MiRNN: A mutual information augmented recurrent neural network framework for reconstruction of gene regulatory networks

Prianka Dey*§, Abhinandan Khan†§, Goutam Saha‡, and Rajat Kumar Pal§
*Department of Computer Science and Engineering, Narula Institute of Technology,
81 Nilgunj Road, Kolkata – 700109, India.
Email: priankadey2011@gmail.com,
†Product Development and Diversification, ARP Engineering, 147 Nilgunj Road, Kolkata – 700056, India.
Email: khan.abhinandan@gmail.com
‡Department of Information Technology, North Eastern Hill University, Shillong – 793022, Meghalaya, India.
Email: dr.goutamsaha@gmail.com
§Department of Computer Science and Engineering, University of Calcutta,
Acharya Prafulla Chandra Roy Siksha Prangan, JD-2, Sector-III, Saltlake, Kolkata – 700106, India.
Email: pal.rajatk@gmail.com

***Abstract*—Genes act as the blueprint for regulating all activities of a living system. Genes produce proteins, which in turn, sit on the promoter regions of other genes to regulate their activity. Thus, a gene regulatory network is formed. This network is critical in disclosing the various mysteries in the operations of living systems. Often it is very difficult to find these networks in the Wet Lab. As a result, various computational approaches have been used to reconstruct these networks from gene expression data. The techniques primarily used for this purpose include *Bayesian* networks, *Boolean networks*, *recurrent neural networks*, *S-systems*, and *mutual information based methods*. The contemporary literature indicates that these techniques often fail to reliably reconstruct real-life networks. In this paper, we have proposed a new technique based on a modified recurrent neural network strategy that is augmented by mutual information. The proposed methodology has been implemented on an 8-gene network of *Escherichia coli* and a 10-gene network, which have been extensively used by other researchers. The experimental results indicate that the proposed technique achieves satisfactory results when compared to other such techniques developed by contemporary researchers.**

***Index Terms*—gene regulatory network, mutual information, recurrent neural network, reverse engineering**

I. Introduction

Genes act as the blueprints for all cellular activities. In a living system, genes produce proteins and a single protein or a group of proteins regulate other gene/s. Thus, a *gene regulatory network* or GRN is formed. These networks contain a lot of information regarding several intrinsic activities of cellular systems. Many of the cellular activities are still a mystery for researchers, and thus, a structural study of GRNs becomes very important for the unfolding of many such mysteries. Practically, this may potentially mean drug design and the treatment of various diseases originating from genetic disorders. However, it is very difficult to extract these networks in the Wet Lab.

Thus, computational approaches are the only alternative to reconstruct GRNs from time-series gene expression data. Due to the advent of newer technologies, it is no longer costly to produce such data. Past researchers have investigated many techniques for reverse engineering GRNs from temporal expression datasets. These techniques include *Boolean networks* (BoN) [1], *Bayesian networks* (BaN) [2], [3], *recurrent neural networks* (RNN) [4], *S-systems* [5], mutual information based tools, like GENIE3 [6] and ARACNE [7], [8], etc.

On evaluating the performances of these techniques, it becomes clear that these reverse engineering methodologies can perform satisfactorily in the case of smaller networks. However, even for such small-scale GRNs, the existing methodologies require a significant amount of computational time. Moreover, the number of incorrect predictions are also usually much more compared to the number of correct ones. On the other hand, in the case of larger networks, most state-of the-art techniques cannot achieve satisfactory results, especially for networks with more than 50 genes.

Therefore, in this work, a newer technique has been proposed, where our endeavour is to augment the concepts of mutual information into the RNN framework. The proposed methodology comprises two steps: (1) extracting an undirected network from the temporal expression data using the concepts of mutual information and (2) identifying the directions of the network edges (i.e., the nature of regulation) and the corresponding edge weights (i.e., the type of the regulation, whether activation or inhibition) using RNN.

The proposed framework has been named as MI-RNN. Experiments have been performed on two networks: the 8-gene SOS DNA repair network of *E. coli* and a 10-gene network extracted from GNW [9], both of which have been used extensively by researchers of this domain. In both the experiments, better performance has been achieved.

The rest of the paper has been organised as follows. Section II introduces the related works in this domain. In Section III, the proposed methodology has been explained in detail. Experimental results have been presented in Section IV along with relevant discussions. The paper concludes with Section V.

II. Related Works and Preliminaries

A. Background

Before getting into the specifics of the methodology proposed in this work, here in this section, we have briefly introduced some of the models/approaches that researchers have used to infer GRNs from time series expression datasets. Over the years, various computational methods have been implemented for reconstruction of GRNs, viz., BaNs, BoNs, RNNs, S-systems and half-systems, regression based approaches, etc.

The first and one of the oldest methods for reconstruction of GRNs is *Boolean networks* (BoNs) [10]. Together with biological phenomena like oscillation, multi-stationarity, switch-like behaviour, and hysteria, it aids in determining the dynamics of gene regulation [11]. In the contemporary literature, we can usually find two different BoN based methodologies: *correlation* and *inferring* [12]. Akutsu et al. [13] proposed a simpler algorithm and proved that, for n number of nodes in a network, only $O(\log n)$ state transition pairs are required for the identification of the original BoN (representing the original GRN). BoNs are deterministic in nature and can handle large-scale GRNs. However, BoN based methodologies failed to reduce the required computational time and handle incomplete expression datasets. To overcome this problem, various models based on *probabilistic Boolean networks* has been proposed [14]–[17].

Bayesian network (BaN) is another approach in the field of GRN reconstruction, which combines graph theory, using the *directed acyclic graph* (DAG), with a probabilistic approach. Friedman et al. [18] first implemented BaNs fore reverse engineering GRNs using a graph $G(V, E)$, which represents the interactions between the genes. V denotes the set of nodes (which represent the genes) and E represents the set of edges (which constitute the interactions between the genes). BaNs can handle the noise present in the network and uncertainty and incorporate past information to strengthen the causal connections between genes but the primary issue with BaN is its inability to accurately reflect self-regulation. Its incapability to manage a large-scale GRN network and increased computation cost cause it to become less significant. To resolve this issue *dynamic Bayesian network* has been proposed [19]–[22].

Another approach for reverse engineering GRNs from temporal expression data is based on regression analysis. A well-known tool is GENIE3 [6] and its subsequent improved and extended version, dynGENIE3 [23]. In GENIE3 [6], the problem is split into p distinct regression problems for p inferred networks in this approach. The expression profiles of all the other genes in each sub problem are used to forecast the expression profiles of each gene using ensemble processes like *random forests* or *extra-trees*. It is fast and scalable, builds directed GRNs, can handle combinatorial and non-linear interactions, and makes no assumptions about the nature of gene regulation.

Another entropy based approach that is suggested for GRN reconstruction is ARACNE [7], [8]. It is one of the most used approaches, where mutual information is found using entropy, which then provides an estimate of the amount of information transferred between the genes. ARACNE employs an innovative method called *data processing inequality* (DPI) to define the information interactions. Despite its ability to identify all gene-gene interactions, ARACNE cannot provide appropriate guidance regarding the nature of such interactions.

B. S-systems

S-system (SS) [5] is a power law based technique that, in comparison to other systems, helps provide a more accurate interpretation of the gene-to-gene interactions by faithfully representing the two crucial real-world aspects of biological behaviour: *saturation* and *synergy*. The majority of biological phenomena, according to literature reviews [5], [24], can be represented by a power law, which is why SS is popular among researchers in this domain.

For example, Noman et al. [25] developed a decoupled SS formalism for the purpose of inferring GRNs through the use of *trigonometric differential evolution* (TDE), an extension of *differential evolution* or DE, for the optimisation of S-system model parameters. The mathematical formulation of SS is as follows:

$$\frac{dx_i(t)}{dt} = \alpha_i \cdot \prod_{k=1}^{N} [x_k(t)]^{g_{i,j}} + \beta_i \cdot \prod_{k=1}^{N} [x_k(t)]^{h_{i,j}}, \quad (1)$$

where N is the number of genes in the network, x_i and x_k are the expression levels of the ith and the kth gene, respectively, where $i, k = 1, \ldots, N$; $g_{i,j}$ and $h_{i,j}$ are the exponential tuning parameters, and α_i and β_i are the rate constants. Activation is indicated by the first term on the right hand side of (1) and inhibition by the second. Using (1), the expression level, x_i, of the ith gene at any time-point $t+\Delta t$ can be defined as follows, taking the approximation of $\frac{dx(t)}{dt} \approx \frac{\Delta x(t)}{\Delta t} = \frac{x_i(t+\Delta t)-x_i(t)}{\Delta t}$:

$$x_i(t+\Delta t) = x_i(t) + (\Delta t \cdot \alpha_i) \cdot \prod_{k=1}^{N} [x_k(t)]^{g_{i,j}} - (\Delta t \cdot \beta_i) \cdot \prod_{k=1}^{N} [x_k(t)]^{h_{i,j}} \quad (2)$$

Since this strategy involves $2N + 2$ number of parameters, the primary disadvantage of SS is its much higher computational time requirement. An additional issue with this widely used power law method is that it may identify both activation and inhibition simultaneously for the ith gene at time point t, i.e., $g_{i,j}, h_{i,j} \neq 0$ which is usually not feasible in any biological system. Additionally, if $g_{i,j}$ and $h_{i,j}$ have the same sign it indicates dual regulation, which is again biologically improbable.

C. Half-Systems

To resolve the above-mentioned problems another approach was proposed by Khan et al. [26] for the reconstruction of GRNs, which is modified *half-systems* (HS). The mathematical formulation of the traditional HS formalism is given by the following:

$$\frac{dx_i\left(t\right)}{dt} = \alpha_i \cdot \prod_{k=1}^{N} \left[x_k\left(t\right)\right]^{g_{i,j}}, \tag{3}$$

where α_i is the only rate constant, $g_{i,j}$ is the only tuning parameter. It stipulates that gene k is activating gene i if $g_{i,j} > 0$, and that gene k is inhibiting gene i if $g_{i,j} < 0$. For this reason, the HS formalism is superior than S-systems in two ways: (*i*) compared to $2N+2$, just $N+1$ parameters need to be trained, and (*ii*) as there is just one kinetic parameter $g_{i,j}$, it is impossible to predict any dual regulation. In addition to this, another advantage of using HS is that it is also a power law based approach which gives the same advantage as SS.

A self-degradation term that improves the robustness of the GRN reconstruction was added to (3) by Khan et al. [26], as follows:

$$\frac{dx_i\left(t\right)}{dt} = \alpha_i \cdot \prod_{k=1}^{N} \left[x_k\left(t\right)\right]^{g_{i,j}} - \epsilon_i \cdot x_i\left(t\right) \tag{4}$$

Using this modified equation, the expression level x_i of the ith gene at any time-point $t+\Delta t$ can be defined as follows, taking the approximation of $\frac{dx(t)}{dt} \approx \frac{\Delta x(t)}{\Delta t} = \frac{x(t+\Delta t)-x(t)}{\Delta t}$:

$$\begin{aligned} x_i\left(t+\Delta t\right) = {} & \left(\Delta t \cdot \alpha_i\right) \cdot \prod_{k=1}^{N} \left[x_k\left(t\right)\right]^{g_{i,j}} \\ & + \left(1 - \Delta t \cdot \epsilon_i\right) \cdot x_i\left(t\right) \end{aligned} \tag{5}$$

where ϵ_i is the feedback term which helps in the self degradation of the system.

D. Recurrent Neural Networks

Another effective network reconstruction method is *recurrent neural network* (RNN). Using time-series expression datasets, Vohradsky [4] first employed RNN for the reconstruction of a GRN. Subsequently, Wahde et al. [27] suggested a continuous-time RNN formalisation for model parameter training using *genetic algorithms* (GA) to infer GRNs. Each node in an RNN defines a distinct gene, and the edges show how the genes interact with one another in a regulatory capacity. Each phase of the neural network describes the genetic expression level of all genes ($g_i\ \forall i$), at a specified time t. The genetic expression level of all genes at the previous time point, t', and the weights of their associated connecting edges ($\omega_{i,j}\ \forall j$) with gene g_i determines the expression level of gene g_i, at time point t, where $t = t' + dt$. As a result, the overall regulatory impact of every gene inside a network, on any given gene g_i, can be concisely expressed as follows:

$$g_i = \sum_{j=1}^{N} \omega_{i,j} x_j + \beta_i \tag{6}$$

Vohradsky [4] demonstrated that this can be translated inside the interval $[0,1]$ using a sigmoid function. In this case, β_i represents an external input that can be seen as an response latency parameter. A larger value of this parameter denotes a decrease in the impact of $\omega_{i,j}$ on g_i. The more general equation of RNN is as follows:

$$\tau_i \frac{dx_i\left(t\right)}{dt} = f\left(\sum_{j=1}^{N} \omega_{i,j} x_j\left(t\right) + \beta_i\right) - x_i\left(t\right), \tag{7}$$

where $f\left(\cdot\right)$ is defined as:

$$f\left(x\right) = \frac{1}{1+e^{-x}}$$

The weight matrix, $\omega_{i,j}$ describes a finite-time network that approaches stability if it is symmetrical in character. Time-series datasets can only be obtained in the real-world at discrete time points or intervals, and thus (7) can be recast into a discrete form [28], assuming $\frac{dx(t)}{dt} \approx \frac{\Delta x(t)}{\Delta t} = \frac{x(t+\Delta t)-x(t)}{\Delta t}$. In that case, the expression level of gene g_i at time point $t+\Delta t$, i.e., $x_i\left(t+\Delta t\right)$, can be defined as follows:

$$\begin{aligned} x_i\left(t+\Delta t\right) = {} & \frac{\Delta t}{\tau_i} \cdot \frac{1}{1+\exp\left[-\left(\sum_{j=1}^{N} \omega_{i,j} x_j\left(t\right) + \beta_i\right)\right]} \\ & - \left(1 - \frac{\Delta t}{\tau_i}\right) x_i\left(t\right), \end{aligned} \tag{8}$$

where $\omega_{i,j}$ denotes the connections between the genes. A non-zero value of $\omega_{i,j}$ indicates that there is a connection between the ith and kth genes, and the strength of this interaction is determined by the magnitude of $\omega_{i,j}$. On the other hand, β_i is the external input, which, in this instance, might be seen as a reaction delay parameter. A larger value of this parameter denotes a decrease in the impact of $\omega_{i,j}$ on gene g_i. $\omega_{i,j}$ gives the final predicted interactions between the genes, which needs to be extracted from the temporal expression datasets using metaheuristic algorithms, like *particle swarm optimisation* (PSO) to identify the actual interactions.

E. Particle Swarm Optimisation

The *particle swarm optimisation* or PSO [29], [30] is a population based optimisation algorithm that is inspired by the social behaviour of birds and fish. The goal is often to identify the set of regulatory interactions between genes that best explains observed gene expression data. Applying PSO to GRN inference helps in navigating the vast search space of possible gene interactions efficiently. The algorithm explores different combinations of regulatory interactions to find a set that optimally explains the observed gene expression patterns. The mathematical formulation of PSO is given by:

$$\begin{aligned} v_i\left(it+1\right) = {} & w \cdot v_i\left(it\right) + c_1 \cdot r_1 \cdot \left(pbest_i\left(it\right) - p_i\left(it\right)\right) \\ & + c_2 \cdot r_2 \cdot \left(gbest\left(it\right) - p_i\left(it\right)\right) \end{aligned} \tag{9}$$

$$p_i\left(it+1\right) = p_i\left(it\right) + v_i\left(it+1\right), \tag{10}$$

where it denotes the current generation/iteration, $v_i\,(it+1)$ and $p_i\,(it+1)$ signify the velocity and position of the ith particle in the next (i.e. $(it+1)$th) generation, respectively, $pbest_i\,(it)$ is the best position of the ith particle in the current (i.e., itth) generation, $gbest$ is the best position of the swarm, w is the inertia weight, c_1 and c_2 are the acceleration coefficients, and r_1 and r_2 are random values in the range $[0, 1]$ used to introduce stochasticity. The inertia weight controls the impact that the velocity of a particle in the current generation (i.e., $v_i\,(it)$) has on the velocity of the same particle in the next iteration.

III. Methodology

In this work, we have attempted to reconstruct a GRN from the corresponding temporal expression dataset(s) using a RNN based strategy that is augmented by the concepts of mutual information. We have first extracted a probable network of genetic interactions using this statistical approach. However, this network is not directed and there exist no self-loops, both critical biological aspects of a GRN. To resolve this issue, next, we have implemented the RNN formalism to predict the final GRN, where the RNN model parameters have been trained using PSO.

Here, we have not randomly initialised the solution GRNs for PSO. We have used the GRN obtained by the mutual information based step as the starting point of optimisation for training the RNN model. This essentially provides PSO with a truncated search space that is highly likely to contain the correct network structure. Thus, PSO requires a lesser number of iterative steps to generate a satisfactory solution. Additionally, the quality of the solution also improves compared to those present in the contemporary literature.

Let us consider a GRN comprising N genes that needs to be reconstructed from a time-series gene expression dataset containing T time points. For the ith gene, where $i = 1, 2, \ldots, n$, the temporal expression profile has been converted to a random variable X_i (using existing methods [8]) which can take on any value in $\mathcal{X} = \{0, 1\}$. We know that any biological system like a GRN, by virtue of its adaptive nature, becomes unstable. Hence, in this work, we have computed the entropy associated with each gene in a network to estimate this level of uncertainty or unpredictability associated with it. This measure will help us in providing information about the likelihood of interactions amongst the genes in a GRN. The mathematical equation for computing the entropy, $\mathrm{H}\,(X_i)$, of the ith gene, where $i = 1, 2, \ldots, N$, is given below:

$$\mathrm{H}\,(X_i) = -\sum_{x_i \in \mathcal{X}} \Pr\,(x_i) \log_2 \Pr\,(x_i) \tag{11}$$

$\mathrm{H}\,(X_i)$ is the amount of unpredictability in the expression of the ith gene, and $\Pr\,(x_i)$ is the conditional probability associated with the ith gene. The multi-information of the variables indicates an overall statistical dependency between them. A lower entropy value signifies a more predictable and ordered gene expression pattern, while a higher entropy indicates greater disorder and unpredictability.

Further, mutual information is another useful concept that can be used to show how much information is shared between the adjacent nodes in a network. Thus, mutual information has been used in this work to determine the amount of information shared between the genes in a GRN. The mutual information between any two genes, X_i and X_j, is given by:

$$\mathrm{I}\,(X_i, X_j) = \sum_{x,y \in \mathcal{X}} \Pr\,(x_i, x_j) \log_2 \left(\frac{\Pr\,(x_i, x_j)}{\Pr\,(x_i) \cdot \Pr\,(x_j)} \right), \tag{12}$$

where $\Pr\,(X_i)$ and $\Pr\,(X_j)$ are the marginal distribution of the expression levels of genes i and j, respectively, and $\Pr\,(X_i, X_j)$ is the joint probability distribution between them.

Nevertheless, there arises a major problem while using mutual information for GRN reconstruction. The quantity of information transferred from the ith gene, g_i, to the jth gene, g_j, is denoted by $\mathrm{I}\,(X_i, X_j)$. Similarly, the amount of information conveyed by g_j to g_i is given by $\mathrm{I}\,(X_j, X_i)$. Now, the notion of mutual information states that: $\mathrm{I}\,(X_i, X_j) = \mathrm{I}\,(X_j, X_i)$. However, from a biological perspective, this relation does not always hold in the case of a GRN. In other words, the quantity of information shared between genes g_i and g_j and that between genes g_j and g_i is not always the same. More specifically, if gene g_i regulates gene g_j, it does not necessarily mean that gene g_j will also regulate gene g_i. Even for the few cases where this does happen, the nature and strength of regulation are not exactly the same in both the cases.

To mitigate this issue, in this work, we have selected either $g_i \rightarrow g_j$, i.e., gene g_i regulates gene g_j, or $g_j \rightarrow g_i$, i.e., gene g_j regulates gene g_i, to be present in the network predicted using mutual information. This is crucial as it forms the starting point for the next phase of our proposed methodology, where we have identified the direction and strength of the edges (i.e., genetic regulations) using the RNN formalism and PSO.

Finding the direction and strength of the genetic interactions is critical, as they are crucial factors in a GRN. Here, in this work, we have employed the exponential law based methodology, *recurrent neural network* or RNN, to completely reconstruct a GRN from the temporal expression profiles. The main aim of any reverse engineering framework is to accurately deduce its model parameters such that it can faithfully reproduce the provided time-series dataset. For this purpose, the model parameters need to be trained, and in this work, it has been done using *particle swarm optimisation* or PSO. However, instead of randomly initialising several probable solutions, PSO uses the network structure obtained in the previous step (using mutual information) as the initial solution.

In this way, we have attempted to augment the concept of mutual information with RNN for a more accurate and efficient reconstruction of GRNs from time-series gene expression datasets. The limitations of mutual information based models are: (*i*) they are undirected, i.e., no information regarding the nature and strength of the regulations can be known, and (*ii*) there can be no self-loops, i.e., self-regulations have no way of being identified. These limitations have been mitigated by

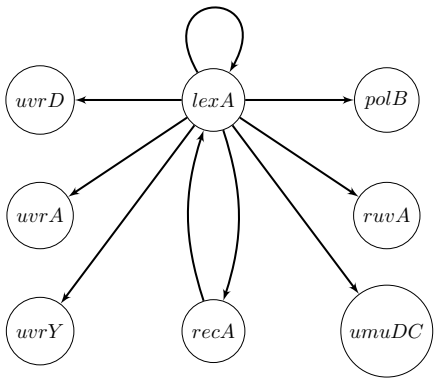

Fig. 1: The original network of the *E. coli* SOS DNA Repair network.

using the RNN framework which can identify the nature and strength of the regulations. Moreover, it can be seen from the contemporary literature that RNN based frameworks usually fare better compared to other power law based methods while reconstructing GRNs.

IV. Result and Discussions

The above methodology has been implemented on the temporal datasets of two networks, one *in vivo* and the other an *in silico* network. The former is the 8-gene SOS DNA repair network of *E. coli* [31], while the latter is a 10-gene network extracted using GeneNetWeaver (GNW) [9]. These networks and the corresponding datasets have been used extensively by researchers in this domain [26], [28], [32], which provides us with an opportunity to carry out a comparative evaluation of the performance of the present investigation with other such methodologies existing in the contemporary literature. Here, The GRNs have been reconstructed using a mutual information based system that helps in reconstructing the connected undirected graph. Next, the RNN formalism has been applied on the same connected undirected graph to investigate the directivity.

From the RNN formalism, we obtain the objective function which we need to optimise (in this case, minimise). And, for the purpose of optimisation, in this work, we have used the one of the most popular swarm based metaheuristic techniques, the *particle swarm optimisation* or PSO. The objective function, in this case, has been chosen as the *mean squared error* (mse), as it has been used by nearly all research works in this domain, and is defined as follows:

$$mse = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \left[x_i\left(t\right) - \tilde{x}_i\left(t\right) \right]^2, \tag{13}$$

where N is the total number of genes in the network, T is the total number of time points, $x_i\left(t\right)$ is the actual expression of the ith gene at time point t, and $\tilde{x}_i\left(t\right)$ is the estimated expression value of the ith gene at time point t.

Experimentation has been carried out on the four datasets of the 8-gene network [31] and one dataset of the 10-gene network, generated using GNW [9]. The actual structure of both the GRNs have been depicted in Figure 1 and Figure 2, respectively. We have assumed a swarm population of 100 and the maximum number of iterations as 1000 in the case of PSO. We have repeated each experiment ten times. The final inferred network has been reconstructed after ensembling the ten networks for each dataset. Here, for ensembling, the threshold, μ, has been considered as 0.9. Another threshold, α, used in the case of mutual information has been considered to be 0.5.

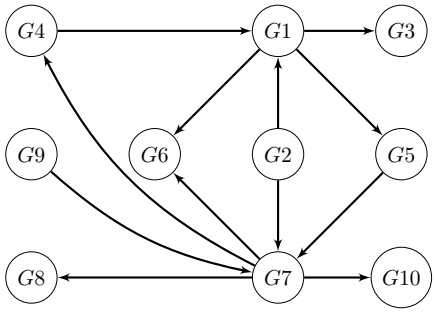

Fig. 2: The original structure of the 10-gene network extracted from GNW [9].

Next, for the selection of edges to be considered, a validity score based selection strategy has been designed. A parameter named, $ps_{i,j}$, has been assigned to provide information about the presence of all the edges in each reconstructed network. In other words, it has been used for the selection of valid edges. The parameter, $ps_{i,j}$, can be defined as:

$$ps_{i,j} = \frac{1}{M} \sum_{m=1}^{M} \omega_{i,j}^{m}, \tag{14}$$

where $\omega_{ij}^{m} \in W^m$ defines the weights of the edges that have been evaluated in the mth simulation. Here, the total number of simulations carried out is $M = 10$, and $ps_{i,j} \in [0, 1]$. After the evaluation of $ps_{i,j}$, $\forall i, j$, the resultant network is constructed using the following condition:

$$g_{ij} = \begin{cases} 1, & \text{if } ps_{ij} \geq \mu, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Here, μ is the threshold corresponding to ps_{ij}, which governs the elimination or retention of edges in the final reconstructed network.

The performance of the tool has been measured with respect to some standard statistical metrics defined by equations (16) through (21), where a true positive is represented by TP, a false positive by FP, a true negative by TN, and a false negative by FN, which in turn are defined as follows: (*i*) TP: $g_{ij}^{or} = g_{ij}^{ob} = 1$, (*ii*) FP: $g_{ij}^{or} = 0$ & $g_{ij}^{ob} = 1$, (*iii*) TN: $g_{ij}^{or} = g_{ij}^{ob} = 0$, and (*iv*) FN: $g_{ij}^{or} = 1$ & $g_{ij}^{ob} = 0$. Here, G^{ob} represents the obtained GRN and G^{or} the original GRN.

$$S_n = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{16}$$

TABLE I: Experimental results for the 8-gene SOS DNA repair network of *E. coli*.

	TP	FP	TN	FN	S_n	S_p	PPV	ACC	F_1
					Dataset 1				
eDSF [32]	3	10	45	6	0.33	0.82	0.23	0.75	0.27
RNN [28]	5	7	48	4	0.56	0.87	0.42	0.83	0.48
HS [26]	5	9	46	4	0.56	0.84	0.36	0.8	0.43
Proposed	5	6	49	4	0.56	0.89	0.45	0.84	0.50
					Dataset 2				
eDSF [32]	8	5	50	1	0.89	0.91	0.62	0.91	0.73
RNN [28]	4	6	49	5	0.44	0.89	0.40	0.83	0.42
HS [26]	4	10	45	5	0.44	0.82	0.29	0.77	0.35
Proposed	5	8	47	4	0.56	0.85	0.38	0.81	0.45
					Dataset 3				
eDSF [32]	4	9	46	5	0.44	0.84	0.31	0.78	0.36
RNN [28]	5	5	50	4	0.56	0.91	0.50	0.86	0.53
HS [26]	5	8	47	4	0.56	0.85	0.38	0.81	0.45
Proposed	6	6	49	3	0.67	0.89	0.50	0.86	0.57
					Dataset 4				
eDSF [32]	0	9	46	9	0	0.84	0	0.72	0
RNN [28]	5	11	44	4	0.56	0.80	0.31	0.77	0.40
HS [26]	3	8	47	6	0.33	0.85	0.27	0.78	0.30
Proposed	6	7	48	3	0.67	0.87	0.46	0.84	0.55

$$S_p = \frac{\text{TN}}{\text{FP} + \text{TN}} \tag{17}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} = 1 - S_p \tag{18}$$

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{19}$$

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{20}$$

$$F_1 = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{21}$$

The performance of the proposed methodology has been compared with existing techniques, like eDSF [32], RNN [28], and HS [26]. The corresponding results have been displayed in Table I and Table II, respectively, for the two networks. The interactions between the genes that are inferred after applying mutual information has been shown in figures 3(a), 4(a), 5(a), and 6(a), respectively for the four datasets of the 8-gene network. Similarly the final inferred network has been shown in figures 3(b), 4(b), 5(b), and 6(b), respectively for the four datasets.

For the 10-gene network extracted for GNW [9], Fig. 7(a) represents the undirected network inferred using mutual information, while 7(b) depicts the final predicted GRN.

It is clear from the evaluated results as well as the comparison, that in most of the cases, the proposed MiRNN technique outperforms the existing techniques. The performance of the proposed tool shows better efficiency in case of comparatively bigger networks. It is worth mentioning here that the number of iterations required for the proposed technique is found to be much lesser than that of the existing methodologies.

TABLE II: Experimental results for the 10-gene network extracted from GNW [9].

	TP	FP	TN	FN	S_n	S_p	PPV	ACC	F_1
eDSF [32]	4	15	81	8	0.33	0.92	0.36	0.85	0.34
RNN [28]	4	12	76	12	0.33	0.86	0.25	0.80	0.29
HS [26]	6	15	73	6	0.50	0.83	0.28	0.79	0.36
Proposed	7	9	78	6	0.54	0.90	0.44	0.85	0.49

V. Conclusion

The computational reconstruction of GRN suffers from many aspects. Firstly, most techniques fail to produce deserved results for bigger networks. Secondly, the computational time increases exponentially as the network size increases. Therefore, there remains enormous scope for the development of suitable techniques that will try to rectify these limitations. The proposed technique, MiRNN, has been developed to resolve the difficulties in the reconstruction of bigger genetic networks. In this investigation, an 8-gene and a 10-gene network have been investigated. The results were found to be quite satisfactory as the number of TPs predicted has improved, while the number of FPs identified has reduced significantly. As a result, accuracy and F_1 also improved satisfactorily in the case of the 8-gene network and especially the 10-gene network. The computational time requirement was found to be comparable with that of existing techniques. Therefore, the proposed technique shows better proficiency in the case of applications in bigger networks. As a future scope of research, the proposed techniques can be studied for much bigger networks like 20-gene, 50-gene, or 100-gene networks.

References

[1] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, 1969.

[2] B.-E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. d'Alché Buc, "Gene networks inference using dynamic bayesian networks," *Bioinformatics-Oxford*, vol. 19, no. 2, pp. 138–148, 2003.

[3] M. Zou and S. D. Conzen, "A new dynamic bayesian network (dbn) approach for identifying gene regulatory networks from time course microarray data," *Bioinformatics*, vol. 21, no. 1, pp. 71–79, 2005.

[4] J. Vohradsky, "Neural model of the genetic network," *Journal of Biological Chemistry*, vol. 276, no. 39, pp. 36 168–36 173, 2001.

[5] E. O. Voit, *Computational analysis of biochemical systems: A practical guide for biochemists and molecular biologists*. Cambridge University Press, 2000.

[6] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts, "Inferring regulatory networks from expression data using tree-based methods," *PloS One*, vol. 5, no. 9, p. e12776, 2010.

[7] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins, and T. S. Gardner, "Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles," *PLoS Biology*, vol. 5, no. 1, p. e8, 2007.

[8] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. D. Favera, and A. Califano, "Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context," in *BMC Bioinformatics*, vol. 7, no. 1. BioMed Central, 2006, pp. 1–15.

[9] T. Schaffter, D. Marbach, and D. Floreano, "Genenetweaver: in silico benchmark generation and performance profiling of network inference methods," *Bioinformatics*, vol. 27, no. 16, pp. 2263–2270, 06 2011. [Online]. Available: https://doi.org/10.1093/bioinformatics/btr373

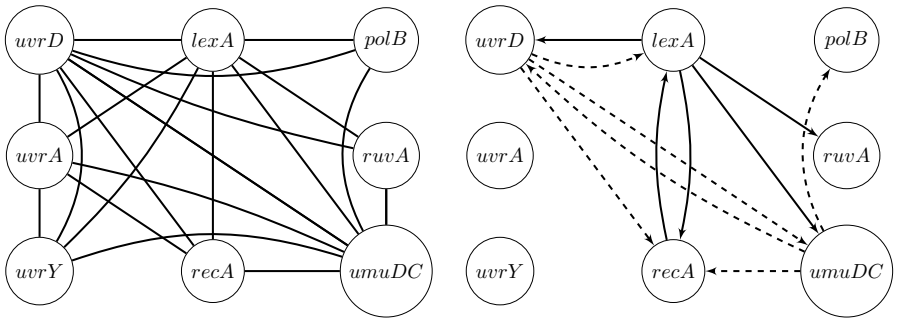

Fig. 3: (a) The predicted (undirected) network of the *E. coli* SOS DNA Repair network using mutual information for Dataset 1. (b) The final predicted network after implementing RNN. Solid lines denote TPs, while dashed lines signify FPs.

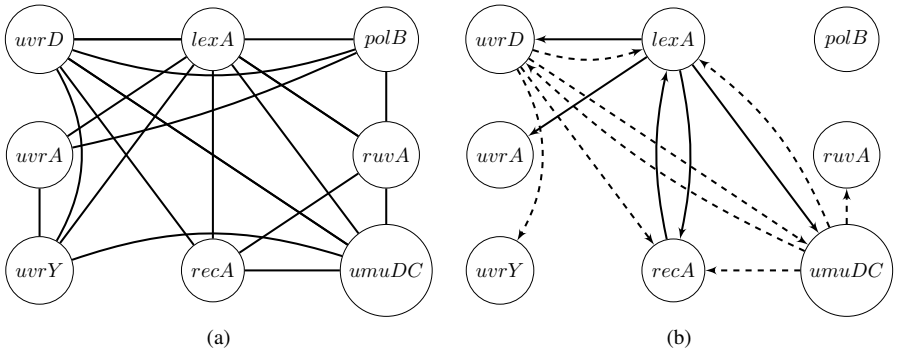

Fig. 4: (a) The predicted (undirected) network of the *E. coli* SOS DNA Repair network using mutual information for Dataset 2. (b) The final predicted network after implementing RNN. Solid lines denote TPs, while dashed lines signify FPs.

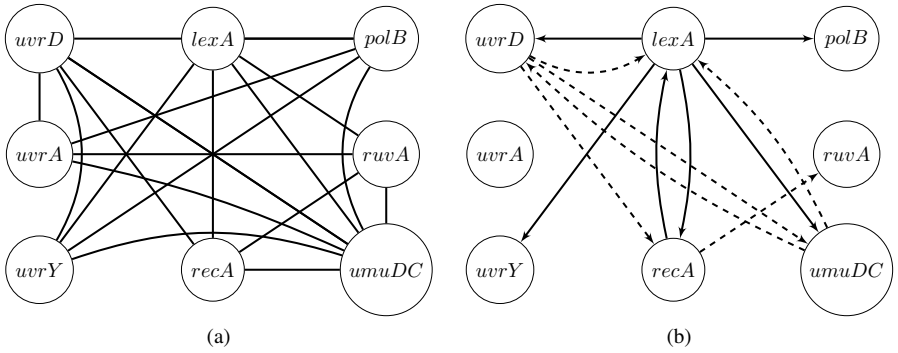

Fig. 5: (a) The predicted (undirected) network of the *E. coli* SOS DNA Repair network using mutual information for Dataset 3. (b) The final predicted network after implementing RNN. Solid lines denote TPs, while dashed lines signify FPs.

[10] H. Lähdesmäki, I. Shmulevich, and O. Yli-Harja, "On learning gene regulatory networks under the boolean network model," *Machine Learning*, vol. 52, pp. 147–167, 2003.

[11] C.-C. Chen and S. Zhong, "Inferring gene regulatory networks by thermodynamic modeling," *BMC Genomics*, vol. 9, no. 2, pp. 1–7, 2008.

[12] W.-P. Lee and W.-S. Tzou, "Computational methods for discovering gene networks from expression data," *Briefings in Bioinformatics*, vol. 10, no. 4, pp. 408–423, 2009.

[13] T. Akutsu, S. Miyano, and S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under the boolean network model," in *Biocomputing'99*. World Scientific, 1999, pp. 17–28.

[14] R. Pal, A. Datta, and E. R. Dougherty, "Optimal infinite-horizon control for probabilistic boolean networks," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2375–2387, 2006.

[15] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[16] W. Zhao, E. Serpedin, and E. R. Dougherty, "Inferring gene regulatory networks from time series data using the minimum description length principle," *Bioinformatics*, vol. 22, no. 17, pp. 2129–2135, 2006.

[17] W.-K. Ching, S. Zhang, M. K. Ng, and T. Akutsu, "An approximation method for solving the steady-state probability distribution of probabilistic boolean networks," *Bioinformatics*, vol. 23, no. 12, pp. 1511–1518,

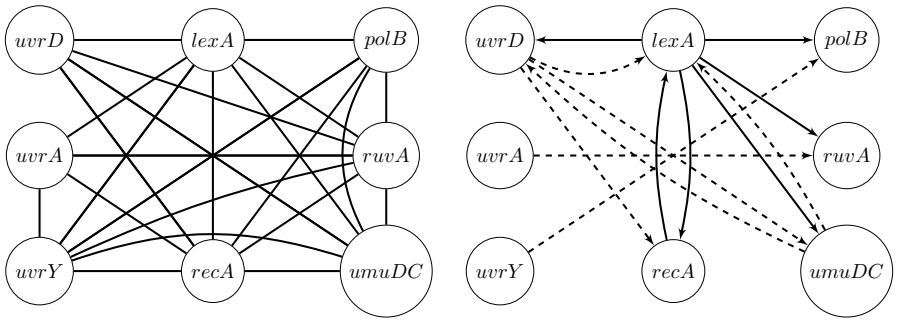

Fig. 6: (a) The predicted (undirected) network of the *E. coli* SOS DNA Repair network using mutual information for Dataset 4. (b) The final predicted network after implementing RNN. Solid lines denote TPs, while dashed lines signify FPs.

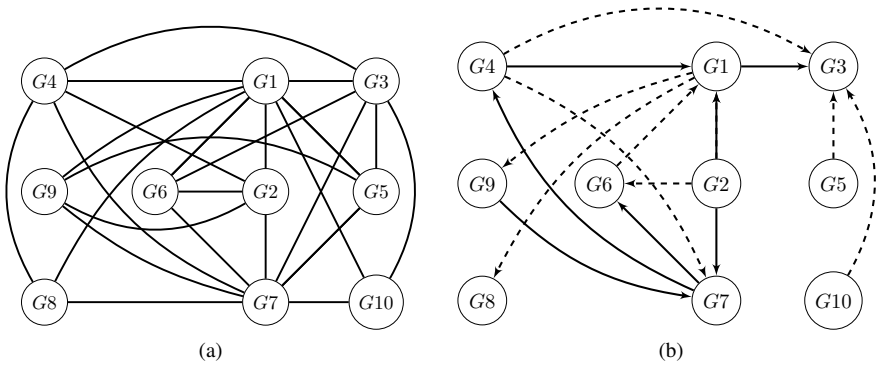

Fig. 7: (a) The predicted structure of the 10-gene network extracted from GNW [9] applying mutual information. (b) Inferred network obtained after applying the proposed method. Solid lines denote TPs, while dashed lines signify FPs.

2007.

[18] V. Filkov, "Identifying gene regulatory networks from gene expression data," *Chapman & Hall/CRC Computer & Information Science Series*, 2005.

[19] C. Manfredotti, "Modeling and inference with relational dynamic bayesian networks," in *Advances in Artificial Intelligence: 22nd Canadian Conference on Artificial Intelligence, Canadian AI 2009 Kelowna, Canada, May 25-27, 2009 Proceedings 22*. Springer, 2009, pp. 287–290.

[20] M. Grzegorczyk and D. Husmeier, "Improvements in the reconstruction of time-varying gene regulatory networks: dynamic programming and regularization by information sharing among genes," *Bioinformatics*, vol. 27, no. 5, pp. 693–699, 2011.

[21] L. E. Chai, M. S. Mohamad, S. Deris, C. K. Chong, Y. W. Choon, Z. Ibrahim, and S. Omatu, "Inferring gene regulatory networks from gene expression data by a dynamic bayesian network-based model," in *Distributed Computing and Artificial Intelligence: 9th International Conference*. Springer, 2012, pp. 379–386.

[22] N. Xuan Vinh, M. Chetty, R. Coppel, and P. P. Wangikar, "Gene regulatory network modeling via global optimization of high-order dynamic bayesian network," *BMC Bioinformatics*, vol. 13, pp. 1–16, 2012.

[23] V. A. Huynh-Thu and P. Geurts, "dynGENIE3: dynamical GENIE3 for the inference of gene networks from time series expression data," *Scientific Reports*, vol. 8, p. 3384, 2018.

[24] M. A. Savageau and E. O. Voit, "Recasting nonlinear differential equations as s-systems: A canonical nonlinear form," *Mathematical Biosciences*, vol. 87, no. 1, pp. 83–115, 1987.

[25] N. Noman and H. Iba, "Reverse engineering genetic networks using evolutionary computation," *Genome Informatics*, vol. 16, no. 2, pp. 205–214, 2005.

[26] A. Khan, G. Saha, and R. K. Pal, "Modified half-system based method for reverse engineering of gene regulatory networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 17, no. 4, pp. 1303–1316, 2020.

[27] M. Wahde and J. Hertz, "Modeling genetic regulatory dynamics in neural development," *Journal of Computational Biology*, vol. 8, no. 4, pp. 429–442, 2001.

[28] A. Khan, S. Mandal, R. K. Pal, G. Saha *et al.*, "Construction of gene regulatory networks using recurrent neural networks and swarm intelligence," *Scientifica*, vol. 2016, 2016.

[29] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. IEEE, 1995, pp. 39–43.

[30] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*. Elsevier, 2001.

[31] M. Ronen, R. Rosenberg, B. I. Shraiman, and U. Alon, "Assigning Numbers to the Arrows: Parameterising a Gene Regulation Network by using Accurate Expression Kinetics," *Proceedings of the National Academy of Sciences*, vol. 99, no. 16, pp. 10 555–10 560, 2002.

[32] K. Kentzoglanakis and M. Poole, "A swarm intelligence framework for reconstructing gene networks: searching for biologically plausible architectures," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 2, pp. 358–371, 2011.