A Hybrid Methodology for the Reverse Engineering of Gene Regulatory Networks

Abhinandan Khan*, Ankita Dutta†, Goutam Saha‡, and Rajat Kumar Pal*
*Department of Computer Science and Engineering, University of Calcutta
Acharya Prafulla Chandra Roy Siksha Prangan, JD-2, Sector-III, Saltlake, Kolkata – 700106, India
Email: {khan.abhinandan, pal.rajatk}@gmail.com
†Machine Intelligence Unit, Indian Statistical Institute, Kolkata
203, Barrackpore Trunk Road, Kolkata – 700108, India
Email: ankitadutta1993@gmail.com
‡Department of Information Technology, North-Eastern Hill University, Shillong
Umshing Mawkynroh, Shillong, Meghalaya – 793022, India
Email: dr.goutamsaha@gmail.com

***Abstract*—In this work, a computational approach has been proposed based on the hybridisation of two modelling formalisms, *recurrent neural networks* and *half-systems*, for the reconstruction of gene regulatory networks from time-series gene expression datasets. To the best of our knowledge, the proposed hybridisation has not been attempted previously in this domain. Here, recurrent neural networks and half-systems have been hybridised to capture the underlying dynamics present in the temporal gene expression profiles. The motivation behind this work is to integrate the advantages of both the techniques in the proposed model such that the problem of reverse engineering of gene regulatory networks can be resolved more efficiently. Artificial bee colony optimisation has been used for the estimation of the model parameters. We have implemented the proposed hybrid methodology on the real-world experimental datasets (*in vivo*) of the SOS DNA Repair network of *Escherichia coli*. The obtained results are comparable to or better than that of other reverse engineering methodologies present in contemporary literature.**

***Index Terms*—artificial bee colony, gene regulatory network, half-system, recurrent neural network, reverse engineering**

I. Introduction

This work proposes a hybrid mathematical formalism for the reconstruction of biologically plausible *gene regulatory networks* (GRNs) from time-series gene expression datasets. Modern technological advancements has enabled us to generate an enormous amount of gene expression data by monitoring the expression levels of thousands of genes simultaneously under a particular condition [1]–[3]. This monitoring process is also known as *gene expression analysis*, and it involves:

- detection of the differences in the expression levels of differentially expressed genes, at a particular time-point (for time-course experiments [3]),
- discovery of the differences in the expression levels of a specific gene over multiple time-points, and
- identification of the similarly expressed genes over numerous time-points.

Acquiring biological data is becoming increasingly easier with technological advancements. Nevertheless, biological information is meaningless until and unless it is analysed and interpreted accurately. Various mathematical models have been developed to serve this purpose, although achieving the desired accuracy is still an open problem for researchers.

The expression values of genes denote the amount of mRNA synthesised by genes during transcription. The value of gene expression is a measure of how active or functional a gene is [3]. mRNAs are translated to proteins, and a protein, or a set of proteins, controls the expression level of other genes. Thus, complex regulatory relationships exist amongst genes that can be characterised using GRNs. These regulations are of two types: (*i*) *activation*: the expression of a target gene is initiated or the expression level increases, and (*ii*) *inhibition* or *repression*: the expression of the target gene stops or the level of expression reduces. Such regulatory relationships are indirect in nature because transcription factors (proteins) act as intermediary players by binding to specific regions in the sequence of a target gene and induce changes in the rate of its protein production. Two well-known difficulties faced during the analysis of temporal expression datasets are the *curse of dimensionality* [4] and *noise* [5].

Various mathematical formalisms have been used for the reverse engineering of GRNs from time-series gene expression datasets. *Ordinary differential equations*, *Bayesian networks* (dynamic and static), *linear additive models*, *recurrent neural networks* (RNNs), *S-systems*, etc., are among the most common techniques used by researchers for this purpose. Reconstruction of GRNs is an *ill-posed* problem, and thus suffers from *over-fitting* [6]. Therefore, a delicate balance is needed between the reduction of prediction error and the actual network topology. Real-world GRNs are sparse in nature [5], [7], [8], i.e. there exist only a few regulators among the genes in a network. However, most of the proposed methodologies have thus far been unable to attain an entirely correct predicted model, even for small-scale, real-world networks. Some of the models have been successful in inferring all the actual regulations, but also include a substantial number of incorrect predictions. Addition of biological information proves to be somewhat useful towards improving the accuracy of the inferred models [9]. However, accurate prediction of large-

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

scale GRNs is still an open problem. The time required for the reconstruction of large-scale GRNs is also huge.

Thus, in the present research endeavour, our motivation is to develop a new computational framework for the reconstruction of GRNs that are more biologically relevant. In this paper, we have proposed a new computational formalism that is based on the hybridisation of RNN and *half-systems* (HS). HS is similar in nature to S-system [10]. However, as the name suggests, HS requires exactly half the number of parameters for modelling a GRN. To the best of our knowledge, the two existing models, RNN and HS, have not been hybridised previously for the extraction of underlying network dynamics from temporal expression datasets. We have employed *artificial bee colony* (ABC) optimisation [11] to train the parameters of our proposed hybrid model.

We have implemented the proposed hybrid methodology for the reconstruction of the *E. coli* SOS DNA Repair network, comprising eight genes, from four *in vivo* datasets [12]. The obtained results clearly demonstrate that the performance of the proposed technique is comparable or better compared to other similar methods present in contemporary literature. Also, for the sake of retaining uniformity, we have implemented the state-of-the-art GRN inference tool, **GENIE3** [13], on the data and calculated the *area under the curve* or **AUC** scores, i.e. the *area under the precision-recall curve*, **AUPR**, and the *area under the receiver operating characteristic curve*, **AUROC**. These form the basis of comparison between our proposed technique and **GENIE3** [13]. The hybrid GRN inference formalism achieved better **AUC** scores than **GENIE3** [13] for all the four cases.

The rest of the paper has been organised as follows: Section II presents an overview of earlier research works in this domain, along with the basics of HS, RNN, and ABC. Section III illustrates the proposed hybridisation of HS and RNN and how it has been implemented for the reconstruction of GRNs from temporal expression data. The experimental results have been presented in Section IV, along with a detailed discussion. Finally, Section V concludes the paper.

II. Preliminaries

A. Scientific Background

In recent years, several techniques have been developed to infer GRNs from time-series gene expression data based on different mathematical formalisms, namely, *ordinary differential equations*, *Bayesian networks*, *Boolean networks*, *linear additive models*, *recurrent neural networks*, *S-systems*, etc. Kim *et al*. [14] proposed a model based on ODEs, whose performance was observed in the presence of noise and time-delay, separately, as well as, in combination. Chen *et al*. [15] proposed a differential equation based model using two different methods: Fourier Transform for Stable Systems (FTSS) and Minimum Weight Solutions to Linear Equations (MWSLE).

Bayesian network (both static and dynamic) is another mathematical formalism that has been widely used in this domain of reverse engineering GRNs. *Dynamic Bayesian networks* (DBN) form the basis of different models proposed by various researchers [16]–[19]. Zou and Conzen [16] described DBN as a mathematical model that can identify regulator-target pairs of genes based on some statistical analysis of their expression relationships over different time-points. Husmeier *et al*. [18] used a Markov Chain Monte Carlo, for Bayesian learning. On the other hand, Friedman *et al*. [20] used a Static Bayesian network to simulate the dynamic behaviour of GRNs in their proposed model.

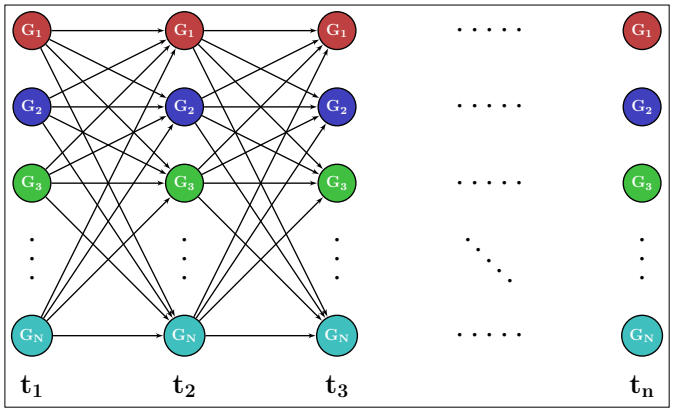


Fig. 1: The representation of a GRN using an RNN model. The network shown is unfolded from $t = t_1$ to $t = t_3$. All possible connections amongst the genes have been shown, whereas, real-world networks are sparse.

Researchers also proposed models based on *Boolean networks* [21]–[23], which is binary in nature. In this formalism, nodes represent genes, and edges denote the regulatory interactions amongst the genes. Genes are considered to be binary devices, which can either be turned on or off under the combined effects of regulator(s). The level of expression of a gene is functionally related to the state of all the regulators by some logical rules [23]. D'haeseleer *et al*. [24] proposed a linear model that considers the different degrees of regulatory effects by incorporating weights to each connection amongst the genes in the network.

B. Recurrent Neural Network and S-System

Recurrent neural networks or RNNs are a special family of an artificial neural networks. RNNs are popular because of their ability to learn from data and robustness to noisy data [25]. RNNs can capture the complex, dynamic, and temporal behaviour of biological organisms. A simple RNN model has been shown in Fig. 1.

Being the basic functional units of cells, genes participate in the signalling and control of all processes necessary for life. Genes encode proteins, one of the fundamental molecular components the living cells are composed of. Protein synthesis from a particular gene is carried out through two essential sequential biological processes, namely, transcription and translation. These two procedures change the state of the cells and influence the expression of other genes. This phenomenon can be modelled very well by a genetic network

of feed-forward type, where all the genes and their product participating in one regulatory event are members of the network. Based on this assumption, Vohradsky [26] proposed a special type of RNN. The mathematical representation of the RNN model is as follows:

$$\tau_i \frac{dx_i}{dt} = \frac{1}{1 + exp\left[-\left\{\sum_{j=1}^{N} w_{ij} x_j + \beta_i\right\}\right]} - x_i, \quad (1)$$

where τ_i is the constant coefficient; Δt is the gap between two consecutive time-points; $x_i\,(t + \Delta t)$ is the expression level of gene i at next time-point, $(t + \Delta t)$; $x_i\,(t)$ is the expression level of gene i at the current time-point, t; w_{ij} is the weight of the edge from a node (gene) j to another node (gene) i; β_i denotes an external input that may be visualised as a reaction delay parameter; and N is the number of genes in the network. Since the number of time-points is limited, we can assume $\frac{dx_i}{dt} \approx \frac{\Delta x_i}{\Delta t} = \frac{x_i(t+\Delta t) - x_i(t)}{\Delta t}$. Thus, we get the following:

$$x_i\,(t + \Delta t) = \frac{\Delta t}{\tau_i} \cdot \frac{1}{1 + exp\left[-\left\{\sum_{j=1}^{N} w_{ij} x_j\,(t) + \beta_i\right\}\right]} + \left(1 - \frac{\Delta t}{\tau_i}\right) \cdot x_i\,(t) \quad (2)$$

The expression value of a particular gene at any time-point can be predicted using (2) from the expression values of the other genes at the previous time-point. The total number of parameters that need to be estimated for each gene in the RNN formalism is $(N + 2)$. Thus, the model requires $N\,(N + 2)$ parameters for a N-gene network.

Xu *et al.* [27] were one of the first to propose an RNN based model for GRN reconstruction, and implemented two variants of the *particle swarm optimisation* (PSO) algorithm, namely, PSO-FIXEW and PSO-RADW, for model parameter training. The authors replaced PSO with *differential evolution* (DE) in one of their future works [28]. First, the authors observed the performance of the model using DE and PSO separately. Then, they employed a hybridised version of DE and PSO, termed as DEPSO. Kentzoglanakis *et al.* [29], on the other hand, used a decoupled strategy. First, the authors implemented the *ant colony optimisation* algorithm to generate biologically plausible candidate architectures by searching the discrete space of network topologies. Next, PSO was used to train the obtained RNN model by examining the continuous space of parameters. Khan *et al.* [30] proposed a *bat algorithm* (BA) inspired version of PSO, named BAPSO, for the reconstruction of GRNs from time-series gene expression datasets.

S-system is a set of nonlinear differential equations of the first order commonly used to reconstruct GRNs [10] from time-series gene expression datasets. The 'S' in the name stands for *saturation* and *synergy*, two essential characteristics of biological systems. S-system is a power-law based formalism and is found to be very suitable for representing nonlinear biological systems like GRNs. The mathematical representation of the S-system model is given as:

$$\frac{dx_i}{dt} = \alpha_i \prod_{j=1}^{N} x_j^{g_{ij}} - \beta_i \prod_{j=1}^{N} x_j^{h_{ij}}, \quad (3)$$

where α_i and β_i are the rate constants for the production and degradation terms, respectively; g_{ij} and h_{ij} are the kinetic orders of the system, also known as the exponential parameters; and N is the number of genes in the network. Again assuming $\frac{dx_i}{dt} \approx \frac{\Delta x_i}{\Delta t} = \frac{x_i(t+\Delta t) - x_i(t)}{\Delta t}$, we get:

$$x_i\,(t + \Delta t) = x_i\,(t) + (\Delta t \cdot \alpha_i) \cdot \prod_{j=1}^{N} [x_j\,(t)]^{g_{ij}} - (\Delta t \cdot \beta_i) \cdot \prod_{j=1}^{N} [x_j\,(t)]^{h_{ij}} \quad (4)$$

The expression value of a particular gene at any time-point can be predicted using (4) from the expression values of the other genes at the previous time-point. The total number of parameters that need to be estimated for each gene in the S-system formalism is $(2N + 2)$. Thus, the model requires $2N\,(N + 1)$ parameters for the case of an N-gene network.

Palafox *et al.* [31] employed *dissipative* PSO or DPSO to optimise the parameter values of a decoupled version of S-system. Juang *et al.* [32] proposed an S-system based model involving a hybrid algorithm for identifying the network structure with minimal connectivity and parameter estimation of the determined network structure. BA was used by Mandal *et al.* [33] for estimating the parameter values of a decoupled and regularised S-system based model. A comprehensive review on various methodologies for the reconstruction of GRNs from time-series gene expression data can be found in [34], [35].

C. Half-System

RNNs are more resistant to noisy data than S-systems. While, on the other hand, S-systems are more suited for modelling the temporal dynamics of genetic expressions because they can model both *synergy* and *saturation*. Hence, we have attempted to combine the strengths of these two models to create a new formalism that would be robust as well as have better biological relevance.

Nevertheless, S-system modelling poses a few problems. It is computationally expensive to train the model due to the larger number of parameters required compared to RNN, i.e. $2N\,(N + 1)$ compared to $N\,(N + 2)$. Also, in (4), $g_{ij} > 0$ or $h_{ij} < 0$ denotes activation of gene i by gene j, while $g_{ij} < 0$ or $h_{ij} > 0$ signifies inhibition of gene i by gene j. However, a problem arises when both the predicted g_{ij} and h_{ij} are of the same sign, which suggests dual regulations. This is unrealistic, as a gene cannot activate as well as inhibit another gene at the same time. This poses a critical issue during network reconstruction from the estimated model parameters. To avoid such issues, we have employed the HS formalism [6], in place

of S-system, and hybridised it with RNN in this work. The mathematical representation of HS is as follows:

$$\frac{dx_i}{dt} = \alpha_i \prod_{j=1}^{N} x_j^{w_{ij}}, \tag{5}$$

where α_i is the single rate constant; w_{ij} is the only kinetic order of the system; and N is the number of genes in the network. Again assuming $\frac{dx_i}{dt} \approx \frac{\Delta x_i}{\Delta t} = \frac{x_i(t+\Delta t)-x_i(t)}{\Delta t}$, we get:

$$x_i\left(t+\Delta t\right) = x_i\left(t\right) + \left(\Delta t \cdot \alpha_i\right) \cdot \prod_{j=1}^{N} \left[x_j\left(t\right)\right]^{w_{ij}} \tag{6}$$

The total number of parameters required for training is thus $(N+1)$. Thus, the model requires $N\left(N+1\right)$ parameters for the case of an N-gene network, which is half of that in S-system. The authors [6] have added two modifications to the traditional HS formalism to improve the stability of the model as well as increase the prediction accuracy. We have used the same in this work.

Firstly, the traditional HS given by (5) does not have any self-degradation term that is critical for its stability. Also, the product of gene expression is not accumulated at the reaction site but is used up in the process of regulating the expression of other genes. The depletion of the expressed product is dependent on the current level of expression of a gene. For all these reasons, the traditional HS formulation given by (5) and (6) has been further modified as given in (7) and (8) by adding a new term $\gamma_i x_i\left(t\right)$, where γ_i is assumed to be positive (>0). In this modified HS formulation, the change in the expression level x_i of gene i, varies with time according to:

$$\frac{dx_i}{dt} = \alpha_i \prod_{j=1}^{N} x_j^{w_{ij}} - \gamma_i \cdot x_i \tag{7}$$

Again assuming $\frac{dx_i}{dt} \approx \frac{\Delta x_i}{\Delta t} = \frac{x_i(t+\Delta t)-x_i(t)}{\Delta t}$, we get:

$$\begin{aligned} x_i\left(t+\Delta t\right) = \left(\Delta t \cdot \alpha_i\right) \cdot \prod_{j=1}^{N} \left[x_j\left(t\right)\right]^{w_{ij}} \\ + \left(1 - \Delta t \cdot \gamma_i\right) \cdot x_i\left(t\right) \end{aligned} \tag{8}$$

The HS model in (8) has been further modified as:

$$\begin{aligned} x_i\left(t+\Delta t\right) = \left(\Delta t \cdot \alpha_i\right) \cdot \prod_{j=1}^{N} \left[x_j\left(t\right)\right]^{w_{ij}} \\ + \left(1 - \Delta t \cdot \gamma_i\right) \cdot x_i\left(t\right) + q_i \cdot df, \end{aligned} \tag{9}$$

where df is the difference between the original and predicted expression values of the associated gene at the last sampling instance; and q_i is a positive constant. The term $q_i \cdot df$, in essence, is an error corrector, or in other words, a biologically-motivated penalty term that has been introduced to prevent the predicted expression profiles from diverging from the original too much. The total number of parameters to be trained for each gene in the modified HS formalism [6] is $(N+3)$, which is still less than S-system.

D. Artificial Bee Colony Optimisation

Artificial Bee Colony (ABC) optimisation was introduced by Karaboga [11], which emulates the collective foraging behaviour of a swarm of honey bees. An artificial bee colony consists of three groups of artificial bees: *employed*, *onlooker*, and *scout* bees. Initially, each employed bee is associated with a single randomly initialised food source leading to the equality of the number of employed bees and food sources. Each food source is evaluated by the fitness (objective) function under consideration to find out their profitability. The food source with the best profitability or nectar amount is memorised. From this point onwards, ABC performs everything iteratively.

$$v^i = x^i + \varphi^i \cdot \left(x^i - x^j\right), \tag{10}$$

where v^i is a new food source in the neighbourhood of the ith food source, x^i, such that $i \neq j$; φ^i is a random number within the range $[-1, 1]$. Next, all the newly explored sources are evaluated and compared with the old ones. If a new source is found to be better, it replaces the earlier one. Each employed bee then computes the probability of its associated food source being selected for exploration by an onlooker bee, and subsequently, shares the information with different onlooker bees. In this paper, the probabilities are computed using (11), as proposed by Babayigit and Ozdemir [36].

$$p^i = exp\left(\frac{-1}{\rho \cdot f^i}\right), \tag{11}$$

where p^i and f^i denote the probability and normalised fitness of the ith food source, respectively; ρ is a control (input) parameter. In this work, $\rho = 1$. In the next step, the onlooker bees search for new food sources in the neighbourhood of the current best source [36] using (12).

$$v^i = x^{best} + p^i \cdot \left(x^{best} - x^j\right), \tag{12}$$

where the current best solution is denoted by x^{best}. Unlike (10), j can be the same as best (the index of the best solution) as the current best may be the final best solution. Each time new food sources are explored, each source is evaluated and compared with the old one. If a source does not improve in quality over a predefined number of iterations, that source is considered as exhausted. The bee associated with the exhausted source becomes a scout bee and randomly generates a new solution. It should be noted that the population size and the minimum number of iterations needed to declare a source as exhausted are two other control parameters of ABC.

III. Methodology

A. The Proposed Hybrid Model

In this section, we have illustrated the proposed hybrid model comprising the two formalisms: *half-system* and *recurrent neural networks*. The proposed hybridisation makes it possible to combine the advantages of both the paradigms in one model. Inherently, HS has all the benefits of S-systems except its stability, which has been rectified by the authors [6].

We have proposed to combine (2) and (9) as $a\times(2)+b\times(9)$, which gives the following:

$$
\begin{aligned}
&(a+b)\cdot x_i\left(t+\Delta t\right)\\
&=a\cdot\left[\frac{\Delta t}{\tau_i}\cdot\frac{1}{1+exp\left(-\left[\sum_{j=1}^{N}w_{ij}x_j\left(t\right)+\beta_i\right]\right)}\right]\\
&+a\cdot\left[\left(1-\frac{\Delta t}{\tau_i}\right)\cdot x_i\left(t\right)\right]+b\cdot\left[\left(\Delta t\cdot\alpha_i\right)\cdot\prod_{j=1}^{N}\left[x_j\left(t\right)\right]^{w_{ij}}\right]\\
&+b\cdot\left[\left(1-\Delta t\cdot\gamma_i\right)\cdot x_i\left(t\right)+q_i\cdot df\right],
\end{aligned}
$$

where $a+b=1$. In this work, we have assumed $a=b=0.5$, i.e. RNN and HS have been given equal weightage. Simplifying the above, we get:

$$
\begin{aligned}
&x_i\left(t+\Delta t\right)\\
&=\frac{a}{a+b}\cdot\frac{\Delta t}{\tau_i}\cdot\frac{1}{1+exp\left[-\left\{\sum_{j=1}^{N}w_{ij}x_j\left(t\right)+\beta_i\right\}\right]}\\
&+\frac{b}{a+b}\cdot\left(\Delta t\cdot\alpha_i\right)\cdot\prod_{j=1}^{N}\left[x_j\left(t\right)\right]^{w_{ij}}+\frac{b}{a+b}\cdot q_i\cdot df\\
&+\left(1-\frac{a}{a+b}\cdot\frac{\Delta t}{\tau_i}-\frac{b}{a+b}\cdot\Delta t\cdot\gamma_i\right)\cdot x_i\left(t\right) \qquad (13)
\end{aligned}
$$

B. Network Reconstruction using the Proposed Model

Researchers have previously [29], [30] employed a decoupled scheme, where the problem of reconstruction of GRNs is divided into two sub-problems: (*i*) search for a suitable and biologically relevant network structure, and (*ii*) proper training of the corresponding HS model parameters.

We have also employed this strategy in this work. Bolouri and Davidson [8] stated that in any genetic network, a gene is likely to be regulated by a maximum of *four* to *eight* genes. We have utilised this biological knowledge in this work. The proposed technique has been implemented on a real-world network comprising eight genes. Since it is a small-scale network, the maximum number of regulators allowed for a gene has been assumed to be *four*. This assumption reduces the discrete search space of candidate network structures and the computational cost, significantly.

Precisely, the search space dimension is reduced from an unconstrained 2^N to $\binom{N}{m}$, where N is the number of genes in a GRN and m is the maximum number of regulators allowed for a gene. We have assumed $m=4$ in this work. This means that the maximum number of parameters to be trained is also reduced significantly. In other words, only four parameters of w_{ij} in (13) can be non-zero and hence need to be estimated. There are several other advantages of this approach as well.

Firstly, all possible combinations of regulators are taken into account, which is expected to maintain the biological reliability of the candidate network topologies to the maximum extent possible. Moreover, over-fitting is also minimised by putting a limit on the number of regulators because it leads to simplification of the proposed model. If no restriction is put on the number of regulators, ABC is likely to use all N values of w_{ij} to train the model defined in (13). This would increase the complexity of the proposed model needlessly, as well as make the network architecture biologically improbable.

Next, the training of the parameters of the proposed hybrid model has been explained. If there are N genes in a network, $N+5$ parameters need to be estimated for each gene according to (13), i.e. w_{ij} (N parameters), α_i, β_i, τ_i, γ_i, and q_i. Thus, the dimension of the optimisation problem becomes $N\left(N+5\right)$. To simplify the training and reduce the computational cost further, researchers [29], [30] have proposed a strategy to decompose this $N\left(N+5\right)$-dimensional problem into N sub-problems of $\left(N+5\right)$ dimensions. Each gene can then be investigated separately, and the corresponding $\left(N+5\right)$ model parameters estimated for each case. Thus, the fitness/objective function (MSE) for ABC has been defined here as:

$$
MSE=\frac{1}{T}\sum_{t=1}^{T}\left[x_i\left(t\right)-\tilde{x}_i\left(t\right)\right]^2, \qquad (14)
$$

where T is the number of available time points; $x_i\left(t\right)$ is the original level of expression of gene i; and $\tilde{x}_i\left(t\right)$ is the predicted expression level of gene i at time-point t.

In the present research endeavour, GRNs have been represented with the help of a directed graph $G=(V,E)$, where V is the set of all nodes (genes) and E is the set containing the edges (the relationships amongst the genes). Here, we have represented G computationally as $G=\left[g_{ij}\right]_{N\times N}$, where N is the number of genes in the network. The value of the element g_{ij} depends on whether an edge exists from node j to node i or not, i.e. $g_{ij}=1$ if gene j regulates gene i; $g_{ij}=0$, otherwise.

Due to the stochastic nature of the ABC technique used for parameter estimation, the predicted network structures are likely to vary with each experiment. As a result, a cooperative training approach has been employed, where K experiments have been performed, and the resultant K GRNs have been stored separately. Next, a selection process has been designed technique based on an inclusion score, is_{ij} allocated to each edge, according to (15):

$$
is_{ij}=\frac{1}{K}\cdot\sum_{k=1}^{K}g_{ij}^{k} \qquad (15)
$$

The final inferred network, $G^F=\left[g_{ij}^f\right]_{N\times N}$ has been generated based on the inclusion score, is_{ij}, according to (16).

$$
g_{ij}^f=\begin{cases}0, & \text{if } is_{ij}<\phi\\ 1, & \text{otherwise,}\end{cases} \qquad (16)
$$

where ϕ is a threshold of the inclusion score, is_{ij}, based on which g_{ij}^f is either assigned a 0 or a 1, i.e. an edge is either included or discarded from the final inferred topology.

TABLE I: Comparison of results obtained by the proposed methodology for the *E. Coli* SOS DNA Repair network [12] with other results present in the contemporary literature.

	TP	FP	$\mathbf{S_n}$	$\mathbf{S_p}$	PPV	ACC	$\mathbf{F_1}$	TP	FP	$\mathbf{S_n}$	$\mathbf{S_p}$	PPV	ACC	$\mathbf{F_1}$
	Dataset 1							**Dataset 2**						
eDSF [29]	3	10	0.33	0.82	0.23	**0.99**	0.27	**8**	**5**	**0.89**	**0.91**	**0.62**	0.98	**0.73**
Khan *et al.* (RNN) [30]	**7**	9	**0.78**	0.84	**0.44**	0.98	**0.56**	7	10	0.78	0.82	0.41	**0.99**	0.54
Khan *et al.* (HS) [6]	5	**7**	0.56	**0.87**	0.42	0.98	0.48	4	6	0.44	0.89	0.40	0.98	0.42
Proposed HS+RNN	5	13	0.56	0.76	0.28	0.73	0.37	7	9	0.78	0.84	0.44	0.98	0.56
	Dataset 3							**Dataset 4**						
eDSF [29]	4	10	0.44	0.82	0.29	**0.99**	0.35	0	9	0.00	**0.84**	0.00	0.98	0.00
Khan *et al.* (RNN) [30]	7	15	0.78	0.73	0.32	**0.99**	0.45	**4**	12	0.44	0.78	0.25	**0.99**	0.32
Khan *et al.* (HS) [6]	5	**5**	0.56	**0.91**	**0.50**	0.98	0.53	5	11	**0.56**	0.80	**0.31**	**0.99**	**0.40**
Proposed HS+RNN	**9**	11	**1.00**	0.80	0.45	**0.99**	**0.62**	**4**	**9**	0.44	**0.84**	**0.31**	0.98	0.36

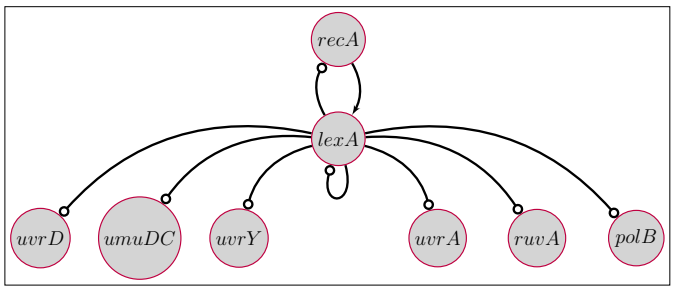


Fig. 2: The original topology of the *E. coli* SOS DNA Repair network [35]. The arrowheads represent activation, and the T-heads denote inhibition.

IV. Experimental Results and Discussions

In this work, we have applied the proposed method based on the hybridisation of RNN and HS for the reverse engineering of the *E. coli* SOS DNA Repair network [12] from four experimental (*in vivo*), time-series gene expression datasets. The obtained results have been compared with those achieved by other such network identification techniques present in the contemporary literature. The comparison has been made based on the following metrics:

- the *true positive rate* $(\mathbf{TPR})$, *sensitivity*, or *recall*;
- the *true negative rate* $(\mathbf{TNR})$ or *specificity*;
- the *positive predictive value* $(\mathbf{PPV})$ or *precision*;
- the *accuracy* $(\mathbf{ACC})$; and
- the *F-score* $(\mathbf{F}_1)$;

which can be mathematically defined as follows:

$$\mathbf{TPR}\,(\mathbf{S_n}) = \frac{\mathbf{TP}}{\mathbf{TP}+\mathbf{FN}}, \tag{17}$$

$$\mathbf{TNR}\,(\mathbf{S_p}) = \frac{\mathbf{TN}}{\mathbf{FP}+\mathbf{TN}}, \tag{18}$$

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

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

$$\mathbf{F}_1 = \frac{2\mathbf{TP}}{2\mathbf{TP}+\mathbf{FP}+\mathbf{FN}}, \tag{21}$$

where $\mathbf{TP}$ stands for true positives, i.e. the number of existing edges that have been identified correctly, while $\mathbf{FP}$ signifies the false positives, i.e. the number of non-existent edges, which have been inferred in accurately. Likewise, $\mathbf{TN}$ stands for true negatives, i.e. the number of non-existent edges, which have been inferred correctly, whereas $\mathbf{FN}$ stands for false negatives, which is the number of existing edges that have been predicted erroneously.

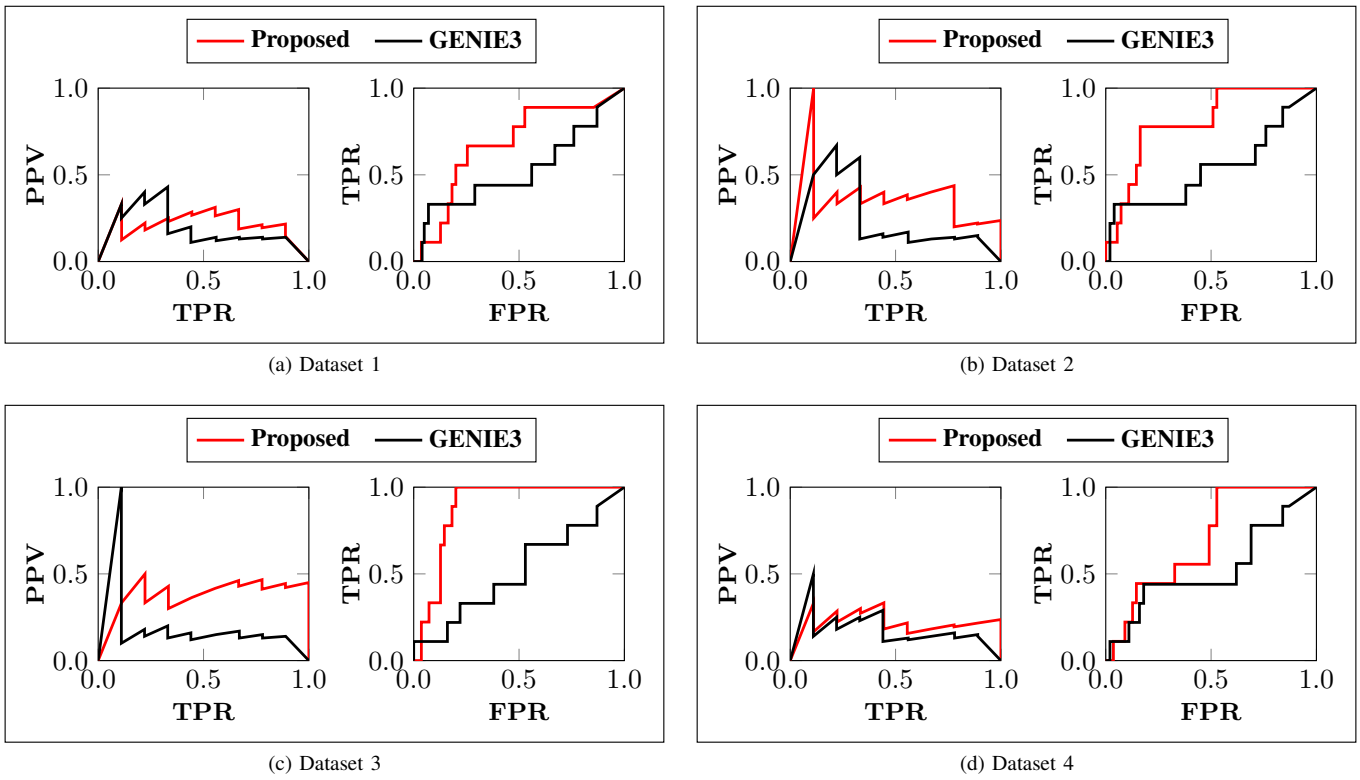
The GRN under investigation, in this work, is the SOS DNA Repair network of the bacterium *Escherichia Coli* [12], shown in Fig. 2. It is usually considered as a benchmark for GRN reconstruction strategies present in contemporary literature. Ronen *et al.* [12] studied this network comprising eight genes, namely, $recA$, $lexA$, $uvrA$, $uvrD$, $uvrY$, $umuDC$, $ruvA$, and $polB$, and analysed their temporal expression levels. Four experimental setups were prepared by the authors, and in each such experiment, they noted the expression levels of all the genes for 50 instances at a temporal resolution of six minutes. These generated datasets are amongst the most useful ones for research on the computational modelling of reverse engineering methodologies that can infer biologically plausible GRNs from temporal expression data. The datasets are freely available at http//wws.weizmann.ac.il/mcb/UriAlon/sites/mcb.UriAlon/files/uploads/DownloadableData/sosdata.zip.

The gene expression values in the datasets mentioned above have been normalised in the range $[0, 1]$ in this work. Also, we have removed the expression value of each gene at the first time point from each dataset, as they are all zero. Kentzoglanakis and Poole *et al.* [29] were one of the early researchers to present their experimental results for the individual datasets separately. Subsequently, other authors [6], [30] have also given their results for the individual datasets. Here, we have followed the same settings as these research works, in setting the inclusion score threshold, $\phi = 0.9$. The comparison of results has been presented in Table I.

It can be clearly seen from Table I that the performance of the proposed hybridised methodology is comparable to or better than the other techniques used for comparison. The proposed method is better than eDSF [29], except in the case of

TABLE II: Comparison of the **AUC** scores achieved by the proposed hybrid technique with those obtained by **GENIE3** [13] for the *E. coli* SOS DNA Repair network [12].

	Dataset 1		Dataset 2		Dataset 3		Dataset 4	
	Proposed	**GENIE3**	**Proposed**	**GENIE3**	**Proposed**	**GENIE3**	**Proposed**	**GENIE3**
AUPR	**0.2071**	0.1831	**0.3518**	0.2386	**0.3818**	0.1772	**0.2168**	0.1690
AUROC	**0.6787**	0.5263	**0.8061**	0.5384	**0.8828**	0.5162	**0.6929**	0.5283



(a) Dataset 1

(b) Dataset 2

(c) Dataset 3

(d) Dataset 4

Fig. 3: Comparison of the proposed hybrid methodology with **GENIE3** [13] based on Precision Recall (PR) and Receiver Operating Characteristic curves for the *E. coli* SOS DNA Repair network [12].

Dataset 2, w.r.t the number of **TP**s, and thus has much better **TPR** values. Similarly, the proposed technique can identify the same or a greater number of **TP**s compared to the RNN based technique [30], except for Dataset 1, and the HS based technique [6], except for Dataset 4. Furthermore, for Dataset 3, the proposed technique can infer all the genetic relationships, i.e. 9 **TP**s, with the highest possible a $\mathbf{TPR} = 1$, which is one of the best results obtained by GRN inference strategies proposed till date.

In addition to the above-mentioned results, we have also implemented **GENIE3** to compare our proposed methodology with a state-of-the-art tool. We have compared the performance of our framework with **GENIE3** based on the area under the receiver operating characteristic curve (**AUROC**) and the area under the precision-recall curve (**AUPR**). The area under the curve or **AUC** scores has been shown in Table II. The corresponding curves have also been presented in Fig.3 for all the datasets, separately.

V. Conclusion

In this work, we have investigated the reconstruction of GRNs from temporal expression profile. For this, we have developed a hybrid framework combining the characteristics of two existing techniques: RNN and HS. The model parameters have been estimated using ABC. The proposed methodology has been used to reconstruct the *E. coli* SOS DNA Repair network [12] from four experimental datasets. The obtained results clearly show that the hybrid technique is comparable to or better than the other network identification techniques [29], [30] for almost all cases, from the point of view of **TP**s.

We have also implemented the state-of-the-art tool, **GENIE3** [11], to maintain uniformity. The proposed technique also achieves better AUC scores compared to **GENIE3** [11]. We have also implemented a restriction on the maximum number of regulators allowed for a gene in a network, based on biological information prevalent in the domain. This helps

in reducing the computational cost and also minimising overfitting. There is scope for future research on the proposed methodology for its implementation in the reverse engineering of large-scale genetic networks.

ACKNOWLEDGEMENT

The authors would like to acknowledge the contribution of the Senior Research Fellowship (NET), awarded by the Council of Scientific & Industrial Research (CSIR), India to the corresponding author (Award No.: 09/028(0974)/2015-EMR-I).

REFERENCES

[1] M. M. Babu, "Introduction to Microarray Data Analysis," *Computational Genomics: Theory and Application*, vol. 225, p. 249, 2004.

[2] G. J. McLachlan, K.-A. Do, and C. Ambroise, *Analysing Microarray Gene Expression Data*. John Wiley & Sons, 2005, vol. 422.

[3] S. Aluru, *Handbook of Computational Molecular Biology*. Chapman and Hall/CRC, 2005.

[4] D. L. Donoho, "High-dimensional Data Analysis: The Curses and Blessings of Dimensionality," *AMS Math Challenges Lecture*, vol. 1, no. 2000, p. 32, 2000.

[5] E. v. Someren, L. Wessels, E. Backer, and M. Reinders, "Genetic Network Modelling," *Pharmacogenomics*, vol. 3, no. 4, pp. 507–525, 2002, pMID: 12164774.

[6] 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*, 2019.

[7] P. D'haeseleer, "Reconstructing Gene Networks from Large-Scale Gene Expression Data," Ph.D. dissertation, University of New Mexico Albuquerque, 2000.

[8] H. Bolouri and E. H. Davidson, "Modelling Transcriptional Regulatory Networks," *BioEssays*, vol. 24, no. 12, pp. 1118–1129, 2002.

[9] A. R. Chowdhury and M. Chetty, *Reconstruction of Large-Scale Gene Regulatory Network using S-System Model*. John Wiley & Sons, Ltd, 2016, ch. 8, pp. 185–210.

[10] 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.

[11] D. Karaboga, "An Idea based on Honey Bee Swarm for Numerical Optimisation," Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, Tech. Rep., 2005.

[12] 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.

[13] A. I. Vân Anh Huynh-Thu, L. Wehenkel, and P. Geurts, "Inferring Regulatory Networks from Expression Data using Tree based Methods," *PloS One*, vol. 5, no. 9, 2010.

[14] S. Kim, J. Kim, and K.-H. Cho, "Inferring Gene Regulatory Networks from Temporal Expression Profiles under Time-Delay and Noise," *Computational Biology and Chemistry*, vol. 31, no. 4, pp. 239–245, 2007.

[15] T. Chen, H. L. He, and G. M. Church, "Modelling Gene Expression with Differential Equations," in *Pacific Symposium on Biocomputing*, 1999, pp. 29–40.

[16] 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.

[17] J. Yu, V. A. Smith, P. P. Wang, A. J. Hartemink, and E. D. Jarvis, "Advances to Bayesian Network Inference for Generating Causal Networks from Observational Biological Data," *Bioinformatics*, vol. 20, no. 18, pp. 3594–3603, 2004.

[18] D. Husmeier, "Sensitivity and Specificity of Inferring Genetic Regulatory Interactions from Microarray Experiments with Dynamic Bayesian Networks," *Bioinformatics*, vol. 19, no. 17, pp. 2271–2282, 2003.

[19] C. Rangel, J. Angus, Z. Ghahramani, M. Lioumi, E. Sotheran, A. Gaiba, D. L. Wild, and F. Falciani, "Modelling T-Cell Activation using Gene Expression Profiling and State-Space Models," *Bioinformatics*, vol. 20, no. 9, pp. 1361–1372, 2004.

[20] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian Networks to Analyse Expression Data," *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 601–620, 2000.

[21] S. Liang, S. Fuhrman, and R. Somogyi, "REVEAL, A General Reverse Engineering Algorithm for Inference of Genetic Network Architectures," in *Pacific Symposium on Biocomputing*, 1998, pp. 18–29.

[22] S. A. Kauffman, "Metabolic Stability and Epigenesis in Randomly Constructed Genetic Nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, 1969.

[23] 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.

[24] P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Linear Modelling of mRNA Expression Levels during CNS Development and Injury," in *Pacific Symposium on Biocomputing*, 1999, pp. 41–52.

[25] K. Raza and M. Alam, "Recurrent Neural Network based Hybrid Model for Reconstructing Gene Regulatory Network," *Computational Biology and Chemistry*, vol. 64, pp. 322–334, 2016.

[26] J. Vohradsky, "Neural Model of the Genetic Network," *Journal of Biological Chemistry*, vol. 276, no. 39, pp. 36 168–36 173, 2001.

[27] R. Xu, D. Wunsch II, and R. Frank, "Inference of Genetic Regulatory Networks with Recurrent Neural Network Models using Particle Swarm Optimisation," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 4, pp. 681–692, 2007.

[28] R. Xu, G. K. Venayagamoorthy, and D. C. Wunsch II, "Modelling of Gene Regulatory Networks with Hybrid Differential Evolution and Particle Swarm Optimisation," *Neural Networks*, vol. 20, no. 8, pp. 917–927, 2007.

[29] 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.

[30] A. Khan, S. Mandal, R. K. Pal, and G. Saha, "Construction of Gene Regulatory Networks using Recurrent Neural Networks and Swarm Intelligence," *Scientifica*, vol. 2016, no. 1060843, pp. 1–14, 2016.

[31] L. Palafox, N. Noman, and H. Iba, "Reverse Engineering of Gene Regulatory Networks using Dissipative Particle Swarm Optimisation," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 577–587, 2012.

[32] J.-N. Juang, S. J. Shiau, and W. Wu, "A Hybrid Parameter Estimation Algorithm for S-System Model of Gene Regulatory Networks," *The Journal of the Astronautical Sciences*, vol. 60, no. 3-4, pp. 559–576, 2013.

[33] S. Mandal, A. Khan, G. Saha, and R. K. Pal, "Reverse Engineering of Gene Regulatory Networks based on S-Systems and Bat Algorithm," *Journal of Bioinformatics and Computational Biology*, vol. 14, no. 03, p. 1650010, 2016.

[34] L. E. Chai, S. K. Loh, S. T. Low, M. S. Mohamad, S. Deris, and Z. Zakaria, "A Review on the Computational Approaches for Gene Regulatory Network Construction," *Computers in Biology and Medicine*, vol. 48, pp. 55–65, 2014.

[35] N. A. Kiani, H. Zenil, J. Olczak, and J. Tegnér, "Evaluating Network Inference Methods in terms of Their Ability to Preserve the Topology and Complexity of Genetic Networks," *Seminars in Cell & Developmental Biology*, vol. 51, no. Supplement C, pp. 44–52, 2016.

[36] B. Babayigit and R. Ozdemir, "A Modified Artificial Bee Colony Algorithm for Numerical Function Optimization," in *2012 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2012, pp. 245–249.