Modified Half-System based Method for Reverse Engineering of Gene Regulatory Networks

Abhinandan Khan, *Student Member, IEEE,* Goutam Saha, and Rajat Kumar Pal, *Member, IEEE*

Abstract—The accurate reconstruction of gene regulatory networks for proper understanding of the intricacies of complex biological mechanisms still provides motivation for researchers. Due to accessibility of various gene expression data, we can now attempt to computationally infer genetic interactions. Among the established network inference techniques, S-system is preferred because of its efficiency in replicating biological systems though it is computationally more expensive. This provides motivation for us to develop a similar system with lesser computational load. In this work, we have proposed a novel methodology for reverse engineering of gene regulatory networks based on a new technique: *half-system*. Half-systems use half the number of parameters compared to S-systems and thus significantly reduce the computational complexity. We have implemented our proposed technique for reconstructing four benchmark networks from their corresponding temporal expression profiles: an 8-gene, a 10-gene, and two 20-gene networks. Being a new technique, to the best of our knowledge, there are no comparable results for this in the contemporary literature. Therefore, we have compared our results with those obtained from the contemporary literature using other methodologies, including the state-of-the-art method, **GENIE3**. The results obtained in this work stack favourably against the competition, even showing quantifiable improvements in some cases.

Index Terms—gene expression data, gene regulatory networks, half-systems, nonlinear ordinary differential equations, particle swarm optimisation.

1 Introduction

Genes are the basic functional units of all living organisms, from unicellular organisms like bacteria to complex multi-cellular organisms like human beings. All activities like the growth and development of organs, onset of diseases and their corresponding resistance, immunity, etc. that transpire inside a cell are the complex regulatory effects of different genes. Proteins are the end products of gene expression, and genes communicate amongst themselves via proteins. A gene regulatory network (GRN) is a graphical representation of these complex regulatory interactions amongst genes. GRNs are usually denoted by directed graphs, where the genes are specified by nodes, and the regulations between the genes are represented by edges [1]. There are two types of genetic regulations, activation and repression, depending on whether a gene (or a group of genes) instructs another gene (or a group of genes) to start (or increase) expression or cease (or decrease) expression, respectively.

The analysis of GRNs has emerged as very crucial in Molecular Biology as it may hold the key to discovering the reasons behind a disease, and its subsequent treatment. However, the reverse engineering of GRNs is a complex task. With the advancements in genetic research technologies, a surprising amount of gene expression data like microarray data [2] has been generated. Regardless of some limitations, the simplicity of current gene expression profiling technologies allows large-scale simultaneous measurements of genetic expression values [3], [4], [5]. This has made it possible for researchers to study the dynamic behaviour [6] and interactions among different genes that are crucial for the elucidation of primary cellular activities, characterising genetic functions, diagnosis of diseases, and assessing drug effects [7], [8], etc.

The size and the variety of the currently available data provide the primary motivation for researchers universally to study and develop computational tools for the analysis of this data. The necessary consequence of this analysis has resulted in the development of valuable tools for the reverse engineering of GRNs, finding of culprit gene(s) for a particular disease, etc. The prominent methodologies are Boolean Networks, Bayesian Networks, Information Theory based approaches, Regression based approaches, Recurrent Neural Networks, and S-systems. One may refer to Chai *et al.* [9] and Kaini *et al.* [10] for a comprehensive review of the various computational approaches for the reverse engineering of GRNs. Some of them have been highlighted in the next section.

The proposed techniques for the reconstruction of real-world GRNs suffer from the problem of overfitting. Therefore, a precise balance between the accuracy of the predicted expression profiles and the actual network structure is required. Also, real-life GRNs are sparse [11], [12], [13], [14], i.e. there are only a handful of regulators among all the related genes. Nevertheless, most of these proposed methodologies are yet to achieve an entirely accurate prediction capability, which is lacking even for small-scale real-world GRNs. Some of the techniques are successful in finding out all the true regulations, but they are prone to inferring a significant number of false connections. Adding partial biological information has been apparently helpful

- *Abhinandan Khan and Rajat Kumar Pal are with the Department of Computer Science and Engineering, University of Calcutta, Acharya Prafulla Chandra Roy Siksha Prangan, JD-2, Sector-III, Saltlake, Kolkata - 700106, India.*
  E-mail: khan.abhinandan@gmail.com
- *Goutam Saha is with the Department of Information Technology, North Eastern Hill University, Shillong - 793022, Meghalaya, India.*

Manuscript received September 28, 2017.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2892450, IEEE/ACM Transactions on Computational Biology and Bioinformatics

in improving the accuracy of the predicted models [15], but the prediction of large-scale GRNs are still inaccurate. The time required for the inference of large-scale networks is also immense.

This is the primary motivation behind our current research endeavour, where we have proposed a novel technique based on a modified half-system [16], [17]. To the best of our knowledge, GRN inference using half-systems has been implemented for the very first time in this research paper. It has been derived and modelled in a similar manner as that of the S-system based work by the authors [16], but employing exactly half the number of model parameters, hence the name. Savageau *et al.* [16] opined that while "mathematically very interesting, they are generally inconvenient for the analysis of biochemical systems." Yet, a little investigation reveals that the half-system formalism is very suitable for the problem of reverse engineering of GRNs. However, it is quite surprising that no other researcher has ever attempted to implement half-systems for the GRN inference problem in the next two decades.

Nevertheless, we have implemented our proposed methodology based on modified half-systems for inferring GRNs from temporal gene expression profiles, and we have performed four experiments. The parameter training has been achieved by a hybrid swarm intelligence algorithm, namely, a bat algorithm inspired particle swarm optimisation (BAPSO) algorithm, developed in [18] and used in [19]. The results obtained using the modified half-system formalism from each experiment is quite encouraging. To retain uniformity, we have also implemented **GENIE3** [20] on all the datasets and compared the AUPR (Area Under Precision-Recall curve) and AUROC (Area Under Receiver Operating Characteristic curve) scores with those obtained by our proposed technique. Our proposed half-system based GRN reconstruction framework consistently obtained better AUC scores than **GENIE3** [20]. Furthermore, we have performed two-fold cross-validation to prove the generalisation property of the proposed half-system based model.

The rest of the paper has been organised as follows. A brief overview of the various approaches for GRN reconstruction from time-series gene expression datasets has been given in Section 2, along with a background of the proposed half-system formalism and BAPSO. The section concludes with the detailed explanation of the proposed methodology. The experimental results have been presented in Section 3, along with the relevant analyses and discussions. The paper concludes with Section 4.

2 Preliminaries and Methods

2.1 Background

Over the years, GRNs have been reconstructed using a variety of models, viz., Bayesian networks, Boolean networks, information theory based approaches, regression based approaches, recurrent neural networks, and S-systems.

Boolean networks (BoNs), one of first models to be implemented, are simple and easy to simulate and thus quite popular for portraying the global dynamics of GRNs and complex genetic interactions [21]. BoNs can portray various biological occurrences like oscillations, multistationarity, switch-like behaviour, stability, and hysteresis [22]. There are two types of BoN based methodologies present in the contemporary literature, namely, *correlation* and *inferring* [23]. Akutsu *et al*. [24] were one of the first group of researchers to implement BoNs for reverse engineering of GRNs, and were subsequently followed by several other prominent works on BoN based GRN modelling [25], [26], [27], [28], [29], [30]. BoNs can analyse large GRNs and supervised learning in the binary domain is well established. Nevertheless, BoNs are deterministic in nature, incapable of handling incomplete expression data, and have a high computational time requirement. As a result, Probabilistic Boolean Networks were developed [31], [32], [33], [34].

Bayesian networks (BaNs), on the other hand, represent the qualitative properties of GRNs using mathematical concepts of probability and graph theory [35], and visualise the independent genetic interactions using directed acyclic graphs (DAG). Friedman *et al*. [36] were one of the first to implement BaNs for GRN modelling, closely followed and improved upon by Peer *et al*. [37]. One may refer to [38], [39], [40], [41] for further works on BaN based GRN modelling. BaNs can handle noise and uncertainty and integrate prior knowledge to reinforce the causal relationships amongst genes but fail to capture the underlying dynamics in temporal expression data. BaNs are time-consuming, cannot handle large-scale networks due to their high computational complexity, and cannot infer self-regulations and feedback loops, a critical attribute for modelling GRNs. To resolve this issue and also incorporate the dynamic or temporal nature of gene expression, Dynamic Bayesian Networks were proposed [42], [43], [44], [45].

Two other subcategories of GRN modelling approaches are mutual information [46], [47] based and regression based [20], [48]. The former infers regulatory relationships amongst genes based on the pairwise mutual information, has low computational complexity, and requires smaller number of samples. On the other hand, **GENIE3** [20], which is a regression based approach, has been implemented in this work to present a uniform comparison of the performance of our proposed technique. **GENIE3** [20] decomposes the problem of inferring a p-gene network into p different regression problems. In each subproblem, the expression profile of one gene is predicted from the expression profiles of all the other genes by means of tree-based ensemble methods, viz., Random Forests or Extra-Trees.

Last but not the least, recurrent neural networks (RNN) and S-systems are two major formalisms, which have been used for the reconstruction of GRNs most recently. Reverse engineering of GRNs from temporal expression data using these two methodologies becomes a two-fold process: (*i*) establishing the network topology/structure and (*ii*) fine-tuning the corresponding model parameters. In this regard, the research works undertaken thus far may be broadly categorised into two groups, christened as *Global Structure Approach* and *Substructure Approach* by Biswas *et al*. [49]. RNN was formally adapted for the reconstruction of GRNs by Vohradsky [50]. Subsequently, Wahde *et al*. proposed a continuous-time RNN formalism to infer GRNs using genetic algorithm (GA) for model parameter training [51], [52]. Xu *et al*. [53], [54] proposed a global structure approach.

The authors used particle swarm optimisation (PSO) to train the RNN model parameters in [53], and PSO, differential equation (DE), and a DE-PSO hybrid algorithm in [54]. Zhang *et al.* [55] and Maraziotis *et al.* [56], also implemented PSO for RNN model parameter estimation. In addition to PSO, Kentzoglanakis *et al.* [57] formally presented a decoupled strategy for GRN inference and used a graph-theoretic method along with ant colony optimisation (ACO) for finding the network structure first. Recently, a new metaheuristic, namely, elephant swarm water search (ESWS) algorithm was proposed by Mandal [58] for training the RNN model parameters [59].

Although RNN is more robust compared to S-systems, the latter is more suited for modelling the temporal dynamics of genetic expressions as it can represent both *synergy* and *saturation* – two critical characteristics of real-world biological systems [16], [17]. A review on several S-system based approaches to GRN reconstruction was given by Voit [60]. Noman *et al.* [61] implemented a decoupled S-system formalism to infer GRNs using an extension of DE, namely trigonometric differential evolution (TDE), for the S-system model parameter optimisation. Liu *et al.* [62] posed the GRN inference problem using S-systems as a multi-objective optimisation problem, and proposed to solve it using the ϵ-constrained method. Vilela *et al.* [63], on the other hand, proposed to solve the S-system equations using a method of eigenvector optimisation of a matrix that is created from the multiple regression equations of the linearised decoupled S-system. Nakayama *et al.* [64] used the immune algorithm (IA) for model parameter training, while Palafox *et al.* [65] used a dissipative particle swarm optimisation (DPSO) algorithm with an L1-regularizer and the Runge-Kutta numerical solver. Last but not the least, Mandal *et al.* [66] implemented bat algorithm (BA) for model training. A slightly different S-system based approach was proposed by Thomas *et al.* [67]. This method is based on the concentrations of both mRNA and proteins. In this S-system formalism proposed, it is assumed that the rate of change in the concentration of any mRNA depends not only on the amount of the given mRNA but also on the amount of proteins. It was also assumed that the rate of change of concentration of the proteins depends on the concentrations of both mRNA and proteins.

It is well-known that reverse engineering of GRNs is an ill-posed problem and so suffers from the problem of overfitting. There exist several techniques to prevent/minimise this problem, viz., cross-validation, regularisation, model architecture simplification, early stopping, etc. The prominent regularization techniques are L1, L2, and dropout. L1-regularizers are generally used for sparse systems that include GRNs. In the domain of reverse engineering of GRNs from temporal expression data, there are some examples where L1-regularizers have been employed by researchers [1], [61], [65], [66], [68], [69]. However, almost all such instances are research works based on the S-system formalism. And surprisingly, RNN based techniques used by researchers for the reconstruction of GRNs have not employed regularisation. They have used either available biological information for model architecture simplification or a metaheuristic to search for a simplified model a priori.

Next, we have introduced the modified half-system based framework proposed in this work. Subsequently, we have briefly exposed the proposed bat algorithm inspired particle swarm optimisation (BAPSO) strategy. Finally, we have presented the proposed methodology in detail.

2.2 Half-Systems

The mathematical representation of S-systems is as shown below:

$$\frac{dx_i}{dt} = \alpha_i \cdot \prod_{j=1}^{N} x_j^{g_{ij}} - \beta_i \cdot \prod_{j=1}^{N} x_j^{h_{ij}}, \tag{1}$$

where α_i and β_i are the rate constants for the production and degradation term, respectively, and g_{ij} and h_{ij} are the kinetic orders of the system, also known as the exponential parameters. Now, $g_{ij} > 0$ or $h_{ij} < 0$ represents activation of gene i by gene j, while $g_{ij} < 0$ or $h_{ij} > 0$ signifies repression of gene i by gene j. In S-systems, the expression level x_i of gene i, varies with time according to (assuming $\frac{dx}{dt} \approx \frac{\Delta x}{\Delta t}$ as there are only a few timepoints available):

$$\begin{aligned} \frac{x_i(t+\Delta t) - x_i(t)}{\Delta t} &= \alpha_i \cdot \prod_{j=1}^{N} [x_j(t)]^{g_{ij}} - \beta_i \cdot \prod_{j=1}^{N} [x_j(t)]^{h_{ij}} \\ \Rightarrow x_i(t+\Delta t) - x_i(t) &= \Delta t \cdot \left(\alpha_i \cdot \prod_{j=1}^{N} [x_j(t)]^{g_{ij}} \right) \\ &\quad - \Delta t \cdot \left(\beta_i \cdot \prod_{j=1}^{N} [x_j(t)]^{h_{ij}} \right) \\ \Rightarrow x_i(t+\Delta t) &= x_i(t) + \Delta t \cdot \left(\alpha_i \cdot \prod_{j=1}^{N} [x_j(t)]^{g_{ij}} \right) \\ &\quad - \Delta t \cdot \left(\beta_i \cdot \prod_{j=1}^{N} [x_j(t)]^{h_{ij}} \right) \end{aligned} \tag{2}$$

During the reconstruction of GRNs from temporal expression datasets, the pair of kinetic/exponential parameters g_{ij} and h_{ij} imposes some problems. The most apparent problem is the increase in the number of parameters to be trained, i.e. the total number of model parameters to be trained in the RNN formalism is $N + 2$, whereas, in the S-system formalism, it becomes $2N + 2$, where N denotes the number of genes. Thus, it becomes computationally very expensive as N increases.

Another problem arises when both the obtained values of g_{ij} and h_{ij} are of same sign that implies dual regulations. This is impractical, as a gene cannot activate as well as suppress another gene at the same time. This presents a critical issue during the network reconstruction from the predicted model parameters. To circumvent the issues mentioned above, we have proposed to implement half-systems [16], [17] in this work, for the first time, for the reverse engineering of GRNs from time-series gene expression datasets. Half-systems can be represented by the following:

$$\frac{dx_i}{dt} = \delta_i \cdot \prod_{j=1}^{N} x_j^{f_{ij}}, \tag{3}$$

where δ_i is the only rate constant, and f_{ij} is the only kinetic/exponential parameter. The difference between the production and degradation terms in the S-system formulation has been approximated by a single term in the half-system formulation. Here, $f_{ij} > 0$ signifies that gene j is activating gene i, and $f_{ij} < 0$ denotes that gene j is inhibiting gene i.

Thus, the half-system formalism has a two-fold advantage over S-systems: (*i*) only $N + 1$ parameters need to be trained compared to $2N+2$, and (*ii*) there is no possibility of prediction of any dual regulation, as there is only one kinetic parameter f_{ij} compared to g_{ij} and h_{ij}. Additionally, half-system, like S-system, is a power-law based formalism and hence is quite suitable for modelling GRNs. We have proposed some modifications to the half-system formulation. First, in likeness of the RNN formalism, we have adapted (3) to include the self-degradation effect based on the current expression value of gene i, i.e. x_i, as shown below:

$$\frac{dx_i}{dt} = \delta_i \cdot \prod_{j=1}^{N} x_j^{f_{ij}} - \epsilon_i \cdot x_i, \tag{4}$$

where ϵ_i has been assumed to be positive ($\epsilon_i > 0$). Here, ϵ_i is another parameter that needs to be trained along with δ_i and f_{ij}. The mathematical formulation of the traditional half-system given by (3) did not have any self-degradation term. However, here it has been included in the half-system formalism to represent the phenomena: 'the product of gene expression is not accumulated at the reaction site, rather it is used up or depleted in the process of regulating the expression of other genes'. This depletion process depends on the current level of gene expression. For this reason, the traditional half-system formulation given by (3) has been further modified as given in (4). In this modified half-system formulation, the change in the expression level x_i of gene i, varies with time according to (taking the approximation of $\frac{dx}{dt} \approx \frac{\Delta x}{\Delta t}$):

$$\begin{aligned}
\frac{x_i(t+\Delta t) - x_i(t)}{\Delta t} &= \delta_i \cdot \prod_{j=1}^{N} [x_j(t)]^{f_{ij}} - \epsilon_i \cdot x_i(t) \\
\Rightarrow x_i(t+\Delta t) - x_i(t) &= \Delta t \cdot \left(\delta_i \cdot \prod_{j=1}^{N} [x_j(t)]^{f_{ij}} - \epsilon_i \cdot x_i(t) \right) \\
\Rightarrow x_i(t+\Delta t) &= \Delta t \cdot \left(\delta_i \cdot \prod_{j=1}^{N} [x_j(t)]^{f_{ij}} \right) \\
&\quad + (1 - \Delta t \cdot \epsilon_i) \cdot x_i(t)
\end{aligned} \tag{5}$$

While implementing the modified half-systems for the first time using BAPSO, we have also added another term to maintain a delicate balance between the accuracy of the predicted expression profiles and the actual network structure. Thus, (5) has been modified as below:

$$\begin{aligned}
x_i(t+\Delta t) &= \Delta t \cdot \left(\delta_i \cdot \prod_{j=1}^{N} [x_j(t)]^{f_{ij}} \right) \\
&\quad + (1 - \Delta t \cdot \epsilon_i) \cdot x_i(t) + \mu_i \cdot d_i,
\end{aligned} \tag{6}$$

where μ_i is another parameter that needs to be estimated along with δ_i, f_{ij}, and ϵ_i. The term d_i is the difference between the actual expression value of gene i at any timepoint t (i.e. $x_i(t)$) and its predicted value for the same timepoint (i.e. $\tilde{x}_i(t)$), i.e. it represents the prediction error at this timepoint t. The term $\mu_i d_i$ essentially is an error correcting term, or more precisely, a biologically motivated penalty term, and prevents the predicted expression profiles from deviating much compared to the original profiles.

2.3 Particle Swarm Optimisation (PSO)

Particle swarm optimisation or PSO is one of the most unsophisticated and simple-to-code, yet quite robust and competent swarm intelligence algorithms [70], [71]. It is a population-based algorithm that has been proven to yield better-quality solutions compared to GA. This has been established over an extensive range of optimisation problems. PSO has the additional benefit of possessing a faster exploration rate. PSO comprises a swarm of particles that are randomly scattered in the search space. The aim of each particle is to achieve the fittest solution in the search space, through social interactions with the other particles in the swarm. PSO can be defined completely by the following expression:

$$\begin{aligned}
v_i(it+1) &= w \cdot v_i(it) + c_1 r_1 \cdot [pb_i(it) - p_i(it)] \\
&\quad + c_2 r_2 \cdot [gb(it) - p_i(it)] && (7) \\
p_i(it+1) &= p_i(it) + v_i(it+1), && (8)
\end{aligned}$$

where it denotes the current iteration/generation; $v_i(it+1)$ is the velocity of the ith particle for the next or the $(it+1)$-th generation; $v_i(it)$ is the particle velocity in the current generation; $p_i(it+1)$ is the position of the i-th particle for the next or the $(it+1)$-th generation; and $p_i(it)$ is the particle position in the current or the it-th generation. Also, $pb_i(it)$ indicates the best solution achieved by the i-th particle thus far, and $gb(it)$ denotes the best solution obtained by the entire swarm thus far. Each particle in a swarm can be characterised completely by p, v, and pb. The term w is the inertia weight parameter that controls the subtle balance between exploration and exploitation in the search space.; r_1 and r_2 are random numbers uniformly drawn from $[0, 1]$, and $c_1 = c_2 = 2$. The terms $c_1 r_1$ and $c_2 r_2$ supervise the effect of the best solutions achieved by a particle and the swarm, respectively, on the velocity of a particle.

The three terms of the velocity update equation (7) have different roles in PSO. The first term, $w \cdot v_i(it)$, is the inertia component, responsible for keeping the particle moving in the same direction it was originally heading. The second term, $c_1 r_1 \cdot [pb_i(it) - p_i(it)]$, called the cognitive component, acts as the particle's memory, causing it to tend to return to the regions of the search space in which it has experienced high individual fitness. And, the third term, $c_2 r_2 \cdot [gb(it) - p_i(it)]$, known as the social component, causes the particle to move to the best region the swarm has found so far.

2.4 Bat Algorithm inspired Particle Swarm Optimisation (BAPSO)

Yang *et al.* [72] introduced BA, a nature-inspired metaheuristic based on the echolocation behaviour of bats. Bats transmit loud and high-pitch sounds continuously and listen to the echo from neighbouring objects. A bat can, therefore, figure out the direction and distance of an object from

Algorithm 1 BAPSO algorithm

```
Input: The number of genes in the GRN ($N$); the maximum number of iterations of BAPSO ($maxit$); and the size of the swarm in BAPSO ($ps$)
Output: The weight matrix representing the structure of the inferred network ($F$)
for gene $g = 1$ to $N$ do
    Initialise position vector, $\mathcal{P}^g = [p_i^g]_{1\times ps}$ randomly and velocity vector, $\mathcal{V}^g = [v_i^g]_{1\times ps}$ to $\emptyset$.
    Each element, $p_i^g$ (and $v_i^g$) is defined as: $p_i^g = [f_{i1}^g, f_{i2}^g, \ldots, f_{iN}^g, \delta_i^g, \epsilon_i^g, \mu_i^g]$, where $N$ is the number of genes.
    Calculate the fitness, $er_i^g$, of each of the particles in the swarm using Algorithm 2 and store them in the fitness vector, $\mathcal{E}^g \leftarrow [er_i^g]_{1\times ps}$.
    Store the minimum (best) fitness, $fit_{best} \leftarrow minimum(\mathcal{E}^g)$ and its index in $min$.
    Set personal best solutions, $\mathcal{PB}^g = [pb_i^g]_{1\times ps} \leftarrow \mathcal{P}^g$ and their corresponding fitness, $\mathcal{PE}^g = [per_i^g]_{1\times ps} \leftarrow \mathcal{E}^g$.
    Calculate the global best solution, $gb^g \leftarrow pb_{min}^g$.
    for $iter = 2$ to $maxit$ do
        Update all particle velocities, $\mathcal{V}^g$ using (9).
        Update all particle positions, $\mathcal{P}^g$ using (10).
        Update the fitness of the swarm, $\mathcal{E}^g$, using Algorithm 2.
        Update $\mathcal{PB}^g$, $\mathcal{PE}^g$, $gb^g$, $fit_{best}$, and $min$ using Algorithm 3.
    end for
    Store $gb^g$ at the end of $maxit$ iterations.
end for
Combine the stored $gb^g$ (for $1 \leq g \leq N$) to get an $N \times (N+3)$ matrix.
Extract the first $N$ elements from each row to get an $N \times N$ matrix, $[f_{ij}]_{N\times N}$.
Return $F \leftarrow [f_{ij}]_{N\times N}$.
```

the received waves. In BA, the virtual bats are assumed to possess random flight, and in each generation, some of the bats are randomly designated to perform local search or exploitation, while others undertake global search or exploration. We have incorporated this quality of the virtual bats of BA into the particles in PSO, by replacing the inertia weight parameter w with r_0 here; r_0 is a random number uniformly distributed in $[0, 1]$. Thus, in each generation of BAPSO, some of the particles will perform a global search (exploration), and the remaining particles will perform a local search (exploitation). In addition, the virtual bats in BA are at rest at the start, i.e. the velocity vector is initialised to zero for each virtual bat. We have also included this attribute in BAPSO, and thus, the BAPSO algorithm can be mathematically defined as follows:

$$v_i(it) = \begin{cases} \emptyset & \text{if } it = 1 \\ \begin{cases} r_0 \cdot v_i(it-1) \\ + c_1 r_1 \cdot [pb_i(it-1) - p_i(it-1)] \\ + c_2 r_2 \cdot [gb(it-1) - p_i(it-1)] \end{cases} & \text{if } it > 1 \end{cases} \tag{9}$$

Algorithm 2 Fitness calculation of particles, i.e. obtaining the predicted time-series using Half-systems.

```
Input: The time-series gene expression dataset ($X$); the gene being considered ($g$); and the particle positions ($\mathcal{P}^g$)
Output: Fitness of the swarm ($\mathcal{E}^g$)
Extract number of genes, $N$, from $X$.
Extract number of timepoints, $tp$, from $X$.
Extract population size, $ps$, from $\mathcal{P}^g$.
for $i = 1$ to $ps$ do
    Extract $[f_{ij}^g]_{1\times N}, \delta_i^g, \epsilon_i^g, \mu_i^g$ from $p_i^g$.
    for $t = 2$ to $tp$ do
        Calculate the predicted expression level, $\tilde{x}_i^g(t)$ of gene $g$ from the original expression level at the previous timepoint, i.e. $x_i^g(t-1)$ using (6).
    end for
    Calculate the fitness of particle $p_i^g$ and store it in $er_i^g$ using (11).
end for
Return $\mathcal{E}^g \leftarrow [er_i^g]_{1\times ps}$.
```

Algorithm 3 Updating personal and global best solutions in BAPSO.

```
Input: The particle positions ($\mathcal{P}^g$); the personal best solutions ($\mathcal{PB}^g$), and their fitness ($\mathcal{PE}^g$); the global best solution ($gb^g$), its fitness value ($fit_{best}$), and index ($min$); and the fitness vector ($\mathcal{E}^g$)
Output: Updated $\mathcal{PB}^g$; $\mathcal{PE}^g$; $gb^g$; $fit_{best}$; and $min$
Extract population size, $ps$ from $\mathcal{P}^g$.
for $i = 1$ to $ps$ do
    if $er_i^g < per_i^g$ then          ▷ update the personal best solutions
        $per_i^g \leftarrow er_i^g$       ▷ where $er_i^g \in \mathcal{E}^g$ and $per_i^g \in \mathcal{PE}^g$
        $pb_i^g \leftarrow p_i^g$         ▷ where $pb_i^g \in \mathcal{PB}^g$ and $p_i^g \in \mathcal{P}^g$
    end if
    if $er_i^g < fit_{best}$ then       ▷ update the global best solution
        $fit_{best} \leftarrow er_i^g$                    ▷ where $er_i^g \in \mathcal{E}^g$
        $gb^g \leftarrow p_i^g$                           ▷ where $p_i^g \in \mathcal{P}^g$
        $min \leftarrow i$
    end if
end for
Return updated $\mathcal{PB}^g$, $\mathcal{PE}^g$, $gb^g$, $fit_{best}$, and $min$.
```

$$p_i(it) = p_i(it-1) + v_i(it) \tag{10}$$

2.5 Methods

In this section, we have elucidated our proposed half-system based technique for reverse engineering GRNs from temporal gene expression datasets. Researchers have employed a decoupling scheme [18], [19], [57], [61], where the problem of network inference can be divided into two sub-problems: (*i*) search for an appropriate and biologically plausible GRN, and (*ii*) proper training of the corresponding half-system model parameters. We have implemented this approach in this work. The proposed technique has been explained in Algorithms 1-3.

Let us first consider the first half of the decoupled problem. Bolouri *et al.* [13] have stated that on an average, a gene is regulated by ***four*** to ***eight*** other genes. Here, we have employed our proposed methodology on 8-gene, 10-gene, and 20-gene networks for analyses. Since these are relatively smaller networks, we have assumed the in-degree for a gene (i.e. the maximum number of regulators for it) to be ***four*** at maximum. This assumption reduces the discrete search space of probable GRNs, significantly. Moreover, with a maximum in-degree of ***four***, there can be a maximum of $\binom{N}{4}$ number of probable candidate network structures, where N is the number of genes in the given GRN. This shrinks the overall search space from 2^N to $\binom{N}{4}$. Furthermore, since all possible combinations of regulators are being considered, the biological credibility of the candidate GRNs are likely to be maintained to the maximum possible extent.

Additionally, the problem of overfitting can also be minimised by restricting the number of regulators, which results in *model simplification*. Had there been no constraint on the number of regulators, BAPSO would have had the liberty to use all the N parameters to train the modified half-system based model. This would have resulted in an unnecessary increase in the model complexity. While this may have reduced the prediction (training) error significantly, it would also have led to an *overfitted* model. This problem could also have been tackled by using a L1-regularizer. However, we chose to use an explicitly simplified model (using four regulators) because of the availability of biological information in this domain, which has proved to be relevant in this type of work. We have also implemented two-fold cross validation in this work to further prevent overfitting and improve the generalizability of the model.

Next, let us move on to the problem of training the half-system model parameters. If there are N genes in a GRN, $N+3$ parameters for each gene are to be trained according to our proposed model, i.e. f_{ij} (N parameters), δ_i, ϵ_i, and μ_i. Thus, the dimension of the optimisation problem is $N \times (N+3)$, which turns out to be computationally quite expensive for bigger values of N.

Biologically, GRNs are known to be sparse [11], i.e. most of the values of $[f_{ij}]_{N\times N}$ (in (6)) are zero. However, even this apparently advantageous constraint cannot lessen the computational burden appreciably. For this reason, researchers [57], [73] proposed to decompose the $N \times (N+3)$ dimensional optimisation problem into N sub-problems of $(N+3)$ dimensions each. Therefore, each gene can be studied separately, and the respective $(N+3)$ half-system model parameters trained in each case. Thus, in this work, the fitness/objective function for the metaheuristic techniques has been redefined as:

$$er_i = \frac{1}{T} \cdot \sum_{t=1}^{T} (x_i(t) - \tilde{x}_i(t))^2, \tag{11}$$

where T is the number of timepoints available, and $x_i(t)$ and $\tilde{x}_i(t)$ are the original and predicted expression values, respectively, of gene i at any timepoint t.

In this work, we have denoted a GRN by a directed graph $G = (V, E)$, where V and E comprise the vertices (genes) and the edges (relationship among the genes), respectively. For computational purposes, G has been represented as $G = [g_{ij}]_{N\times N}$, where N is the number of genes in the GRN. The value of the element g_{ij} is 0 or 1 depending on the absence or presence, of an edge from node j to node i, respectively. Due to the stochastic nature of the training algorithms implemented, the inferred GRNs are expected to fluctuate in their structures with each independent experimentation for a given network. Thus, we have employed a cooperative training scheme, where we have generated L independent GRNs for each of the L independent experiments. Subsequently, we have developed a selection technique based on a plausibility score ps_{ij}, assigned to each edge of the deduced GRN as:

$$ps_{ij} = \frac{1}{L} \cdot \sum_{l=1}^{L} f_{ij}^{l}, \tag{12}$$

where $f_{ij}^{l} \in F^l$, and F^l denotes the structure of the predicted network obtained in the lth experiment. We have generated each F according to Algorithms 1-3. For each of the four networks studied in this work, we have put together the final inferred GRN, $G = [g_{ij}]_{N\times N}$ based on ps_{ij}, as:

$$g_{ij} = \begin{cases} 0, & \text{if } ps_{ij} < \phi \\ 1, & \text{otherwise,} \end{cases} \tag{13}$$

where ϕ has been defined as a threshold of the plausibility score, ps_{ij}, for inclusion of an edge in the final inferred GRN; g_{ij} denotes the presence or absence of an edge in the predicted GRN (represented by the $N \times N$ matrix G), and does not represent the type of relationship amongst the nodes (activation or repression). Thus, we have reconstructed the GRN structure from a single time-series gene expression dataset.

However, in the present work, we have used multiple expression datasets, e.g., four datasets in the case of the 8-gene network and five datasets in the case of the 10-gene and 20-gene networks. In the case of multiple datasets, according to (13), first we have estimated the G-s corresponding to each dataset. Next, inclusion score, is_{ij}, corresponding to each predicted edge has been calculated using (14) given below:

$$is_{ij} = \frac{1}{D} \cdot \sum_{d=1}^{D} g_{ij}^{d}, \tag{14}$$

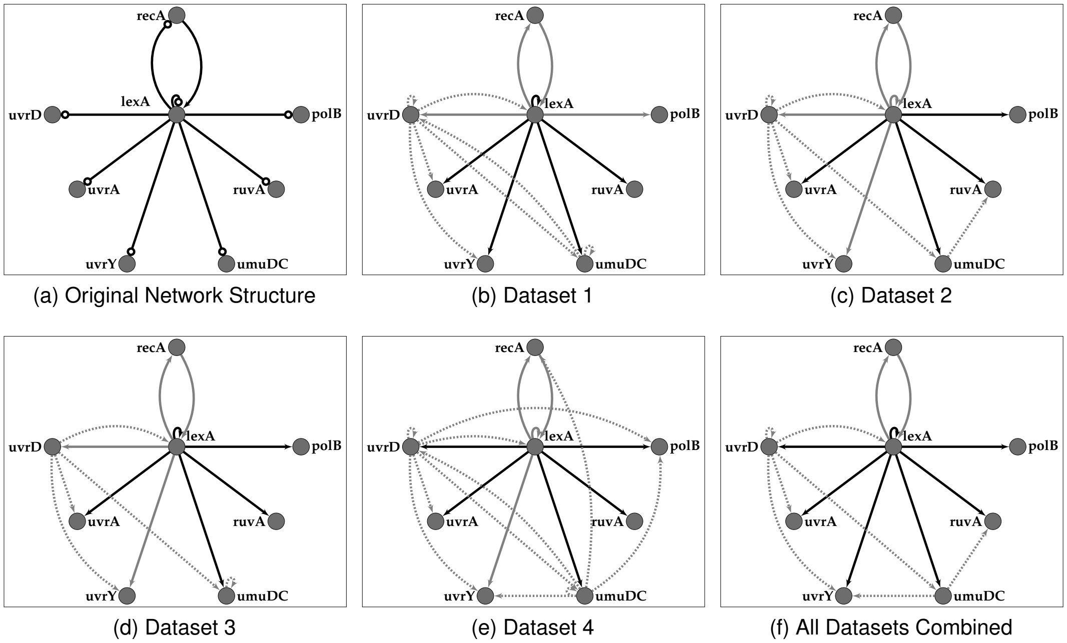
where D is the number of datasets used and $G^d = [g_{ij}^d]_{N\times N}$. G^d represents the reconstructed GRN using gene expression dataset, d. The final inferred GRN, G^M has been generated by the following:

$$g_{ij}^{m} = \begin{cases} 0, & \text{if } is_{ij} < \psi \\ 1, & \text{otherwise,} \end{cases} \tag{15}$$

where $G^M = [g_{ij}^m]_{N\times N}$ and ψ is a threshold of the inclusion score, is_{ij}, for keeping an edge in G^M.

3 Experimental Results and Discussion

In this work, we have implemented this new methodology based on half-systems for the reconstruction of GRNs for four instances, i.e. one 8-gene, one 10-gene, and two 20-gene networks, from their respective temporal expression profiles. The 10-gene and 20-gene networks have been extracted from the GNW [74] database, and the time-series expression datasets have been generated using DREAM4 [75], [76] settings. The 8-gene network is the well-known



(a) Original Network Structure

(b) Dataset 1

(c) Dataset 2

(d) Dataset 3

(e) Dataset 4

(f) All Datasets Combined

Fig. 1. (a) The original topology of the *E. coli* SOS DNA Repair Network. The arrowheads represent activation, and the open circles denote inhibition. (b)-(e) The structure of the *E. coli* SOS DNA Repair Network as predicted by the proposed methodology from individual datasets (with $\phi = 0.9$). (f) The structure of the derived *E. coli* SOS DNA Repair Network, when all datasets are used together for inference (with $\phi = 0.8$ and $\psi = 1.0$). In (b)-(f), the thick black edges represent TPs, the thin gray edges represent FNs, and the gray dashed edges represent FPs.

benchmark SOS DNA Repair Network of *E. coli*, studied by Ronen *et al.* [77]. They analysed the expression of eight genes, namely, recA, lexA, uvrA, uvrD, uvrY, umuD, ruvA, and polB involved in the SOS repair system. The authors in [77] performed four experiments, and in each experiment, the expression of the eight genes was observed for 50 time points with a temporal resolution of 6 minutes. These datasets still prove to be one of the most useful datasets for research on the computational reconstruction of GRNs from temporal expression data, and can be found at http://wws.weizmann.ac.il/mcb/UriAlon/sites/mcb.UriAlon/files/uploads/DownloadableData/sosdata.zip.

The obtained results have been evaluated based on certain metrics defined as follows:

$$\text{TPR or } S_n \text{ or } recall = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{16}$$

$$\text{FPR or } fall\text{-}out = \frac{\text{FP}}{\text{FP} + \text{TN}}, \tag{17}$$

$$\text{TNR or } S_p = \frac{\text{TN}}{\text{FP} + \text{TN}}, \tag{18}$$

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

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

$$F\text{-}score = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}, \tag{21}$$

where **TP** denotes true positives, the number of correctly inferred existing edges, while **FP** denotes false positives, the number of incorrectly inferred non-existent edges. Similarly, **TN** represents true negatives, the number of correctly predicted non-existent edges, while **FN** represents false negatives, the number of incorrectly inferred existing edges.

3.1 *E. coli* SOS DNA Repair Network

There are ***four*** datasets each with 50 timepoints at a temporal resolution of 6 minutes. We have normalised the data in the range [0,1]. Also, the gene expression values at the first timepoint in each dataset are zero, and thus, have been removed from our experimentation. The original network comprising these genes have been shown in Fig. 1(a).

In the contemporary literature, some researchers [18], [57] have implemented their proposed methodologies to construct a network from each of the four datasets, separately. Hence, we have also followed the same approach using our proposed technique. Table 1 presents the obtained results. It can be observed that the proposed half-system based formalism performs in a similar manner to other RNN based approaches. The number of predicted TPs and FPs compares favourably to eDSF (except Dataset 02) [57] and to both PSO and BAPSO [18]. It can also be noted from Table 1 that the proposed methodology is consistent across datasets w.r.t. the number of inferred TPs. However, the other methods all lack this consistency, and their performance is better in some datasets, while very poor in others, such as eDSF [57] can predict 3 and 4 TPs from Datasets 01 and 03, respectively, but cannot identify any TP from Dataset 04; similarly, BAPSO [18] can identify 7 TPs from Datasets 01-03, but can predict only 3 TPs from Dataset 04. In the proposed technique, however, the results are consistent because it successfully predicted 5 TPs in 3 out of the 4 dataets and 4 TPs from the remaining one dataset. The predicted network structures, in comparison to the original, have been shown in Fig. 1(b)-(e).

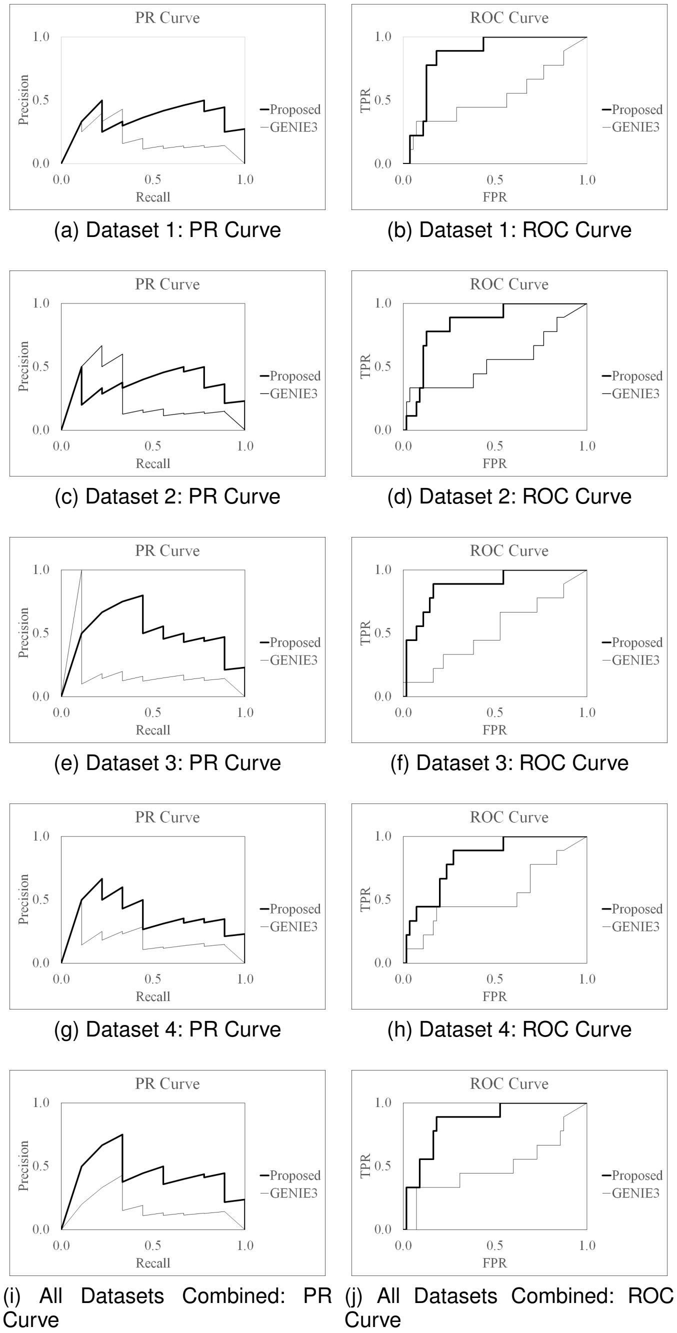
In the contemporary literature, it is also found that several other researchers [14], [53], [61], [66] opted to reconstruct a single network using all the four datasets in some

TABLE 1
Comparison of the results obtained by the proposed half-system based technique with those given in other such works present in the contemporary literature for the 8-gene *E. coli* SOS DNA Repair network [77] (w.r.t. individual datasets and a plausibility score threshold, $\phi = 0.9$).

Technique	TP	FP	TPR (S_n)	FPR	TNR (S_p)	PPV	ACC	*F-score*
Dataset 01								
Proposed	5	7	0.556	0.127	0.873	0.417	0.828	0.476
eDSF [57]	3	10	0.333	0.182	0.818	0.231	0.750	0.273
PSO [18]	5	9	0.556	0.164	0.836	0.357	0.797	0.435
BAPSO [18]	7	9	0.778	0.164	0.836	0.438	0.828	0.560
Dataset 02								
Proposed	4	6	0.444	0.109	0.891	0.400	0.828	0.421
eDSF [57]	8	5	0.889	0.091	0.909	0.615	0.906	0.727
PSO [18]	4	10	0.444	0.182	0.818	0.286	0.766	0.348
BAPSO [18]	7	15	0.778	0.273	0.727	0.318	0.734	0.452
Dataset 03								
Proposed	5	5	0.556	0.091	0.909	0.500	0.859	0.526
eDSF [57]	4	10	0.444	0.182	0.818	0.286	0.766	0.348
PSO [18]	5	8	0.556	0.145	0.855	0.385	0.813	0.455
BAPSO [18]	7	8	0.778	0.145	0.855	0.467	0.844	0.583
Dataset 04								
Proposed	5	11	0.556	0.200	0.800	0.313	0.766	0.400
eDSF [57]	0	9	0.000	0.164	0.836	0.000	0.719	0.000
PSO [18]	3	8	0.333	0.145	0.855	0.273	0.781	0.300
BAPSO [18]	4	12	0.444	0.218	0.782	0.250	0.734	0.320

TABLE 2
Comparison of the results obtained by the proposed half-system based method with those given in other similar works present in the contemporary literature for the 8-gene *E. coli* SOS DNA Repair network [77] (w.r.t. multiple datasets, and $\phi = 0.8$ and $\psi = 1.0$).

Technique	TP	FP	TPR (S_n)	FPR	TNR (S_p)	PPV	ACC	*F-score*
Proposed	7	7	0.778	0.127	0.873	0.500	0.859	0.609
Perrin *et al.* [14]	4	5	0.444	0.091	0.909	0.444	0.844	0.444
Xu *et al.* [53]	5	2	0.556	0.036	0.964	0.714	0.906	0.625
Noman *et al.* [61]	5	10	0.556	0.182	0.818	0.333	0.781	0.417
Mandal *et al.* [66]	8	0	0.889	0.000	1.000	1.000	0.984	0.941



(a) Dataset 1: PR Curve

(b) Dataset 1: ROC Curve

(c) Dataset 2: PR Curve

(d) Dataset 2: ROC Curve

(e) Dataset 3: PR Curve

(f) Dataset 3: ROC Curve

(g) Dataset 4: PR Curve

(h) Dataset 4: ROC Curve

(i) All Datasets Combined: PR Curve

(j) All Datasets Combined: ROC Curve

Fig. 2. Comparison of the Precision Recall (PR) and the Receiver Operating Characteristic (ROC) curves obtained by the proposed half-system based technique with those obtained using **GENIE3** [20] for the *E. coli* SOS DNA Repair network [77]. The thick lines represent the proposed technique, while the thin lines denote **GENIE3** [20].

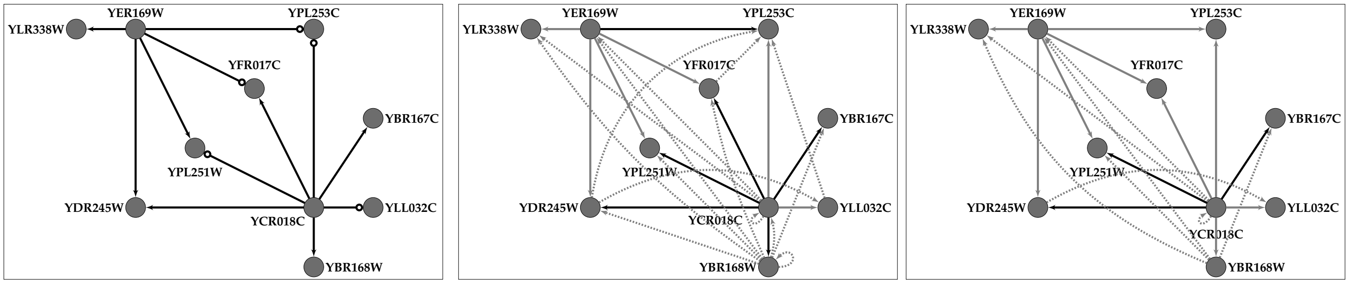
suitable combination. Therefore, we have also implemented the same strategy using our proposed technique. The results produced have been presented in Table 2. The network reconstruction strategy used by the previous authors [14], [53], [61] was not explicitly mentioned in their presentations. Nevertheless, one of the research works [66] used a filtering approach, which is analogous to $\phi = 0.8$ in our work. Here, in the beginning, network structures, G^d, for each dataset d ($d = 1, 2, 3, 4$) have been reconstructed with a plausibility score threshold, $\phi = 0.8$ (instead of $\phi = 0.9$ as in Table 1). As a result, we have obtained the following observations:

$$G^1 : \text{TP} = 8, \text{FP} = 10;$$
$$G^2 : \text{TP} = 7, \text{FP} = 10;$$
$$G^3 : \text{TP} = 8, \text{FP} = 12;$$
$$G^4 : \text{TP} = 7, \text{FP} = 14.$$

Next, these four G^d-s have been combined by filtering out the common edges only (i.e. the inclusion score threshold, $\psi = 1.0$) to infer the final GRN, G^M. The corresponding results have been presented in Table 2 along with the comparison with similar other research works. The network structures produced by our proposed technique along with the original network have been shown in Fig. 1(f). This has facilitated the measurement of the strength of the proposed technique. It can be observed that the results obtained using a plausibility score threshold, $\phi = 0.8$ are also consistent across the datasets, and are considerably better in terms of inferred TPs than those in Table 1.

It can be further observed from Table 2 that the proposed approach can produce satisfactory results (i.e. it can predict the second most number of TPs) compared to similar other methodologies based on RNN and S-systems. Therefore, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2892450, IEEE/ACM Transactions on Computational Biology and Bioinformatics



(a) Original Network Structure

(b) Single Dataset

(c) Multiple Datasets

Fig. 3. (a) The original structure of the 10-gene network extracted from GNW. (b) The network structure derived by the proposed methodology, when a single dataset is used for network inference. (c) The network structure derived by the proposed methodology, when multiple (five) datasets are used together for network inference. In (b)-(c), the thick black edges represent TPs, the thin gray edges represent FNs, and the dashed gray edges represent FPs.

TABLE 3
Comparison of the AUC scores achieved by the proposed half-system based technique with those obtained by **GENIE3** [20] for the 8-gene *E. coli* SOS DNA Repair network [77].

Method	Dataset 01	Dataset 02	Dataset 03	Dataset 04	All Datasets
			AUPR		
Proposed	0.356257	0.352015	0.493781	0.372941	0.430618
GENIE3	0.183096	0.238565	0.177155	0.169015	0.165154
			AUROC		
Proposed	0.854545	0.840404	0.876768	0.822222	0.858586
GENIE3	0.526263	0.538384	0.516162	0.528283	0.497980

proposed methodology succeeds in taking the lead, in the matter of GRN reconstruction, over most of the techniques proposed by other researchers as shown in Table 2.

Therefore, in summary, it can be inferred that the proposed network inference technique based on half-systems is reliable and consistent across all datasets of the *E. coli* SOS DNA Repair network, although it is not outright better in terms of TPs inferred, when each dataset is considered exclusively. On the other hand, the proposed technique gives the second best results, when combining multiple datasets, compared to other such methods proposed in the contemporary literature.

Additionally, we have compared our proposed technique with **GENIE3** [20]. The Precision-Recall (PR) and Receiver Operating Characteristic (ROC) curves have been presented in Fig. 2. The corresponding AUC scores have also been given in Table 3, and it can be clearly observed that the AUC (both AUPR and AUROC) scores for the proposed methodology are always better than **GENIE3** [20].

3.2 10-gene Network Extracted from GNW

The original structure of the 10-gene network, extracted from GNW, has been shown in Fig. 3(a). The network has 12 interactions. We have extracted this network from the genome of *Saccharomyces Cerevisiae* present in the GNW database [74]. The network dynamics have been generated using DREAM4 settings. We have generated five datasets. The number of regulators allowed has been varied from 2 to 4. We have presented the comparison of the obtained results with previous research works in Table 4.

TABLE 4
Comparison of the results obtained by the proposed half-system based method with those given in other such works present in the contemporary literature for the 10-gene network extracted from GNW (Fig. 3(a)), when a single dataset is used for network reconstruction.

Technique	TP	FP	TPR (S_n)	FPR	TNR (S_p)	PPV	ACC	$F\text{-}score$
Proposed	6	15	0.500	0.170	0.830	0.286	0.790	0.364
eDSF [57]	4	7	0.333	0.080	0.920	0.364	0.850	0.348
PSO [18]	5	4	0.417	0.045	0.955	0.556	0.890	0.476
BAPSO [18]	5	9	0.417	0.102	0.898	0.357	0.840	0.385

TABLE 5
Comparison of results obtained by the proposed methodology for single and multiple datasets (with an inclusion score threshold, $\psi = 1.0$).

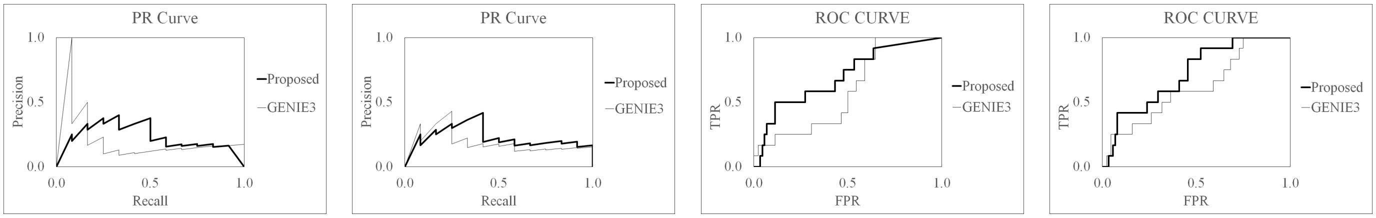
Dataset	TP	FP	TPR (S_n)	FPR	TNR (S_p)	PPV	ACC	$F\text{-}score$
Single	6	15	0.500	0.170	0.830	0.286	0.790	0.364
Multiple	3	8	0.250	0.091	0.909	0.273	0.830	0.261

TABLE 6
Comparison of the AUC scores achieved by the proposed half-system based technique with those obtained by **GENIE3** [20] for the 10-gene network shown in Fig. 3(a).

Method	Single Dataset	Multiple Datasets
	AUPR	
Proposed	0.225522	0.221298
GENIE3	0.193467	0.182607
	AUROC	
Proposed	0.699811	0.717803
GENIE3	0.589015	0.615530

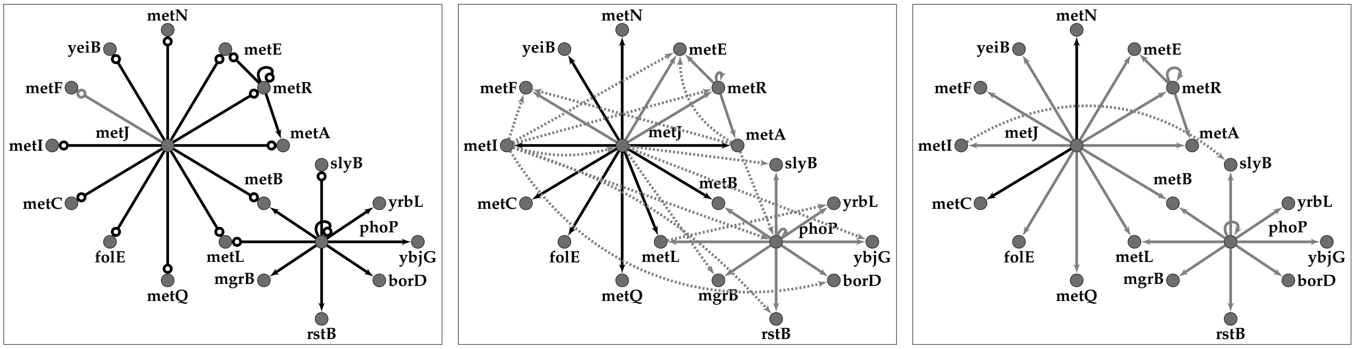
It can be observed from Table 4 that the proposed half-system based technique returned the best results w.r.t. the number of TPs. When multiple datasets are used, the proposed method improves the TNR and ACC of the predicted model, due to a larger reduction in the number of inferred FPs compared to TPs, as can be seen from Table 5. The derived network structures for the single dataset and multiple dataset cases have been shown in Figs. 3(b) and 3(c), respectively. Moreover, we have also implemented **GENIE3** [20] for this and presented the corresponding PR and ROC curves in Fig. 4, and the related AUC scores in Table 6.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2892450, IEEE/ACM Transactions on Computational Biology and Bioinformatics



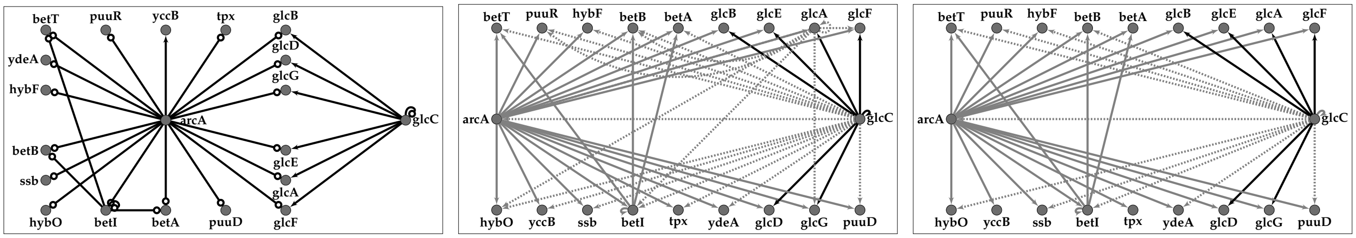
(a) Single Dataset: PR Curve (b) Multiple Datasets: PR Curve (c) Single Dataset: ROC Curve (d) Multiple Datasets: ROC Curve

Fig. 4. Comparison of the Precision Recall (PR) and the Receiver Operating Characteristic (ROC) curves obtained by the proposed half-system based technique with those obtained using **GENIE3** [20] for the 10-gene network extracted from GNW (Fig. 3(a)). The thick lines represent the proposed technique, while the thin lines denote **GENIE3** [20].



(a) Original Network Structure (b) Single Dataset (c) Multiple Datasets

Fig. 5. (a) The original structure of the first 20-gene network extracted from GNW. (b) The network structure derived by the proposed methodology, when a single dataset is used for inference. (c) The structure of the derived network, when multiple (five) datasets are used together for network inference. In (b)-(c) The thick black edges represent TPs, the thin gray edges represent FNs, and the dashed gray edges represent FPs.



(a) Original Network Structure (b) Single Dataset (c) Multiple Datasets

Fig. 6. (a) The original structure of the second 20-gene network extracted from GNW. (b) The network structure derived by the proposed technique, when a single datset is used for inference. (c) The structure of the derived network, when multiple (five) datasets are used together for network inference. In (b)-(c), the thick black edges represent TPs, the thin gray edges represent FNs, and the dashed gray edges represent FPs.

3.3 20-gene Networks Extracted from GNW

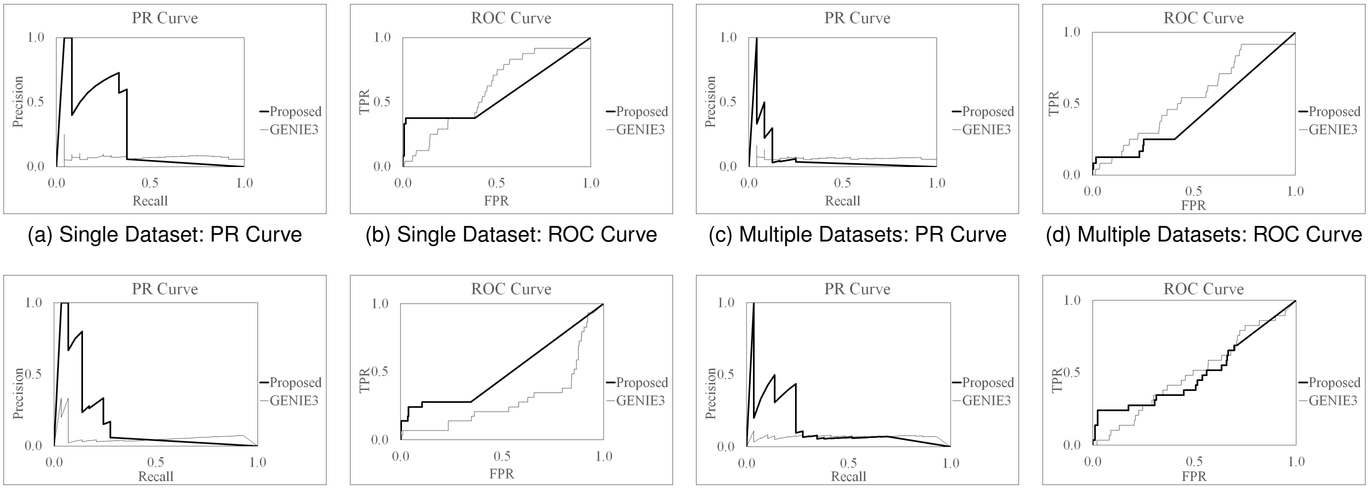
In this case, we have extracted two 20-gene networks from GNW to observe the performance of the proposed methodology for somewhat larger networks. The original structures of the two networks have been shown in Figs. 5(a) and 6(a), and they have 24 and 29 regulatory relationships, respectively. The network dynamics have been generated using DREAM4 settings. We have generated five datasets, each with 41 timepoints. The obtained results have been shown in Tables 7, 8, and 9.

TABLE 7
Comparison of the results obtained by the proposed half-system based method with those given in other such works present in the contemporary literature for the 20-gene network extracted from GNW (Fig. 5(a)), when a single dataset is used for network reconstruction.

Technique	TP	FP	TPR (S_n)	FPR	TNR (S_p)	PPV	ACC	F-$score$
Proposed	9	14	0.375	0.037	0.963	0.391	0.928	0.383
PSO [18]	4	26	0.167	0.069	0.931	0.133	0.885	0.148
BAPSO [18]	4	30	0.167	0.080	0.920	0.118	0.875	0.138
BA [66]	4	41	0.167	0.109	0.891	0.089	0.848	0.116

The results produced by the proposed half-system based formalism for the first 20-gene network (shown in Fig.5(a)) are the best w.r.t. the number of inferred TPs, when a single dataset is used. From Table 7, we can also observe that our proposed technique can predict more than double the number of TPs, compared to the other similar methods present in the contemporary literature. For the second 20-gene network (shown in Fig. 6(a)), the results produced are similar to that of the first network, when a single dataset is used. When multiple datasets are used, the proposed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2892450, IEEE/ACM Transactions on Computational Biology and Bioinformatics



(a) Single Dataset: PR Curve (b) Single Dataset: ROC Curve (c) Multiple Datasets: PR Curve (d) Multiple Datasets: ROC Curve

(e) Single Dataset: PR Curve (f) Single Dataset: ROC Curve (g) Multiple Datasets: PR Curve (h) Multiple Datasets: ROC Curve

Fig. 7. Comparison of the Precision Recall (PR) and the Receiver Operating Characteristic (ROC) curves obtained by the proposed half-system based technique with those obtained using **GENIE3** [20] for: (a)-(d) the 20-gene GRN shown in Fig. 5(a); and (e)-(h) the 20-gene GRN shown in Fig. 6(a). The thick lines represent the proposed technique, while the thin lines denote **GENIE3** [20]

TABLE 8
Results produced by the proposed method for single and multiple datasets of the 20-gene network extracted from GNW (Fig. 5(a)).

Dataset	TP	FP	TPR (S_n)	FPR	TNR (S_p)	PPV	ACC	*F-score*
Single	9	14	0.375	0.037	0.963	0.391	0.928	0.383
Multiple	2	1	0.083	0.003	0.997	0.667	0.943	0.148

TABLE 9
Results produced by the proposed method for single and multiple datasets of the 20-gene network extracted from GNW (Fig. 6(a)).

Datasets	TP	FP	TPR (S_n)	FPR	TNR (S_p)	PPV	ACC	*F-score*
Single	7	22	0.241	0.059	0.941	0.241	0.890	0.241
Multiple	6	12	0.207	0.032	0.968	0.333	0.913	0.255

TABLE 10
Comparison of the AUC scores achieved by the proposed half-system based technique with those obtained by **GENIE3** [20] for the 20-gene network extracted from GNW and shown in Fig. 5(a).

Method	Single Dataset	Multiple Datasets
AUPR		
Proposed	0.256401	0.069107
GENIE3	0.076034	0.065958
AUROC		
Proposed	0.565160	0.441822
GENIE3	0.602837	0.545213

TABLE 11
Comparison of the AUC scores achieved by the proposed half-system based technique with those obtained by **GENIE3** [20] for the 20-gene network extracted from GNW and shown in Fig. 6(a).

Method	Single Dataset	Multiple Datasets
AUPR		
Proposed	0.159205	0.136119
GENIE3	0.058274	0.068078
AUROC		
Proposed	0.505391	0.501069
GENIE3	0.296403	0.498373

TABLE 12
Two-fold cross-validation results for the *E. coli* SOS DNA Repair network.

Network	Dataset #	Training Error	Validation Error
8-gene	1	0.001845	0.002079
	2	0.001186	0.001315
	3	0.003403	0.003532
	4	0.002367	0.002047

methodology improves the specificity (S_p) and accuracy of the predicted models for both the 20-gene networks, due to a larger reduction in the number of inferred FPs compared to TPs, as can be seen from Tables 8 and 9.

The predicted network structures for the first 20-gene network (shown in Fig. 5(a)) have been given in Fig. 5(b), when a single dataset is used, and in Fig. 5(c), when multiple datasets are used together for network inference. Similarly, for the second 20-gene network (shown in Fig. 6(a)), the inferred structures have been presented in Fig. 6(b), when a single dataset is used, and in Fig. 6(c), when multiple datasets are used. We have also implemented **GENIE3** [20] on these two networks. The obtained PR and ROC curves have been shown in Fig. 7. The corresponding AUC scores, in comparison to **GENIE3**, have been shown in Tables 10 and 11 for the two networks.

In addition to all the above-mentioned experimentations, two-fold cross-validation has also been performed on the results of the experimentations to prove the generalisation property of the proposed model. The corresponding errors have been presented in Tables 12, 13. We would also like to mention that for the 10-gene and 20-gene networks, we have used five datasets to reconstruct five separate networks. Subsequently, these networks have been combined using

TABLE 13
Two-fold cross-validation results for the *in silico* cases, taking multiple (five) datasets in each case. The *minimum* (*maximum*) refers to the *smallest* (*largest*) training/validation error amongst the five MSEs obtained from the five datasets. The *mean* represents the *average of all the five training/validation errors* obtained.

Network		Training Error	Validation Error
10-gene	Minimum	0.006672	0.006970
	Mean	0.007830	0.007858
	Maximum	0.009632	0.009748
20-gene (1) (Fig. 5(a))	Minimum	0.002977	0.003037
	Mean	0.004337	0.004444
	Maximum	0.007038	0.007350
20-gene (2) (Fig. 6(a))	Minimum	0.003703	0.003777
	Mean	0.005117	0.005153
	Maximum	0.009206	0.008756

TABLE 14
Comparison of computational times between S-system and the proposed half-system based methodologies.

Method	8-gene	10-gene	20-gene (1)	20-gene (2)
Proposed	0.05 mins	0.22 mins	0.46 hrs	0.47 hrs
S-System	2.18 mins	5.97 mins	12.87 hrs	13.12 hrs

a suitable value of inclusion threshold, ψ to generate the final networks for the cases of multiple datasets that are mentioned in Tables 5, 8, and 9.

4 Conclusion

Overall, in this work, we have proposed for the first time, to the best of our knowledge, a methodology based on modified half-systems for the reconstruction of GRNs from time-series gene expression data. We have implemented it for the reverse engineering of four networks: (*i*) the real-world SOS DNA Repair network of *E. coli* from experimental datasets provided by [77]; (*ii*) a 10-gene network extracted from GNW previously studied by Kentzoglanakis *et al.* [57] and Khan *et al.* [18]; and (*iii*) two larger networks comprising 20 genes, one of them also previously studied by Mandal *et al.* [66] and Khan *et al.* [18], [19].

Previously, many researchers in this domain have employed, and continue to employ S-systems for GRN reconstruction. However, half-systems require much less time, as shown in Table 14 and are thus, less computationally intensive than S-systems. Though S-systems and half-systems have resemblances, intuitively it can be inferred that half-system is a much better model, in a biological sense, for the reverse engineering of GRNs. Moreover, compared to **GENIE3**, the proposed half-system-based technique achieved much better AUC scores for each of the cases studied herein, thereby proving its superiority over **GENIE3**, as well.

The results, as indicated, are either favourably comparable to (8-gene network) or better (10-gene and 20-gene networks) than those presented in the contemporary literature. The proposed half-system based formalism generally predicts a higher number of TPs, when a single dataset is used. The improvement in the quality of the inferred networks is especially prominent in the case of the larger networks. The results obtained for the 20-gene networks are very promising compared to the results in the contemporary literature. Also, we would like to mention that all these benefits are extracted during GRN inference in a *lesser* computational time, which is a significant gain.

For the *E. coli* SOS DNA Repair network, when multiple datasets are considered, the proposed formalism produces the second best results w.r.t. the number of TPs. However, along with this, it is observed that the proposed technique also identifies the second highest number of FPs that is unwanted. There may be a variety of reasons for this unwanted result, and the most prominent among them may be: firstly, a metaheuristic, other than BAPSO, might have been more suitable for model parameter training from multiple datasets, which may have overcome this difficulty; secondly, overfitting is a common issue with such ill-posed problems as the reverse engineering of GRNs. In this investigation, we have minimised this problem of overfitting by limiting the maximum number of regulators allowed for a gene in a network (thus reducing the model complexity), based on prevalent biological knowledge in the domain. However, traditional regularisation techniques may be investigated in future research initiatives in this domain to verify whether this problem can be better avoided or not.

Since no research work (to the best of our knowledge) is available in the cases of the 10-gene and 20-gene networks using multiple datasets, comparison with other research works could not be provided for these cases. Significantly, for these larger networks, the proposed methodology produces better specificity, S_p, and accuracy of the predicted models compared to those obtained from a single dataset. The performance of half-systems in the reverse engineering of larger genetic networks (more than 50 genes) is yet to be investigated. Further refinement of the model equation, which can further improve the results of network inference, is an open challenge for researchers of this domain in the future.

Acknowledgements

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). The authors would also like to acknowledge the respected reviewers for enriching this work.

References

[1] S. Kimura, K. Ide, A. Kashihara, M. Kano, M. Hatakeyama, R. Masui, N. Nakagawa, S. Yokoyama, S. Kuramitsu, and A. Konagaya, "Inference of s-system models of genetic networks using a cooperative coevolutionary algorithm," *Bioinformatics*, vol. 21, no. 7, pp. 1154–1163, 2004.

[2] J. Quackenbush, "Microarray data normalization and transformation," *Nature Genetics*, vol. 32, no. Supp, p. 496, 2002.

[3] P. D'haeseleer, X. Wen, S. Fuhrman, R. Somogyi *et al.*, "Linear modeling of mrna expression levels during cns development and injury." in *Pacific Symposium on Biocomputing*, vol. 4, no. 1, 1999, pp. 41–52.

[4] I. M. Ong, J. D. Glasner, and D. Page, "Modelling regulatory pathways in e. coli from time series expression profiles," *Bioinformatics*, vol. 18, no. suppl_1, pp. S241–S248, 2002.

[5] Z. Bar-Joseph, "Analyzing time series gene expression data," *Bioinformatics*, vol. 20, no. 16, pp. 2493–2503, 2004.

[6] L. Ironi and D. X. Tran, "Nonlinear and temporal multiscale dynamics of gene regulatory networks: A qualitative simulator," *Mathematics and Computers in Simulation*, vol. 125, pp. 15–37, 2016.

[7] G. McLachlan, K.-A. Do, and C. Ambroise, *Analyzing microarray gene expression data*. John Wiley & Sons, 2005, vol. 422.

[8] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[9] 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.

[10] 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S108495211630012X

[11] P. D'haeseleer and S. Forrest, "Reconstructing gene networks from large scale gene expression data," Ph.D. dissertation, University of New Mexico, 2000.

[12] E. v. Someren, L. Wessels, E. Backer, and M. Reinders, "Genetic network modeling," *Pharmacogenomics*, vol. 3, no. 4, pp. 507–525, 2002.

[13] H. Bolouri and E. H. Davidson, "Modeling transcriptional regulatory networks," *BioEssays*, vol. 24, no. 12, pp. 1118–1129, 2002.

[14] B.-E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. d'Alche Buc, "Gene networks inference using dynamic bayesian networks," *Bioinformatics*, vol. 19, no. suppl_2, pp. ii138–ii148, 2003.

[15] A. R. Chowdhury and M. Chetty, "Reconstruction of large-scale gene regulatory network using s-system model," *Evolutionary Computation in Gene Regulatory Network Research*, p. 185, 2016.

[16] 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.

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

[18] 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, 2016.

[19] A. Khan, G. Saha, and R. K. Pal, "An approach for reduction of false predictions in reverse engineering of gene regulatory networks," *Journal of Theoretical Biology*, vol. 445, pp. 9–30, 2018.

[20] 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, pp. 1–10, 09 2010.

[21] H. Lähdesmäki, I. Shmulevich, and O. Yli-Harja, "On learning gene regulatory networks under the boolean network model," *Machine Learning*, vol. 52, no. 1, pp. 147–167, 2003.

[22] C.-C. Chen and S. Zhong, "Inferring gene regulatory networks by thermodynamic modeling," *BMC Genomics*, vol. 9, no. 2, p. S19, 2008.

[23] 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.

[24] T. Akutsu, S. Miyano, S. Kuhara *et al.*, "Identification of genetic networks from a small number of gene expression patterns under the boolean network model." in *Pacific Symposium on Biocomputing*, vol. 4, 1999, pp. 17–28.

[25] M. I. Davidich and S. Bornholdt, "Boolean network model predicts cell cycle sequence of fission yeast," *PloS One*, vol. 3, no. 2, p. e1672, 2008.

[26] A. Saadatpour and R. Albert, "Boolean modeling of biological regulatory networks: a methodology tutorial," *Methods*, vol. 62, no. 1, pp. 3–12, 2013.

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

[28] D. C. Weaver, C. T. Workman, G. D. Stormo *et al.*, "Modeling regulatory networks with weight matrices." in *Pacific Symposium on Biocomputing*, vol. 4, 1999, pp. 112–123.

[29] A. Silvescu and V. Honavar, "Temporal boolean network models of genetic networks and their inference from gene expression time series," *Complex Systems*, vol. 13, no. 1, pp. 61–78, 2001.

[30] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano, "Identification of genetic networks by strategic gene disruptions and gene overexpressions under a boolean model," *Theoretical Computer Science*, vol. 298, no. 1, pp. 235–251, 2003.

[31] 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.

[32] 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.

[33] 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.

[34] 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, 2007.

[35] V. Filkov, "Identifying gene regulatory networks from gene expression data," *Handbook of Computational Molecular Biology*, pp. 27–1–27–29, 2005.

[36] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using bayesian networks to analyze expression data," *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 601–620, 2000.

[37] D. Pe'er, A. Regev, G. Elidan, and N. Friedman, "Inferring subnetworks from perturbed expression profiles," *Bioinformatics*, vol. 17, no. suppl_1, pp. S215–S224, 2001.

[38] 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.

[39] I. Pournara and L. Wernisch, "Reconstruction of gene networks using bayesian learning and manipulation experiments," *Bioinformatics*, vol. 20, no. 17, pp. 2934–2942, 2004.

[40] A. V. Werhli, M. Grzegorczyk, and D. Husmeier, "Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks," *Bioinformatics*, vol. 22, no. 20, pp. 2523–2531, 2006.

[41] L. Xing, M. Guo, X. Liu, C. Wang, L. Wang, and Y. Zhang, "An improved bayesian network method for reconstructing gene regulatory network based on candidate auto selection," *BMC Genomics*, vol. 18, no. 9, p. 844, 2017.

[42] C. E. Manfredotti, "Modeling and inference with relational dynamic bayesian networks." in *Canadian Conference on AI*, vol. 5549. Springer, 2009, pp. 287–290.

[43] 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, 2010.

[44] 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." *DCAI*, vol. 151, pp. 379–386, 2012.

[45] N. X. 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, no. 1, p. 131, 2012.

[46] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Dalla Favera, and A. Califano, "Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context," *BMC bioinformatics*, vol. 7, no. 1, p. S7, 2006.

[47] 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.

[48] A.-C. Haury, F. Mordelet, P. Vera-Licona, and J.-P. Vert, "Tigress: trustful inference of gene regulation using stability selection," *BMC Systems Biology*, vol. 6, no. 1, p. 145, 2012.

[49] S. Biswas and S. Acharyya, "Neural model of gene regulatory network: a survey on supportive meta-heuristics," *Theory in Biosciences*, vol. 135, no. 1-2, pp. 1–19, 2016.

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

[51] M. Wahde and J. Hertz, "Coarse-grained reverse engineering of genetic regulatory networks," *Biosystems*, vol. 55, no. 1, pp. 129–136, 2000.

[52] M. Wahde and J. Hertz, "Modeling genetic regulatory dynamics in

neural development," *Journal of Computational Biology*, vol. 8, no. 4, pp. 429–442, 2001.

[53] R. Xu, D. Wunsch II, and R. Frank, "Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 4, pp. 681–692, 2007.

[54] R. Xu, G. K. Venayagamoorthy, and D. C. Wunsch, "Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization," *Neural Networks*, vol. 20, no. 8, pp. 917–927, 2007.

[55] Y. Zhang, J. Xuan, B. G. de los Reyes, R. Clarke, and H. W. Ressom, "Reverse engineering module networks by pso-rnn hybrid modeling," *BMC Genomics*, vol. 10, no. 1, p. S15, 2009.

[56] I. A. Maraziotis, A. Dragomir, and D. Thanos, "Gene regulatory networks modelling using a dynamic evolutionary hybrid," *BMC Bioinformatics*, vol. 11, no. 1, p. 140, 2010.

[57] 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, 2012.

[58] S. Mandal, "Elephant swarm water search algorithm for global optimization," *Sādhanā*, vol. 43, no. 1, p. 2, 2018.

[59] S. Mandal, G. Saha, and R. K. Pal, "Recurrent neural network-based modeling of gene regulatory network using elephant swarm water search algorithm," *Journal of Bioinformatics and Computational Biology*, vol. 15, no. 04, p. 1750016, 2017.

[60] E. O. Voit, "Biochemical systems theory: a review," *ISRN Biomathematics*, vol. 2013, 2013.

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

[62] P.-K. Liu and F.-S. Wang, "Inference of biochemical network models in s-system using multiobjective optimization approach," *Bioinformatics*, vol. 24, no. 8, pp. 1085–1092, 2008.

[63] M. Vilela, I.-C. Chou, S. Vinga, A. T. R. Vasconcelos, E. O. Voit, and J. S. Almeida, "Parameter optimization in s-system models," *BMC Systems Biology*, vol. 2, no. 1, p. 35, 2008.

[64] T. Nakayama, S. Seno, Y. Takenaka, and H. Matsuda, "Inference of s-system models of gene regulatory networks using immune algorithm," *Journal of Bioinformatics and Computational Biology*, vol. 9, no. supp01, pp. 75–86, 2011.

[65] L. Palafox, N. Noman, and H. Iba, "Reverse engineering of gene regulatory networks using dissipative particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 577–587, 2013.

[66] 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.

[67] R. Thomas, C. J. Paredes, S. Mehrotra, V. Hatzimanikatis, and E. T. Papoutsakis, "A model-based optimization framework for the inference of regulatory interactions using time-course dna microarray expression data," *BMC Bioinformatics*, vol. 8, no. 1, p. 228, 2007.

[68] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita, "Dynamic modeling of genetic networks using genetic algorithm and s-system," *Bioinformatics*, vol. 19, no. 5, pp. 643–650, 2003.

[69] L. Palafox, N. Noman, and H. Iba, "Study on the use of evolutionary techniques for inference in gene regulatory networks," in *Natural Computing and Beyond*. Springer, 2013, pp. 82–92.

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

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

[72] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature inspired cooperative strategies for optimization (NICSO 2010)*, pp. 65–74, 2010.

[73] Y. Maki, T. Ueda, M. Okamoto, N. Uematsu, K. Inamura, K. Uchida, Y. Takahashi, and Y. Eguchi, "Inference of genetic network using the expression profile time course data of mouse p19 cells," *Genome Informatics*, vol. 13, pp. 382–383, 2002.

[74] 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, 2011.

[75] G. Stolovitzky, D. Monroe, and A. Califano, "Dialogue on reverse-engineering assessment and methods," *Annals of the New York Academy of Sciences*, vol. 1115, no. 1, pp. 1–22, 2007.

[76] A. Greenfield, A. Madar, H. Ostrer, and R. Bonneau, "Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models," *PloS One*, vol. 5, no. 10, p. e13397, 2010.

[77] M. Ronen, R. Rosenberg, B. I. Shraiman, and U. Alon, "Assigning numbers to the arrows: parameterizing 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.

Abhinandan Khan received the B.Tech. degree in Electronics and Communication Engineering from the West Bengal University of Technology, India in 2011, and the M.E. degree in Electronics and Telecommunication Engineering from Jadavpur University, India in 2013. He is currently pursuing Ph.D. at the Department of Computer Science and Engineering, University of Calcutta, India. Mr Khan received the University Gold Medal for securing the highest marks among all post-graduate engineering courses at Jadavpur University. He is also a recipient of the Senior Research Fellowship (NET) from the Council of Scientific & Industrial Research (CSIR), Government of India. Mr Khan's research interests include computational biology, bioinformatics, computational intelligence, etc. Mr Khan has published 16 research articles.

Goutam Saha received the B.E. degree in Electrical Engineering and the M.E. degree in Electronics and Telecommunication Engineering from the Bengal Engineering College, Shibpur, under the University of Calcutta, India in 1984 and 1989, respectively. He received the Ph.D. degree from the Indian Institute of Technology, Kharagpur in 1999. He also has Post-Doctoral Research experience at the Indian Institute of Technology, Kharagpur and the Ben Gurion University, Israel between 2000 and 2001. Presently, he is working as a Professor in the Department of Information Technology, North Eastern Hill University, Shillong, and is also holding the position of Dean of the School of Technology at the same university. Dr Saha's current research interests include computational biology, bioinformatics, systems biology, bioremediation, etc. Dr Saha has more than 40 research papers and has authored and co-authored two books. Dr Saha also holds an international patent.

Rajat Kumar Pal received the B.E. degree in Electrical Engineering from the Bengal Engineering College, Shibpur under the University of Calcutta, India, and the M.Tech. degree in Computer Science and Engineering from the University of Calcutta, India in 1985 and 1988, respectively. He received the Ph.D. degree from the Indian Institute of Technology, Kharagpur, India in 1996. Since 1994, he has been as a faculty with the Department of Computer Science and Engineering, University of Calcutta, where he is presently working as a Professor. Dr Pal's major research interests include VLSI design, graph theory and its applications, perfect graphs, logic synthesis, design and analysis of algorithms, computational geometry, parallel computation and algorithms. Dr Pal has published more than 200 research articles and has authored and co-authored two books. Dr Pal also holds several international patents.