$\mathrm{N_{GReNE}} = \mathrm{RAND}\left(1, \mathrm{N_{GN_{A_{MAX}}}}\right) \dots (10)$$

      - Design the sub network using the GReNe at index $\mathrm{N_{GReNE}}$, and estimate its TPR, FPR, PPV and $\mathrm{MTE_k}$ values from equation 1, 2, 3, and 4.
      - Repeat this process until the complete network is designed, and evaluate network correctness value (NCV) using the following equation,

$$\mathrm{NCV} = \sum_{\mathrm{i=1}}^{\mathrm{N_{sel}}} \mathrm{TPR} + \mathrm{PPV} + \mathrm{MTE_k} - \mathrm{FPR} \Big/ \mathrm{N_{sel}} \dots (11)$$

    - Find NCV for each generated network, and then evaluate network correctness threshold using the following equation,

$$\mathrm{N_{CT}} = \sum_{\mathrm{i=1}}^{\mathrm{N_N}} \mathrm{NCV_i} * \frac{\mathrm{G_{LR}}}{\mathrm{N_N}} \dots (12)$$

    - Upon estimation of NCV for each network, mark all networks as 'to be mutated', that satisfy equation 13, while mark all others as 'not to be mutated'

$$\mathrm{NCV_i} < \mathrm{N_{CT}} \dots (13)$$

    - Repeat this process for all generations, and create a table, with the following structure as observed from table 1.

Table 1. Generated network table for GReNe design (GN is Generation Number)

GN	N_{sel}	N_{GReNe} for each N_{sel}	NCV

- Select the network that has maximum value of NCV, and use the selected networks for designing underlying GReNe

Once the GReNe is designed, and deployed in any underlying application; then an incremental learning model is activated. This model aims at iteratively scanning performance of the designed GReNe model, and incrementally improving its performance.

B) Incremental learning model for improving GReNe performance

In this sub-section an incremental learning model for improving efficiency of the proposed GReNe is defined. This process can be applied to any ensemble GReNe model for improving its overall accuracy, precision and matching of top-k edges values. The process can be described as follows,

- Select a validation set (separate from training set), where inputs to each internal GReNe model, and its outputs are known.
- For each entry in the validation set, perform the following steps,
  - Provide the validation value to internal GReNe network, and evaluate its NCV
  - Compare the NCV with its original value as observed from table 1.
  - If the NCV value is reducing, then mark this sub network as 'to be changed'
- For all the sub networks, which are marked as 'to be changed', follow the given process in this step, and generate new sub networks with better performance metrics
  - Generate a random number between 1 to $N_{SN_{MAX}} = N_{sel}$
  - For each ground truth network in 1 to N_{GT}, split each network into N_{sel} equal parts
  - For each ground truth network, and each part in 1 to N_{sel},
    - Generate a random number between 1 to $N_{GN_{A_{MAX}}}$, which will assist in designing the GReNe for this sub network
    - Design the sub network using the GReNe at index N_{GReNE}, and estimate its TPR, FPR, PPV and MTE_k values from equation 1, 2, 3, and 4.
  - Repeat this process until the complete network is designed, and evaluate network correctness value (NCV) using equation 11.
  - Use this new sub network if it satisfies the following equation,

$$NCV_{new} > NCV_{old} \dots (14)$$

- Replace existing sub networks with this new configuration, and redeploy the modified GReNe model

The incremental learning process is performed whenever a validation set is available, or a new ground truth training set is used in place of the existing training set. This process allows incremental improvement in the GReNe model performance, thereby improving its internal efficiency. This efficiency is estimated in terms of area under receiver operating characteristics (AUROC), TPR, PPV and MTE_k values, under different datasets and can be observed from the next section.

Results and comparative analysis

In order to estimate performance of the proposed ensemble learning GReNe model, it was evaluated using the public Zenodo dataset (available at https://zenodo.org/record/3701939). This dataset consists of curated and synthetic datasets of mammalian cortical area development (mCAD), hematopoietic stem cell (HSC), ventral spinal cord (VSC), etc. A total network size of 5000 edges can be created using these datasets, and 50 different expressions per network can be developed. Therefore, a total combination of 250k GReNes can be made from this dataset. In this evaluation, total dataset of 250k GReNes is divided into training, testing and validation sets, in a ratio of 60:30:10, therefore 150k GReNes are used for training the model, while 75k are used for testing, and the remaining 25k are used for validation or incremental learning. For every 3k tested networks, 1k networks are used for validation, which assists in boosting performance of the generated networks. This performance can be observed from figure 2, where AUROC (AR) of different GReNe models can be seen under different testing set sizes.

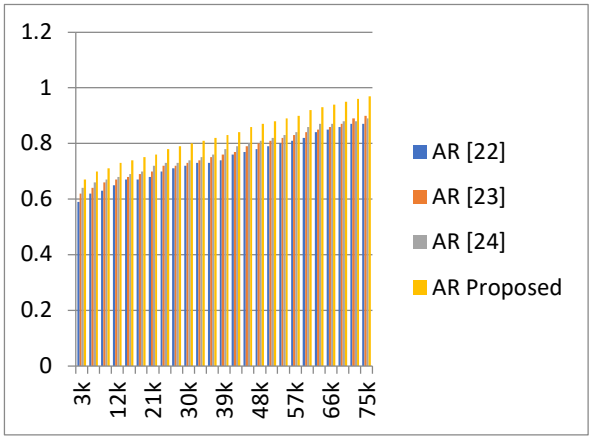

Figure 2: AUROC performance for different GReNes

The AUROC values show an improvement of 10% when compared with [22], 7% when compared with [23] and 8% when compared with [24], thereby making it highly useful for applications that demand high accuracy. This performance is accompanied with high TPR performance, which can be observed from figure 3, where the same algorithms are compared under similar simulation conditions.

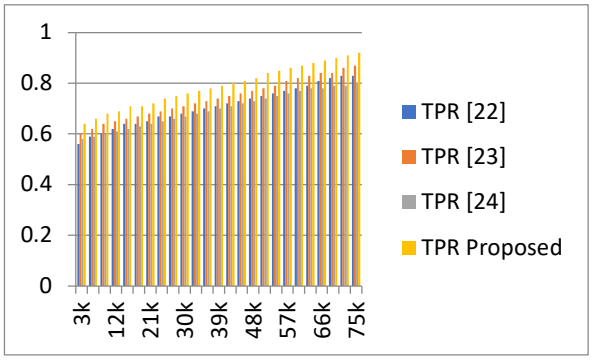

Figure 3: TPR performance for different GReNes

The TPR values show an improvement of 9% when compared with [22], 5% when compared with [23] and 12% when compared with [24], thereby making it highly useful for applications

that demand high performance. This performance is accompanied with high PPV performance, which can be observed from figure 4, where the same algorithms are compared under similar simulation conditions.

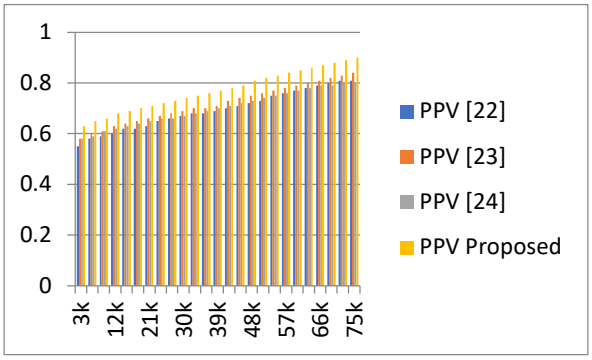

Figure 4: PPV performance for different GReNes

The PPV values show an improvement of 9% when compared with [22], 6% when compared with [23] and 10% when compared with [24], thereby making it highly useful for applications that demand high positive predictions. This prediction performance is accompanied with high MTE_k values, which can be observed from figure 5, where the same algorithms are compared under similar simulation conditions.

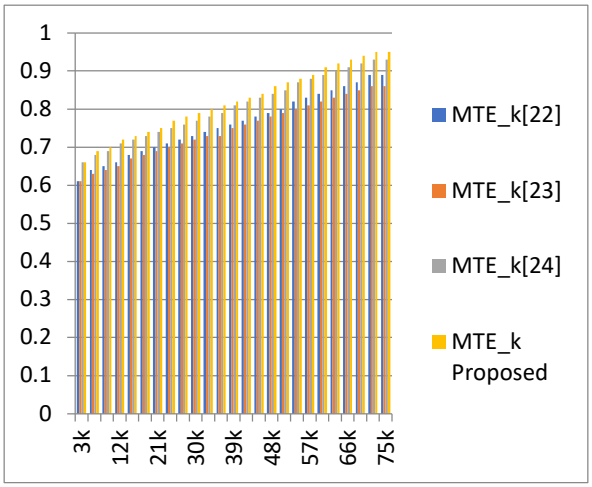

Figure 5: MTE_k performance for different GReNes

The MTE_k values show an improvement of 6% when compared with [22], 9% when compared with [23] and 2% when compared with [24], thereby making it highly useful for applications that demand highly accurate edge outputs. These results indicate that the proposed model is highly efficient, and can be used for design of high accuracy, low error, and better edge performance GReNe models. Average performance of all these parameters is evaluated, and can be observed from table 2, where superiority of the proposed model can be seen.

Table 2. Average performance of different models

Model	**AR**	**TPR**	**PPV**	$\boldsymbol{MTE_k}$

[22]	0.75	0.71	0.69	0.76
[23]	0.76	0.74	0.72	0.75
[24]	0.78	0.7	0.71	0.81
Proposed	0.83	0.79	0.77	0.82

From the average performance, it can be observed that the proposed model is nearly 8% better than [22], 6% better than [23], and 6% better than [24] when compared across multiple parameters. This makes the underlying GReNe design model highly efficient, and improves its scalability to a wider number of applications.

CONCLUSION

Design of a GReNe requires efficient modelling of internal activation units. These units allow the system model to activate and deactivate certain parts of the network, thereby assisting in improved network design. The proposed model breaks the GReNe into multiple sub networks, and aims at optimizing the TPR, PPV, and top-k edge selection performance of each sub network. A combination of these highly optimized sub networks is used to form the final GReNe, which showcases highly improved performance. This performance is estimated across 75k different networks, via training the model on over 150k different networks in order to estimate its real time parameters like PPV, TPR, AUROC, and MTE_k. Moreover, due to incremental learning via the 25k validation networks, the proposed model is able to achieve 10% better AUROC when compared with [22], 7% when compared with [23] and 8% when compared with [24], thus making the underlying network design suitable for high accuracy applications. Similar performance improvements were observed for other parameters, for instance, TPR values show an improvement of 9% when compared with [22], 5% when compared with [23] and 12% when compared with [24]; while, MTE_k values show an improvement of 6% when compared with [22], 9% when compared with [23] and 2% when compared with [24]. An average improvement of nearly 8% when compared with [22], 6% better when compared with [23], and 6% better when compared with [24] can be observed. Due to such a high performance, this network is capable of deployment in applications that require high accuracy, low error rate, and high edge accuracy. In future, researchers can work on reducing the delay of network formation; which is very high due to splitting the network into sub networks, and then estimation & incremental performance improvement. Machine learning techniques can be used to reduce this delay, and optimize network performance for high-speed applications.

References

[1] T. Xu, L. Ou-Yang, X. Hu and X. -F. Zhang, "Identifying Gene Network Rewiring by Integrating Gene Expression and Gene Network Data," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 15, no. 6, pp. 2079-2085, 1 Nov.-Dec. 2018, doi: 10.1109/TCBB.2018.2809603.

[2] S. Bakhteh, A. Ghaffari-Hadigheh and N. Chaparzadeh, "Identification of Minimum Set of Master Regulatory Genes in Gene Regulatory Networks," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 17, no. 3, pp. 999-1009, 1 May-June 2020, doi: 10.1109/TCBB.2018.2875692.

[3] L. Zhang, H. -C. Wu, C. -H. Ho and S. -C. Chan, "A Multi-Laplacian Prior and Augmented Lagrangian Approach to the Exploratory Analysis of Time-Varying Gene and Transcriptional Regulatory Networks for Gene Microarray Data," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 16, no. 6, pp. 1816-1829, 1 Nov.-Dec. 2019, doi: 10.1109/TCBB.2018.2828810.

[4] M. Zamanighomi, M. Zamanian, M. Kimber and Z. Wang, "Gene Regulatory Network Inference from Perturbed Time-Series Expression Data via Ordered Dynamical Expansion of Non-Steady State Actors," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 15, no. 4, pp. 1093-1106, 1 July-Aug. 2018, doi: 10.1109/TCBB.2015.2509992.

[5] Q. CHEN, M. JIANG, Y. ZENG, Y. ZHANG and D. GAO, "Gene Regulatory Networks Analysis for Subtypes of Muscle-Invasive Bladder Cancer Based on Pathway," 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing, 2019, pp. 92-95, doi: 10.1109/ICCWAMTIP47768.2019.9067508.

[6] Biswas, Surama & Acharyya, Sriyankar. (2017). A Bi-Objective RNN Model to Reconstruct Gene Regulatory Network: A Modified Multi-Objective Simulated Annealing Approach. IEEE/ACM Transactions on Computational Biology and Bioinformatics. PP. 1-1. 10.1109/TCBB.2017.2771360.

[7] I. Bilgen and Ö. S. Saraç, "Gene regulatory network inference from gene expression dataset using autoencoder," 2018 26th Signal Processing and Communications Applications Conference (SIU), 2018, pp. 1-4, doi: 10.1109/SIU.2018.8404816.

[8] Jiao, Hongmei & Zhang, Liping & Shen, Qikun & Zhu, Junwu & Shi, Peng. (2018). Robust Gene Circuit Control Design for Time-Delayed Genetic Regulatory Networks Without SUM Regulatory Logic. IEEE/ACM Transactions on Computational Biology and Bioinformatics. PP. 1-1. 10.1109/TCBB.2018.2825445.

[9] R. Zheng, Y. Wang, Z. Lyu and A. Armaou, "Gene Regulatory Network Construction and Key Gene Recognition of Diabetic Nephropathy," 2019 IEEE 7th International Conference on Bioinformatics and Computational Biology (ICBCB), 2019, pp. 1-6, doi: 10.1109/ICBCB.2019.8854648.

[10] A. Karbalayghareh, U. Braga-Neto, J. Hua and E. R. Dougherty, "Classification of State Trajectories in Gene Regulatory Networks," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 15, no. 1, pp. 68-82, 1 Jan.-Feb. 2018, doi: 10.1109/TCBB.2016.2616470.

[11] X. Chen et al., "A novel method of gene regulatory network structure inference from gene knock-out expression data," in Tsinghua Science and Technology, vol. 24, no. 4, pp. 446-455, Aug. 2019, doi: 10.26599/TST.2018.9010097.

[12] Pirayre A, Couprie C, Duval L, Pesquet JC. BRANE Clust: Cluster-Assisted Gene Regulatory Network Inference Refinement. *IEEE/ACM Trans Comput Biol Bioinform*. 2018;15(3):850-860. doi:10.1109/TCBB.2017.2688355

[13] A. Khan, G. Saha and R. K. Pal, "Modified Half-System Based Method for Reverse Engineering of Gene Regulatory Networks," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 17, no. 4, pp. 1303-1316, 1 July-Aug. 2020, doi: 10.1109/TCBB.2019.2892450.

[14] Jimenez-Vargas, Dennis & Acon, Man & Şahin, Özgür & Eyupoglu, Erol & Riazalhosseini, Yasser & Molina Mora, José & Guevara-Coto, Jose & Mora-Rodriguez,

Rodrigo. (2018). Modules of Correlated Genes in a Gene Expression Regulatory Network of CDDP-Resistant Cancer Cells. 1-6. 10.1109/IWOBI.2018.8464190.

[15] Jianxiao Liu, Jianbing Yan, Zonglin Tian, Yingjie Xiao, Haijun Liu, Songlin Hao, Xiaolong Zhang, Chaoyang Wang, Jianchao Sun, and Huan Yu. 2020. Gene Regulatory Relationship Mining Using Improved Three-Phase Dependency Analysis Approach. IEEE/ACM Trans. Comput. Biol. Bioinformatics 17, 1 (Jan.-Feb. 2020), 339–346. DOI: https://doi.org/10.1109/TCBB.2018.2872993

[16] A. Majumder, M. Sarkar and P. Sharma, "A Composite Mode Differential Gene Regulatory Architecture based on Temporal Expression Profiles," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 16, no. 6, pp. 1785-1793, 1 Nov.-Dec. 2019, doi: 10.1109/TCBB.2018.2828418.

[17] M. Imani and U. M. Braga-Neto, "Control of Gene Regulatory Networks Using Bayesian Inverse Reinforcement Learning," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 16, no. 4, pp. 1250-1261, 1 July-Aug. 2019, doi: 10.1109/TCBB.2018.2830357.

[18] H. Chen, D. Maduranga, P. Mundra and J. Zheng, "Bayesian Data Fusion of Gene Expression and Histone Modification Profiles for Inference of Gene Regulatory Network" in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 17, no. 02, pp. 516-525, 2020.doi: 10.1109/TCBB.2018.2869590

[19] Y. Chiu, Kai-Wen Liang, T. Hsiao, Y. Chen and E. Y. Chuang, "Analyzing differential regulatory networks modulated by continuous-state genomic features in glioblastoma multiforme," 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2015, pp. 171-176, doi: 10.1109/BIBM.2015.7359676.

[20] S. S. Ahmed, S. Roy and J. Kalita, "Assessing the Effectiveness of Causality Inference Methods for Gene Regulatory Networks," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 17, no. 1, pp. 56-70, 1 Jan.-Feb. 2020, doi: 10.1109/TCBB.2018.2853728.

[21] S. Pyne and A. Anand, "Rapid Reconstruction of Time-varying Gene Regulatory Networks with Limited Main Memory," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, doi: 10.1109/TCBB.2019.2946826.

[22] Aibar, S., González-Blas, C., Moerman, T. *et al.* SCENIC: single-cell regulatory network inference and clustering. *Nat Methods* **14,** 1083–1086 (2017).

[23] Wang, J., Ma, A., Chang, Y. *et al.* scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses. *Nat Commun* **12,** 1882 (2021).

[24] Park, Sungjoon & Kim, Jung & Shin, Wonho & Han, Sung & Jeon, Minji & Jang, Hyun & Jang, Ik-Soon & Kang, Jaewoo. (2018). BTNET: Boosted tree based gene regulatory network inference algorithm using time-course measurement data. BMC systems biology. 12. 20. 10.1186/s12918-018-0547-0.

[25] M. Foo, J. Kim and D. G. Bates, "Modelling and Control of Gene Regulatory Networks for Perturbation Mitigation," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 16, no. 2, pp. 583-595, 1 March-April 2019, doi: 10.1109/TCBB.2017.2771775.

[26] Zhaohua Wu, Zhiming Wang, Tiejun Zhou, "Finite-Time Stability of Fractional-Order Time-Varying Delays Gene Regulatory Networks with Structured Uncertainties and Controllers", *Complexity*, vol. 2020, Article ID 2315272, 19 pages, 2020.

[27] Y. Zhang, X. Zhang, A. N. Lane, T. W. . -M. Fan and J. Liu, "Inferring Gene Regulatory Networks of Metabolic Enzymes Using Gradient Boosted Trees," in IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 5, pp. 1528-1536, May 2020, doi: 10.1109/JBHI.2019.2931997.

[28] T. Dong and Q. Zhang, "Stability and Oscillation Analysis of a Gene Regulatory Network With Multiple Time Delays and Diffusion Rate," in IEEE Transactions on NanoBioscience, vol. 19, no. 2, pp. 285-298, April 2020, doi: 10.1109/TNB.2020.2964900.

[29] Wang, Hailun & Sham, Pak & Tong, Tiejun & Pang, Herbert. (2019). Pathway-Based Single-Cell RNA-Seq Classification, Clustering, and Construction of Gene-Gene Interactions Networks Using Random Forests. IEEE Journal of Biomedical and Health Informatics. PP. 1-1. 10.1109/JBHI.2019.2944865.

[30] H. A. Chowdhury, D. K. Bhattacharyya and J. K. Kalita, "(Differential) Co-Expression Analysis of Gene Expression: A Survey of Best Practices," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 17, no. 4, pp. 1154-1173, 1 July-Aug. 2020, doi: 10.1109/TCBB.2019.2893170.

[31] S. Pyne and A. Anand, "Rapid Reconstruction of Time-varying Gene Regulatory Networks with Limited Main Memory," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, doi: 10.1109/TCBB.2019.2946826.

[32] A. Karbalayghareh, U. Braga-Neto and E. R. Dougherty, "Classification of Single-Cell Gene Expression Trajectories from Incomplete and Noisy Data," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 16, no. 1, pp. 193-207, 1 Jan.-Feb. 2019, doi: 10.1109/TCBB.2017.2763946.

[33] M. Imani and U. M. Braga-Neto, "Control of Gene Regulatory Networks With Noisy Measurements and Uncertain Inputs," in IEEE Transactions on Control of Network Systems, vol. 5, no. 2, pp. 760-769, June 2018, doi: 10.1109/TCNS.2017.2746341.