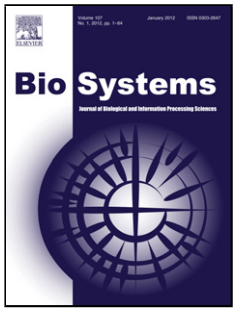

Title: Reverse Engineering Genetic Networks Using Nonlinear Saturation Kinetics

Author: Ahammed Sherief Kizhakkethil Youseph Madhu Chetty Gour Karmakar

PII: S0303-2647(18)30397-6
DOI: https://doi.org/doi:10.1016/j.biosystems.2019.103977
Reference: BIO 103977

To appear in: *BioSystems*

Received date: 16 November 2018
Revised date: 25 April 2019
Accepted date: 27 May 2019

Please cite this article as: Ahammed Sherief Kizhakkethil Youseph, Madhu Chetty, Gour Karmakar, Reverse Engineering Genetic Networks Using Nonlinear Saturation Kinetics, *<![CDATA[BioSystems]]>* (2019), https://doi.org/10.1016/j.biosystems.2019.103977

Reverse Engineering Genetic Networks Using Nonlinear Saturation Kinetics

Ahammed Sherief Kizhakkethil Youseph[a], Madhu Chetty[b], Gour Karmakar[b]

[a]*Faculty of Information Technology, Monash University, Clayton, 3800, Australia.*
[b]*School of Science, Engineering and Information Technology, Federation University Australia, Gippsland, 3842, Australia.*

Abstract

A gene regulatory network (GRN) represents a set of genes along with their regulatory interactions. Cellular behavior is driven by genetic level interactions. Dynamics of such systems show nonlinear saturation kinetics which can be best modeled by Michaelis-Menten (MM) and Hill equations. Although MM equation is being widely used for modeling biochemical processes, it has been applied rarely for reverse engineering GRNs. In this paper, we develop a complete framework for a novel model for GRN inference using MM kinetics. A set of coupled equations is first proposed for modeling GRNs. In the coupled model, Michaelis-Menten constant associated with regulation by a gene is made invariant irrespective of the gene being regulated. The parameter estimation of the proposed model is carried out using an evolutionary optimization method, namely, trigonometric differential evolution (TDE). Subsequently, the model is further improved and the regulations of different genes by a given gene are made distinct by allowing varying values of Michaelis-Menten constants for each regulation. Apart from making the model more relevant biologically, the improvement results in a decoupled GRN model with fast estimation of model parameters. Further, to enhance exploitation of the search, we propose a local search algorithm based on hill climbing heuristics. A novel mutation operation is also proposed to avoid population stagnation and premature convergence. Real life benchmark data sets generated in vivo are used for validating the proposed model. Further, we also analyze realistic in silico datasets generated using GeneNetweaver. The comparison of the performance of proposed model with other existing methods shows the potential of the proposed model.

Keywords: Gene Regulatory Network, Michaelis-Menten Kinetics, Reverse Engineering, Mixed Integer Nonlinear Programming, Trigonometric Differential Evolution, Local Search

1. Introduction

The advances in the field of microarray technology enabled biologists to measure the expression levels of a large number of genes at a time. In turn, this facilitated the computational biologists in accurately modeling and analyzing large-scale gene regulatory networks (GRN). Various inference methods typically make use of the time-series data obtained from the set of transient expression levels of genes measured at different time intervals (in time series) in an experiment. A typical gene expression dataset is characterized by a large number of genes and a few samples. With the increase of the number of genes, the search space for finding the network structure grows exponentially. This problem and unavailability of sufficient data, make the GRN inference a difficult and challenging problem. The number of datasets is shown to have a significant influence on the inference performance (Xenitidis et al., 2017).

Several approaches have been proposed in literature for reverse engineering of GRNs. The key components of the reconstruction process is a modeling framework and a method of learning the parameters of the model. One of the first models proposed for GRN inference is Boolean network (Kauffman, 1969). Boolean networks are simple logic models and learning is computationally fast. However, the model is limited to work with binary values of gene expression data and hence fails to represent the multiple values of gene expressions. Although continuous-valued Boolean networks are being proposed recently for GRN reconstruction (Grieb et al., 2015), the linearity of the model contradicts with the nonlinear nature of GRNs. Bayesian networks (BN), one of the popular models used for GRNs, are capable of representing fractional values of gene expression data (Friedman et al., 2000a,b). But, BNs and dynamic BNs (Yu et al., 2017) work only with discretized data and cannot model system dynamics in continuous time. Similar to Boolean networks, BNs are also linear models and their biological relevance is debatable. GRN inference from time-series gene expression data has been accomplished by using a number of non-parametric methods such as Jump3 (Huynh-Thu and Sanguinetti, 2015) which is based on decision tree, dynGENIE3 (Huynh-Thu and Geurts, 2018) which is based on random forest and lasso (Nguyen and Braun, 2018) which is based on regularized regression.

Various models based on linear ordinary differential equations are also used for GRN reconstruc-

tion (Lu et al., 2011; Zhu et al., 2012; Zheng et al., 2012). However, again, they fail to assimilate the nonlinear behavior of biological systems. State space models (Hirose et al., 2008; Tamada et al., 2011), which are capable of representing system with a very few variables, have been considered to be adequate to deal with problems related to less number of samples in the data. However, the biological relevance of these linear state space models is inadequate. Neural network models used for GRN reconstruction include artificial neural networks (Rubiolo et al., 2018) and recurrent neural networks (Kordmahalleh et al., 2017; Chen, 2017; Khan et al., 2018). It is believed that as computational demand becomes less challenging, neural networks will become more promising (Barbosa et al., 2018). Among the models proposed for GRN reconstruction, nonlinear state space models (Noor et al., 2012) and some recurrent neural network models (Wahde and Hertz, 2000) represent nonlinearity. However, both these approaches are limited to simple nonlinear functions and fail to represent the complex molecular mechanisms underlying GRNs. In general, the identification of nonlinear systems is a cumbersome task and often employ selected nonlinear functions in the parameter estimation processes (Pan et al., 2016).

Needless to say, genetic interactions are best modeled by nonlinear equations. Among the few nonlinear models used for reverse engineering GRNs, S-system has been viewed to be suitable for the biological nature of genetic networks (Maki et al., 2001; Chen et al., 2017). Although the S-system models can represent a biological system more closely than the linear models, because of the power-law functions, they still fail to capture the saturation kinetic behavior of biochemical processes precisely. Experimental studies reported in the literature (Yagil and Yagil, 1971; Bintu et al., 2005; Kaplan et al., 2008) help to understand the variation in gene expression levels as a function of the gene's inducers. Such studies usually show the variation in gene expression levels as nonlinear saturation kinetics (Yagil and Yagil, 1971; Bintu et al., 2005; Kaplan et al., 2008). Moreover, zero or near-zero values of gene expressions are very often found in microarray datasets, but the S-system models are not suitable to be applied for such datasets.

Michaelis-Menten equation is well accepted in modeling biochemical reactions and is widely used over a century for engineering biological systems (Choi et al., 2017). Van den Bulcke et al. (2006) generated synthetic gene expression data by modeling the interaction kinetics using Michaelis-Menten

and Hill kinetics. Further, saturation kinetics can also be effectively modeled by Michaelis-Menten and Hill kinetics. Despite being widely accepted and employed in modeling biological systems including GRNs, Michaelis-Menten kinetics in its fundamental form has not been exploited for reverse engineering GRNs.

In this paper, we apply Michaelis-Menten kinetics and develop a new nonlinear ordinary differential equation (ODE) model which is biologically relevant and thus suitable for obtaining realistic GRNs. The model identification is formulated as a mixed-integer nonlinear programming (MINLP) problem so that, both the structural complexity of the network and deviation of the experimental data from the model predictions are minimized. The optimization is carried out using an evolutionary algorithm called trigonometric differential evolution (TDE) which is a variant of differential evolution. To speed up the convergence, the optimization uses local search heuristics and the flip operation similar to that have been reported for S-system modeling by Noman and Iba (2007) and Chowdhury and Chetty (2011), respectively.

The identification of systems described by nonlinear ordinary differential equation(ODE) models is computationally expensive due to the high cost involved in numerical integration. The parameters involved in such models (e.g. S-system) used in GRN inference are usually estimated in a step by step method (Maki et al., 2002). The ODE describing the rate of change of single gene's expression is considered at a time, while the expression values of other genes are given as input to the estimation algorithm. Thus, the parameters associated with the expression rate of each gene are estimated one by one. In the simpler version of the model (Youseph et al., 2015a) that is being proposed here, we had considered the variable representing the Michaelis-Menten dynamics between the regulating and regulated gene as a constant leading to model with coupled differential equations. Since the parameters appeared coupled, gene-by-gene parameter estimation was not possible. In (Youseph et al., 2015b), we replaced this constant with a variable enabling us to decouple the differential equations.

In this paper, we have reduced the number of decision variables for optimization by coupling a set of linked-parameters in the model. The fitness function that is minimized for estimation of parameters is modified to reflect the trade-off between model prediction accuracy and the scale-free

nature of real genetic networks. A hill-climbing local-search algorithm that is specific to the proposed model is developed to reduce possible false predictions. Moreover, a flip operation specific to the proposed model is developed to avoid stagnation and premature convergence. The algorithm is coded in Matlab and simulations are extended to networks of higher sizes including in vivo and in silico networks. The inferred networks are evaluated using the commonly used performance metrics, i.e., sensitivity, specificity, precision and F-score and are compared with the current existing methods. The results show that the proposed model is competent to all the existing methods and is outperforming all in some cases.

The rest of the paper is organized as follows: Section 2 presents the background of the model. The methods of model development and model identification are described in Section 3. The results are presented and discussed in Section 4. Finally Section 5 concludes the paper.

2. Background

The activities taking place in living systems are the result of various biochemical reactions. A general biochemical reaction is the formation of a molecule, referred as product, from another molecule, referred as substrate. These reactions are catalyzed by biological catalysts, called enzymes. In gene transcription, the primary enzyme is RNA polymerase and the product is mRNA. Substrates are usually not involved here except that the process is usually regulated by transcription factors.

The binding interaction between an enzyme and a substrate was first discovered by Victor Henri (Henri, 1903) in 1903. Michaelis and Menten took up Henri's work of enzyme kinetics and proposed a mathematical model of enzyme kinetics (Michaelis and Menten, 1913) in 1913. Interestingly, both these works are more often cited in the 21st century than they were in the 20th (Cornish-Bowden, 2015). There are following two ways to arrive at the Michaelis-Menten equation.

i The binding reaction, i.e., the first step, is very fast and attains equilibrium quickly. This assumption, called rapid equilibrium approach was used by Michaelis and Menten in deriving the equation (Michaelis and Menten, 1913). Later, G. E. Briggs and J. B. S. Haldane derived the equation, using

ii Quasi-steady state approximation (QSSA), i.e. the enzyme-substrate complex comes to the steady state rapidly (Shuler and Kargi, 2002). Briggs and Haldane suggested that QSSA holds when

the concentration of the enzyme is much less than the concentration of the substrate (or ligand) (Shuler and Kargi, 2002). This is called free-ligand approximation.

The MM constant determined in the proposed method would be from any of the two methods unless the information is provided while generating the data.

When the formation of a product P is activated by an activator A, the rate of formation of the product, denoted by v, can be expressed using Michaelis-Menten kinetics as follows:

$$v = \frac{d[P]}{dt} = v_{max} \frac{[A]}{K_A + [A]} f_1 \tag{1}$$

where, $[P]$ is the concentration of the product P, v_{max} is the maximum production rate, $[A]$ is the concentration of the activator A, K_A is Michaelis-Menten constant which accounts for the affinity of the activator binding and f_1 is a function of other factors affecting the reaction. On the other hand, if the product formation is inhibited by an inhibitor I, v can be expressed as follows:

$$v = \frac{d[P]}{dt} = v_{max} \frac{K_I}{K_I + [I]} f_2 \tag{2}$$

where, $[I]$ is the concentration of the inhibitor I, K_I is Michaelis-Menten constant which accounts for the affinity of the inhibitor binding, f_2 is a function of other factors affecting the reaction. If the product formation is both activated by A and inhibited by I at the same time, v can be expressed as follows:

$$v = \frac{d[P]}{dt} = v_{max} \left(\frac{K_I}{K_I + [I]} \right) \left(\frac{[A]}{K_A + [A]} \right) f_3 \tag{3}$$

We could model biochemical systems according to Eq. (3) if the regulations were known a priori. For example, Arkin et al. (1998) and Chen et al. (2004), using principles of chemical kinetics, modeled the system where the molecular interactions are known or presumed. Such models can be employed for various *in silico* system analyses such as stability analysis and what-if analysis. However, in reverse engineering problems such as inferring GRNs, the regulations are unknown and applying this kinetics are not straightforward.

3. Methods

3.1. The Proposed Model

3.1.1. Michaelis-Menten Coupled GRN Model

Development of the model essentially involves introducing new variables to account for all the possibilities of regulations between two genes in a network. In our case, broadly two cases are possible, i.e., regulation and no regulation, which can be captured by using one binary variable, i.e., $p = 1$ or 0 which indicates the existence or absence of the regulation, respectively. The regulations can further be categorized into two types: activation and inhibition, represented respectively by 1 and 0 with another variable q. In a summation model, the binary variables can be used as the coefficients of the terms being added. However, GRN model, being nonlinear, requires a product form. The method of formulation in integer programming for this case is to use the binary variables as the powers of the terms being multiplied.

The general approach of formulating such a model would be the following:

$$\frac{dy_i}{dt} = v_i^{max} \prod_{j=1}^{N} \left[\left(\frac{y_j}{K_j + y_j} \right)^{q_{ij}} \left(\frac{K_j}{K_j + y_j} \right)^{1-q_{ij}} \right]^{p_{ij}} - v_i^d y_i \tag{4}$$

where, y_i is the concentration of mRNA expressed by gene-i, v_i^{max} is the maximum rate of expression of gene-i, v_i^d is the self decay rate of mRNA expressed by gene-i, p_{ij} and q_{ij} are two binary variables that represent the presence/absence of a regulation and the type of the regulation, respectively in regard to the regulation from gene-j to gene-i and K_j is the Michaelis-Menten constant associated with gene-j to gene-i regulation kinetics. This equation encompasses all the possibilities discussed in the beginning of this section. For example, $p_{ij} = 1$ and $q_{ij} = 1$ give the same fractional term as in Eq. (1). From the observation that the denominator remains the same in both the forms of regulation and a sum of both terms indicates no regulation, we come up with a much simpler model as detailed below.

The terms to be multiplied in the production term in all possible cases of regulation and non-regulation from gene-j to gene-i are shown in Table 1.

Table 1 The terms to be multiplied for possibilities of regulation from gene-j to gene-i

Type of Regulation	Term to be multiplied
Activation	$\frac{y_j}{K_j+y_j}$
Inhibition	$\frac{K_j}{K_j+y_j}$
No regulation	$1=\frac{K_j+y_j}{K_j+y_j}=\frac{y_j}{K_j+y_j}+\frac{K_j}{K_j+y_j}$

From the above observation, for a system of N genes, we define, the Michaelis-Menten Coupled GRN (MMC-GRN) model equation as follows:

$$\begin{aligned}\dot{y}_i=\frac{dy_i}{dt}=&v_i^{max}\prod_{j=1}^{N}\left(g_{ij}\frac{y_j}{K_j+y_j}+h_{ij}\frac{K_j}{K_j+y_j}\right)-v_i^d y_i\\ =&v_i^{max}\prod_{j=1}^{N}\left(\frac{g_{ij}y_j+h_{ij}K_j}{K_j+y_j}\right)-v_i^d y_i\end{aligned}\tag{5}$$

where, v_i^{max} is the maximum rate of expression of gene-i

$h_{ij},g_{ij}\in\{0,1\},1\le h_{ij}+g_{ij}\le 2\ \forall\ i,j=1,\ldots,N$

$h_{ij}=1,g_{ij}=1$ implies that there is no regulation from gene-j to gene-i

$h_{ij}=0,g_{ij}=1$ implies that gene-j activates gene-i

$h_{ij}=1,g_{ij}=0$ implies that gene-j inhibits gene-i

K_j is Michaelis-Menten constant which indicates the binding affinity; the smaller the value of K_j, the larger the binding affinity is.

v_i^d is the self decay rate of mRNA expressed by gene-i.

To demonstrate the modeling process, let us consider a simple example of a 5-gene network shown in Figure 1. Gene-4 is inhibited by gene-2 and activated by gene-3. The rate of change of gene expression for gene-4 can be written by the following general equation:

$$\begin{aligned}\frac{dy_4}{dt}=&v_4^{max}\prod_{j=1}^{5}\left[\frac{g_{ij}y_j+h_{ij}K_j}{K_j+y_j}\right]-v_4^d y_4\\ =&v_4^{max}\left(\frac{g_{41}y_1+h_{41}K_1}{K_1+y_1}\right)\left(\frac{g_{42}y_2+h_{42}K_2}{K_2+y_2}\right)\left(\frac{g_{43}y_3+h_{43}K_3}{K_3+y_3}\right)\left(\frac{g_{44}y_4+h_{44}K_4}{K_4+y_4}\right)\\ &\times\left(\frac{g_{45}y_5+h_{45}K_5}{K_5+y_5}\right)-v_4^d y_4\end{aligned}\tag{6}$$

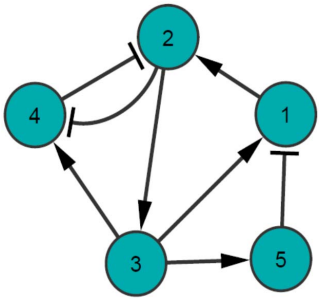

Figure 1 A simple GRN of five genes. Arrow ended arcs indicate activation and block ended arcs indicate inhibition.

This is because there are no regulations from gene-1 to gene-4, gene-4 to gene-4 or gene-5 to gene-4, $g_{41} = h_{41} = g_{44} = h_{44} = g_{45} = h_{45} = 1$. Since the regulation from gene-2 to gene-4 is inhibition, $g_{42} = 0$ and $h_{42} = 1$. Similarly, gene-3 activates gene-4, hence $g_{43} = 1$ and $h_{43} = 0$. Now substituting all the g and h values in the above equation, the equation reduces to the following form which is analogous to the equation (3).

$$\frac{dy_4}{dt} = v_4^{max}\left(\frac{K_2}{K_2 + y_2}\right)\left(\frac{y_3}{K_3 + y_3}\right) - v_4^d y_4 \tag{7}$$

The model parameters h_{ij} and g_{ij}, being binary parameters, provide four (2^2) combinations. However, by the model definition, both these parameters cannot be zero at a time for any i and j. This means that we are left with three options for h_{ij} and g_{ij}, i.e., (1,0), (0,1) and (1,1). The total number of parameters for a system of N genes according to Eq. (5) is $N(1+1+1+N+N) = 2N^2 + 3N$. The lesser the number of parameters, the cheaper will be the computational cost for estimation of the parameters. In order to reduce the computational complexity of the estimation algorithm, instead of h_{ij} and g_{ij}, we can estimate a new variable $\delta_{ij} \in \{-1, 0, 1\}$ as defined below:

$$\delta_{ij} = \begin{cases} 0, \text{if } g_{ij} = 1, h_{ij} = 1 \\ +1, \text{if } g_{ij} = 1, h_{ij} = 0 \\ -1, \text{if } g_{ij} = 0, h_{ij} = 1 \end{cases} \tag{8}$$

The parameter δ_{ij} can be expressed as a function of g_{ij} and h_{ij}, and vice versa, as follows:

$$\delta_{ij} = g_{ij} - h_{ij} \tag{9}$$

$$g_{ij} = \frac{\delta_{ij} + 1}{\mid \delta_{ij} \mid + 1} \tag{10}$$

$$h_{ij} = \frac{1 - \delta_{ij}}{\mid \delta_{ij} \mid + 1} \tag{11}$$

Substituting for g_{ij} and h_{ij} in Eq. (5), we obtain the following equation:

$$\frac{dy_i}{dt} = v_i^{max} \prod_{j=1}^{N} \left(\frac{\frac{\delta_{ij}+1}{|\delta_{ij}|+1} y_j + \frac{1-\delta_{ij}}{|\delta_{ij}|+1} K_j}{K_j + y_j} \right) - v_i^d y_i \tag{12}$$

For a system of N genes, the parameters are $v^{max} \in \mathbb{R}^{N\times 1}$, $v^d \in \mathbb{R}^{N\times 1}$, $K \in \mathbb{R}^{N\times 1}$ and $\delta \in Z^{N\times N}$ and hence, Eq. (12) requires estimating the values of N instances of v^{max}, K and v^d and N^2 instances of delta which can be used subsequently to determine the values of g and h. Therefore, the total number of parameters to be estimated for MMC-GRN model is $(N+N+N+N^2) = (3N+N^2)$ where, N^2 parameters are integer variables.

In order to evaluate the number of parameters to be estimated in the inference process, we write the S-system model in Eq. (13):

$$\dot{y}_i = \frac{dy_i}{dt} = \alpha_i \prod_{j=1}^{N} y_j^{g'_{ij}} - \beta_i \prod_{j=1}^{N} y_j^{h'_{ij}} \tag{13}$$

The number of parameters to be estimated in S-system is $2N + 2N^2$ which is $(2N + 2N^2) - (3N + N^2) = (N^2 - N)$ parameters more than that of MCC-GRN model.

3.1.2. Michaelis-Menten Decoupled GRN Model

In the coupled model, for each regulatory gene, the value of K_j is fixed irrespective of the gene being regulated, i.e., the binding affinity of the regulatory protein from gene-j is considered to be the same for all the genes it regulates. For example, if gene-1 regulates gene-2 and gene-3, the binding affinity of the protein from gene-1 towards gene-2 and towards gene-3 are assumed to be the same. This

$$\begin{array}{c} 1 \\ 2 \\ - \\ i \\ - \\ N \end{array} \begin{pmatrix} v_1^{max} & v_1^d & \delta_{11} & - & \delta_{1N} & K_1 & - & K_N \\ v_2^{max} & v_2^d & \delta_{21} & - & \delta_{2N} & K_1 & - & K_N \\ - & - & - & - & - & - & - & - \\ v_i^{max} & v_i^d & \delta_{i1} & - & \delta_{iN} & K_1 & - & K_N \\ - & - & - & - & - & - & - & - \\ v_N^{max} & v_N^d & \delta_{N1} & - & \delta_{NN} & K_1 & - & K_N \end{pmatrix}$$

Figure 2 The parameters to be estimated for each gene in MMC-GRN model defined in Eq. (5)

assumption is correct when one gene regulates only one gene. However, this is not the case in reality, especially in large scale networks, where a gene is more likely to regulate multiple genes.

In order to get better biological relevance with respect to MMC-GRN, we propose a modification in MMC-GRN model by allowing differing values for the Michaelis-Menten constant for each regulation irrespective of the regulatory gene. If a gene-i is regulating gene-j and gene-k, two different constants K_{ji} and K_{ki} are used to differentiate the binding of protein from gene-i to the promoter region of gene-j from its binding to the promoter region of gene-k. The model defined in Eq. (5) is modified as:

$$\dot{y}_i = v_i^{max} \prod_{j=1}^{N} \left(\frac{g_{ij} y_j + h_{ij} K_{ij}}{K_{ij} + y_j} \right) - v_i^d y_i \ . \tag{14}$$

where, K_{ij} is Michaelis-Menten constant for the regulation of gene-i by gene-j if the regulation exists and other constants are same as in Eq. (5).

Substituting g and h in Eq. (14) with functions of δ gives the following:

$$\frac{dy_i}{dt} = v_i^{max} \prod_{j=1}^{N} \left(\frac{\frac{\delta_{ij}+1}{|\delta_{ij}|+1} y_j + \frac{1-\delta_{ij}}{|\delta_{ij}|+1} K_{ij}}{K_{ij} + y_j} \right) - v_i^d y_i \ . \tag{15}$$

In the coupled model defined in Eq. (5), the parameters to be estimated for each gene are shown in Fig. 2 where in each equation the parameters are different except $K_1, K_2, \ldots, K_N$ that are the same for all N genes. As mentioned before, the total number of parameters to be estimated in this MMC-GRN model is $N(2+N)+N = N^2+3N$ and they all have to be estimated together.

Since we have modified the model, the parameters for each gene in the new model defined in Eq. (14) are shown in Fig. 3. None of the parameters are coupled for different genes and hence the model defined in Eq. (14) can be decoupled, thereby allowing estimation of parameters to be carried out for

$$\begin{array}{c} 1 \\ 2 \\ - \\ i \\ - \\ N \end{array} \begin{pmatrix} v_1^{max} & v_1^d & \delta_{11} & - & \delta_{1N} & K_{11} & - & K_{1N} \\ v_2^{max} & v_2^d & \delta_{21} & - & \delta_{2N} & K_{21} & - & K_{2N} \\ - & - & - & - & - & - & - & - \\ v_i^{max} & v_i^d & \delta_{i1} & - & \delta_{iN} & K_{i1} & - & K_{iN} \\ - & - & - & - & - & - & - & - \\ v_N^{max} & v_N^d & \delta_{N1} & - & \delta_{NN} & K_{N1} & - & K_{NN} \end{pmatrix}$$

Figure 3 The parameters to be estimated for each gene in MMD-GRN model defined in Eq. (14)

one gene at a time. The proposed decoupled model, namely MMD-GRN can finally be expressed as follows:

$$\frac{dy_i}{dt} = v_i^{max} \left(\frac{g_{ii} y_i + h_{ii} K_{ii}}{K_{ii} + y_i} \right) \prod_{\substack{j=1 \\ j \neq i}}^{N} \left(\frac{g_{ij} y_j^* + h_{ij} K_{ij}}{K_{ij} + y_j^*} \right) - v_i^d y_i \quad . \tag{16}$$

where, y_j^* are the expression levels of genes other than gene-i. They can be determined using any numerical interpolation method before the parameter estimation of gene-i is calculated. Figure 3 shows that each gene has $2+N+N = 2(N+1)$ parameters. Therefore, total number of parameters that are required to be estimated for N genes is $2N(N+1)$.

It may be noted that K_{ij} (Michaelis-Menten constant) is nonnegative and the connectivity parameter δ_{ij} can have only three values, -1, 0 or +1. This observation enables us to consider a single entity $K_{ij}\delta_{ij}$ during estimation.

$$K_{min} \leq K_{ij} \leq K_{max} \tag{17}$$

$$-K_{max} \leq K_{ij}\delta_{ij} \leq K_{max} \tag{18}$$

$$K = |K_{ij}\delta_{ij}| \tag{19}$$

$$\delta_{ij} = \begin{cases} 0, & \text{if } K_{ij}\delta_{ij} = 0 \\ \frac{K_{ij}\delta_{ij}}{|K_{ij}\delta_{ij}|}, & \text{otherwise.} \end{cases} \tag{20}$$

Hence the number of parameters is reduced by N^2 as above. As a result, the total number of parameters to be estimated for the MMD-GRN model is $2N+N^2$ which is less than that of S-system model. Figure 4 shows the comparison of the parameters to be estimated for the MMD-GRN model with those of the MMC-GRN model and S-system. Although the curves for MMC-GRN and MMD-

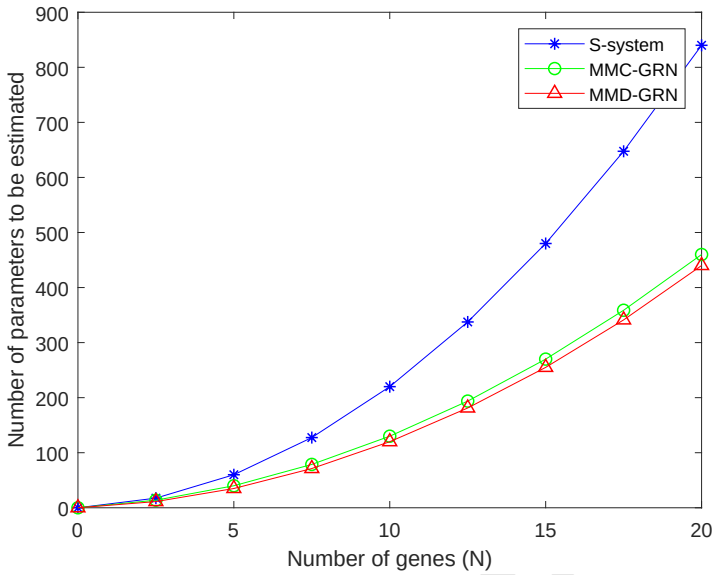

Figure 4 Comparison of the number of parameters to be estimated in MMD-GRN with that in MMC-GRN and S-system

GRN seem to be overlapping each other, in reality, the latter one is slightly below the former one. However, the plot of S-system shows that the number of parameters to be estimated in S-system model is significantly higher that that for both the MM models for large scale networks.

3.2. Estimation of Parameters

The parameter estimation is carried out using trigonometric differential evolution algorithm by which the fitness function is optimized.

3.2.1. Fitness Function

The fitness function considers two factors: (i) model prediction error and (ii) penalty on model complexity. As these two terms are two entities of non-comparable magnitudes, Chowdhury et al. (2013) suggested a balancing factor. We considered the fitness function for inferring MMD-GRN as the sum of absolute square error and a balanced penalty term which is analogous to the penalty term proposed

by Chowdhury et al. (2013). The objective function minimized for each gene is the following.

$$f_i(y_i, \delta_{i1}, \delta_{i2}, \ldots, \delta_i) = \sum_{j=1}^{T} \left\{ \frac{y_{ij}^c - y_{ij}^e}{y_{ij}^e} \right\}^2 + B_i \times C_i \times \frac{N}{(N - r_i)} \tag{21}$$

where, r_i is the total number of regulations for gene-i and can be calculated as follows:

$$r_i = \sum_{j=1}^{N} \mid \delta_{ij} \mid \tag{22}$$

B_i is the balancing factor(Chowdhury et al., 2013) which balances the two terms (SRE and penalty on model complexity) on the R.H.S. of Eq. (21) and C_i is the penalty factor, as defined by Chowdhury et al. (2013)

$$C_i = \begin{cases} 1, \text{ if } J_i < r_i < I_i \\ 1 + (J_i - r_i)^2, \text{ if } r_i \leq J_i \\ 1 + (r_i - I_i)^2, \text{ if } r_i \geq I_i \end{cases} \tag{23}$$

where, J_i and I_i respectively, are the minimum and maximum cardinalities of gene-i . Following Chowdhury et al. (2013), the balancing factor (B_i) is adaptively chosen during the optimization. At the beginning of the optimization, to allow the optimization to minimize the prediction error produced by the model without considering the model complexity, B_i is assigned a value of zero. After a fixed number of generations is reached or the prediction error goes below a threshold value, both the terms on R.H.S. of Eq. (21), the model error (SRE) and the model complexity $\left(\frac{N}{N-r_i}\right)$, are evaluated for the best 50% of the population. B_i is fixed as the ratio of their averages.

$$B_i = \frac{\bar{E}}{\bar{C}} \tag{24}$$

where, $\bar{E}$ is the average model error and $\bar{C}$ is the average model complexity.

3.2.2. Problem Formulation

The estimation of parameters in the equation for each gene gets formulated as a mixed integer non-linear programming (MINLP) problem.

$$\begin{aligned} & \min_{v_i^{max}, v_i^d, K_{ij}, \delta_{ij}} \quad f_i(y_i, \delta_{ij}) \quad j = 1, 2, \ldots, N \\ \textit{subject to} \quad & \frac{dy_i}{dt} = f'(y_j, v_i^{max}, v_i^d, K_{ij}, \delta_{ij}), \\ & y_i(t_o) = y_{i0}, \quad j = 1, 2, \ldots, N \\ & v_i^{max}, v_i^d, K_{ij}, y_i(t) \geq 0, \delta_{ij} \in \{-1, 0, 1\} \end{aligned} \tag{25}$$

where, f_i is the function defined in equation Eq. (21), f' is the R.H.S. of equation Eq. (15). Otero-Muras et al. (2016) developed a software toolbox, SYNBADm, that finds biocircuit designs for given specifications. SYNBADm includes global mixed-integer nonlinear programming (MINLP) optimization solvers.

3.2.3. Optimization Algorithm

A variant of differential evolution (DE), trigonometric differential evolution (TDE) is applied for estimation of the parameters. As the parameter estimation is carried out for each gene separately, each individual of the population in optimization for each gene consists of $(2+N)$ parameters to be learned, i.e., for gene-i, an individual Ind is given by $Ind = [v_i^{max}, v_i^d, K_{i1}\delta_{i1}, K_{i2}\delta_{i2}, \ldots, K_{iN}\delta_{iN}]$. The DE algorithm consists of the following steps:

1. Initialization: Initialize the parent population of N_p individuals, N_p is the population size. In our case, each individual $\in \mathbb{R}^{(N+2)\times 1}$
2. Fitness Evaluation: Evaluate the fitness (objective function being optimized) of each individual in the population.
3. Offspring Generation: Generate a new (children) population by mutation and crossover.
4. Fitness Evaluation: Evaluate the fitness of each individual in the child population.
5. Parent Selection: The fitness of each individual in the parent population is compared with that of the corresponding individual in the child population to select the best as the new parent.

6. Termination: If the stopping criterion is met, return the best solution. Otherwise, go to step 3.

A simple initialization procedure is to generate individuals randomly using the lower and upper bounds on decision variables as follows:

$$w_i^{g=0} = w_{min} + \rho_i(w_{max} - w_{min}), \quad \rho_i \in [0,1] \quad i = 1,2,\ldots,N_p \tag{26}$$

where, ρ_i is a uniformly distributed random number and w_{min} and w_{max} are the lower and upper bounds on the decision variables.

Several mutation operations are proposed in literature. A simple mutation operation proposed in (Storn and Price, 1997), DE/Rand/1/Bin, is as follows:

$$u_i = w_{r_1}^g + F_m(w_{r_2}^g - w_{r_3}^g), \quad r_1, r_2, r_3 \in \{1,2,\ldots,N_p\} \quad i \neq r_1 \neq r_2 \neq r_3 \; i = 1,2,...,N_p \tag{27}$$

where, F_m is the mutation factor and r_1, r_2 and r_3 are any three different random numbers from $\{1,2,\ldots,N_p\}$. Thus, a random member is perturbed by taking the difference between any other two randomly selected members of the current generation, scaling it by F_m and adding it to the random member. The mutation factor is considered to be the most sensitive parameter in DE (Yang, 2014). Although the accepted value of the mutation factor (F_m) is $\in [0,2]$ (Storn and Price, 1997), $F_m \in [0,1]$ is more efficient, $F_m \in [0.4,0.95]$ is good with a first choice $F_m \in [0.7,0.9]$ (Yang, 2014).

In the crossover operation, the new members z_i of the offspring generation are created by randomly selecting elements $u_i(j)$ and $w_i^g(j)$ from the perturbed vector, u_i and the current (parent) generation, w_i^g, respectively based on a user-defined crossover ratio as follows (Storn and Price, 1997):

$$z_i(j) = \begin{cases} u_i(j), \text{if } rand1_j \leq CR, \quad CR \in [0,1] \\ w_i^g(j), \text{otherwise} \end{cases} \qquad j = 1,2,\ldots,N+2; \quad i = 1,2,\ldots,N_p \tag{28}$$

where, $rand1_j$ is a random number $\in [0,1]$ and $CR \in [0,1]$ is a user-defined crossover factor. The higher the value of CR, the more diverse will be the search. A good range would be $CR \in [0.1,0.8]$ with a first choice $CR = 0.5$ (Yang, 2014).

The next step is the evaluation and selection process. For the two generations, namely parent and children, the objective functions are evaluated. From the parent and children generations, for each member, w_i^g and z_i, the new parent generation is selected as follows:

$$w_i^{g+1} = \begin{cases} z_i, \text{if } f(z_i) \text{ is better than } f(w_i^g) \\ w_i^g, \text{otherwise} \end{cases} \tag{29}$$

Trigonometric Differential Evolution (TDE) employs a Trigonometric Mutation Operation (TMO) propsed by Fan and Lampinen (2003) for accelerating convergence. A random number, $rand2_j \in [0,1]$ is generated and if $rand2_j > M_t$ (where M_t is the trigonometric mutation ratio) the normal mutation is carried out and otherwise the trigonometric mutation is carried out. The TMO is done as follows (Fan and Lampinen, 2003):

$$\begin{aligned} u_i = \frac{w_{r_1}^g + w_{r_2}^g + w_{r_3}^g}{3} + (p_2 - p_1)(w_{r_1}^g - w_{r_2}^g) \\ + (p_3 - p_2)(w_{r_2}^g - w_{r_3}^g) + (p_1 - p_3)(w_{r_3}^g - w_{r_1}^g) \end{aligned} \tag{30}$$

$$\text{where,} p_1 = \frac{|f(w_{r_1}^g)|}{p'}, \quad p_2 = \frac{|f(w_{r_2}^g)|}{p'}, \quad p_3 = \frac{|f(w_{r_3}^g)|}{p'}$$

$$p' = |f(w_{r_1}^g)| + |f(w_{r_2}^g)| + |f(w_{r_3}^g)| \tag{31}$$

Because of the good convergence rate and efficiency in optimizing nonlinear GRN models, TDE is preferred over simple DE in many S-system model-based GRN inference methods (Noman and Iba, 2006, 2007; Chowdhury, 2014). After subjective analysis, we also have chosen TDE to implement in this research.

3.2.4. Local Search Heuristics

Noman and Iba (2007) proposed a local search, namely Hill-Climbing Local Search (HCLS), which exploits the sparse nature of GRNs. HCLS was developed for inference of GRNs using S-system model where the possible non-regulations are traced out based on the value of kinetic orders. A similar concept has been adopted for our model based on the following observations.

If gene-*j* activates the expression of gene-*i*, the fraction being multiplied to the production term of

the rate equation for expression of gene-i is given below:

$$\frac{y_j}{K_{ij}+y_j} = \frac{1}{1+\frac{K_{ij}}{y_j}} \tag{32}$$

In this case, if $K_{ij} \ll y_j$, $\frac{K_{ij}}{y_j} \to 0$ which means that there is no regulation. If gene-j inhibits the expression of gene-i, the fraction being multiplied to the production term of the rate equation, Eq. (14), for expression of gene-i is given below:

$$\frac{K_{ij}}{K_{ij}+y_j} = \frac{1}{1+\frac{y_j}{K_{ij}}} \tag{33}$$

In this case, if $K_{ij} \gg y_j$, $\frac{y_j}{K_{ij}} \to 0$ which means that there is no regulation.

To eliminate the possible non-regulations identified as regulations in the candidate individuals of the current population, we sort the regulations based on the following weight function:

$$w_{ij}\left(K_{ij}, \bar{y}_j, \delta_{ij}\right) = \left(\frac{K_{ij}}{\bar{y}_j}\right)^{\delta_{ij}} \tag{34}$$

where, $\bar{y}_j$ is the mean value of expression of gene-j over the given samples of time. For the best 10% of the population in each iteration, our proposed MM Hill-Climbing Local Search (MMHCLS) is applied. For each of the selected individuals, the procedure described in Algorithm 1 is carried out. The regulatory weights of the arcs in the individual (*Ind*) of the population are sorted in ascending order (Step 1). The arc having the lowest value of weight function is removed from *Ind* (i.e., assigned a value of zero) if the fitness evaluated by removing it shows an improvement. This procedure is repeated (N-1) more times for each $K_{ij}\delta_{ij}$ in the order sorted by weight function (Step 2 to Step 8).

To evaluate the impact of MMHCLS, two sets of simulations were run to infer a five-gene in silico network using the TDE algorithm, one with MMHCLS and the other without MMHCLS. The value of the best objective function in each generation is plotted in Figure 5 for both the simulations. It may be noted that the search exploits the local optima and comes to the global optima quickly by using MMHCLS. However, the traditional TDE without any heuristics gets stuck at a local optimum.

Algorithm 1 MM Hill-Climbing Local Search

Require: Ind, the candidate for which local search is to be carried out.
Ensure: Ind, updated through local search execution.

1: Sort the regulations in the ascending order of confidence based on the weight, $w_{i1}(\delta_{i1}, K_{i1}, \bar{y}_1) \leq w_{i2}(\delta_{i2}, K_{i2}, \bar{y}_2) \leq \ldots, w_{iN}(\delta_{iN}, K_{iN}, \bar{y}_N)$
2: **for** $j = 1$ to N **do**
3: $\quad Ind' \leftarrow Ind$.
4: $\quad K_{ij}\delta_{ij}(Ind') \leftarrow 0$.
5: $\quad$ **if** $f_i(Ind') \leq f_i(Ind)$ **then**
6: $\quad\quad Ind \leftarrow Ind'$.
7: $\quad$ **end if**
8: **end for**
9: **return** Ind

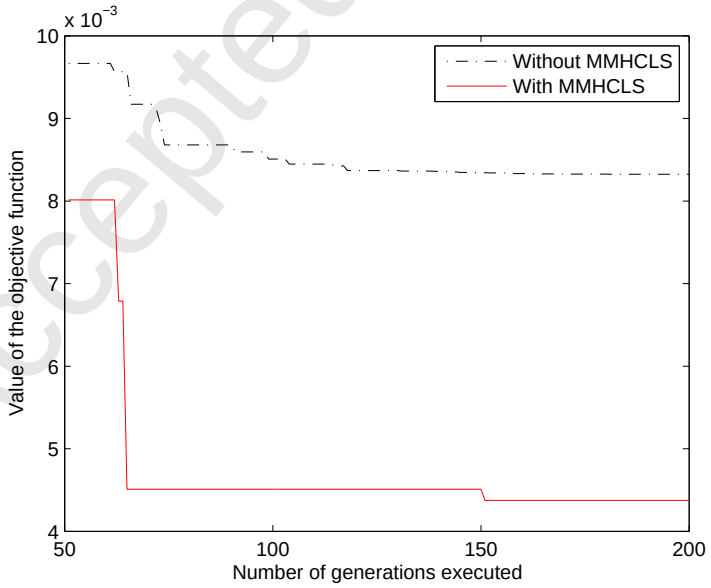

Figure 5 The impact of MMHCLS algorithm on the optimization efficiency

3.2.5. Michaelis-Menten Specific Flip Operation

Chowdhury and Chetty (2011) proposed an S-system specific mutation, namely flip operation, which flips between the regulations in the production and degradation phase. This is based on the observation that the activation (or inhibition) of a gene in the production phase is equivalent to the inhibition (or activation) in the degradation phase. In our model, we propose a flip between activation and inhibition as we observe that activation with a Michaelis-Menten constant is equivalent to inhibition with a different Michaelis-Menten constant which is described below.

If gene-j activates the expression of gene-i, the fraction being multiplied to the production term of the rate equation for expression of gene-i is given below.

$$\frac{y_j}{K_{ij}+y_j} = \frac{y_j^2}{K_{ij}y_j+y_j^2} = \frac{\frac{y_j^2}{K_{ij}}}{y_j+\frac{y_j^2}{K_{ij}}} = \frac{K'_{ij}}{K'_{ij}+y_j} \tag{35}$$

where, $K'_{ij} = \frac{y_j^2}{K_{ij}}$. From the above observation, it is vivid that activation with Michaelis-Menten constant K_{ij} is equivalent to inhibition with Michaelis-Menten constant $\frac{y_j^2}{K_{ij}}$. Based on this observation, we propose the following mutation operation.

When the fitness function does not improve for l iterations, we carry out a flip operation. A specific percentage (in our case 30%) of the best individuals are kept unaltered and a specific percentage (30% in our case) of the worse individuals are selected for flip operation. Each of the regulations in the selected individuals is reversed ($\delta_{ij} = -\delta_{ij}$) and the Michaelis-Menten constant is set as the product of the square of $\bar{y}_j$ and the reciprocal of the current Michaelis-Menten constant, i.e., $K_{ij} = \frac{\bar{y}_j^2}{K_{ij}}$. The flip operation procedure is presented in Algorithm 2. For each of the regulatory arc in the selected individual, a new Michaelis-Menten constant (K') is calculated for a probable arc with an opposite sign (Step 3). If the newly calculated K' lies within twice the feasible range, the sign of the regulatory arc is altered with the new Michaelis-Menten constant and is brought to the nearest bound, if exceeds the limit (Steps 4 to 12).

To evaluate the impact of flip operation (Chowdhury and Chetty, 2011) on MM model, two sets of simulations were run to infer a five-gene in silico network using the TDE algorithm, one with

Algorithm 2 MM Specific Flip Operation

Require: *Ind*, the candidate for flip operation is to be carried out.
Ensure: *Ind*, updated through flip operation.

Find the regulations in Ind, $|\delta_{i1}| = |\delta_{i1}| = \ldots = |\delta_{ir}| = 1$ and $\delta_{i(r+1)} = \delta_{i(r+2)} = \ldots = \delta_{iN} = 0$
for $j = 1$ to r **do**
    $K'_{ij} \leftarrow \frac{\bar{y}_j^2}{K_{ij}(Ind)}$
    if $2 \times K_{min} \leq K'_{ij} \leq 2 \times K_{max}$ **then**
        $\delta_{ij}(Ind) \leftarrow -\delta_{ij}(Ind)$.
        $K_{ij}(Ind) \leftarrow K'_{ij}$.
        if $K_{ij}(Ind) < K_{min}$ **then**
            $K_{ij}(Ind) \leftarrow K_{min}$.
        else if $K_{ij}(Ind) > K_{max}$ **then**
            $K_{ij}(Ind) \leftarrow K_{max}$.
        end if
    end if
end for
return *Ind*

flip operation and the other without flip operation. The value of the best objective function in each generation is plotted in Figure 6 for both the simulations. It may be noted that while the traditional TDE without flip operation gets stuck at a local optimum, the flip operation helps in directing the search to a better solution.

3.3. Computational Complexity Analysis

The nonlinear ODE models of GRN inference typically use population based (and often evolutionary) optimization algorithms for parameter estimation. Solving the differential equations is carried out using numerical integration techniques like Runge-Kutta (RK) methods. For a traditional coupled S-System model, the complexity of inference is $O\left(G_m N_p N^3 N_t\right)$ where, G_m is the maximum number of generations used in the optimization, N_p is the population size and N_t is the number of samples. Considering a decoupled S-System Model, the complexity is $O\left(G_m N_p N^2 N_t\right)$. For the proposed MMC-GRN Model, the complexity is $O\left(G_m N_p N^3 N_t\right)$. Similarly, the complexity of the proposed MMD- GRN Model is $O\left(G_m N_p N^2 N_t\right)$. Although the computational complexity of a decoupled S-system and MM model are of the same order, the real simulation time is expected to be lesser for MM model as the number of parameters is lesser.

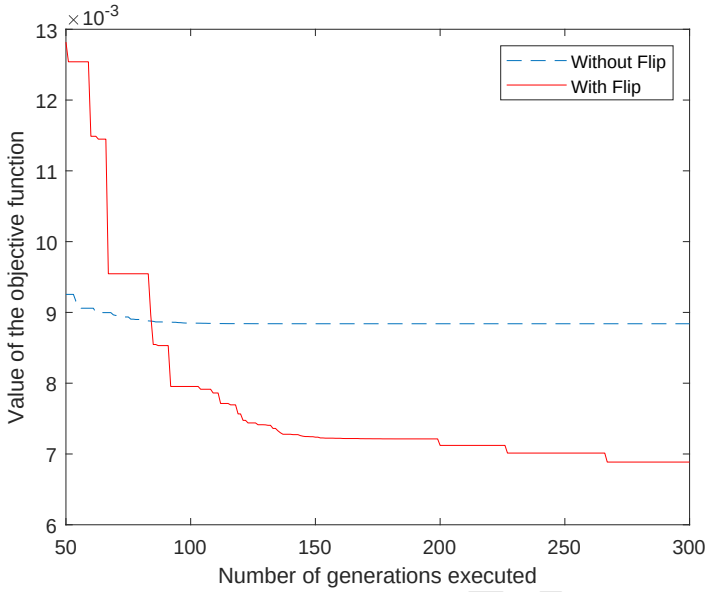

Figure 6 The impact of flip operation on the optimization efficiency

4. Results

The following commonly used performance metrics are used for evaluation of the proposed modeling framework.

1. **Sensitivity:**

$$S_n = \frac{TP}{TP+FN} \tag{36}$$

2. **Specificity:**

$$S_p = \frac{TN}{TN+FP} \tag{37}$$

3. **Precision:**

$$Pr = \frac{TP}{TP+FP} \tag{38}$$

4. **F-score:**

$$F = \frac{1}{\alpha\frac{1}{P_r}+(1-\alpha)\frac{1}{S_n}} = \frac{2P_rS_n}{P_r+S_n} \quad (\text{when } \alpha = 0.5) \tag{39}$$

where, TP is the total number of regulations inferred by the algorithm correctly, FP is the total number of regulations falsely predicted by the algorithm, TN is the total number of non-regulations

correctly inferred by the algorithm and *FN* is the total number of non-regulations falsely predicted by the algorithm.

Sensitivity or recall is the true positive rate and gives the probability of capturing a true regulation by a given method. A high value of sensitivity implies that the method has captured most of the true regulations. Specificity is the true negative rate and gives the probability of capturing a true non-regulation by a given method. A high value of specificity implies that the method has captured most of the true non-regulations. Precision or positive predictive value is the fraction of the captured regulations by a given method that are true regulations. A high value of precision implies that most of the captured regulations are true regulations. F-score gives a weighted balance between sensitivity and precision. A harmonic mean of the two are usually considered in evaluation.

4.1. IRMA- An *In Vivo* Network

Cantone et al. (2009) reported a 5-gene synthetic network in yeast, (i.e., *Saccharomyces cerevisiae*) for “In vivo Reverse engineering and Modeling Assessment (IRMA)", shown in Figure 7a. The five genes are CBF1, GAL4, SWI5, GAL80 and ASH1. There are six transcriptional regulations, namely CBF1 activates GAL4, GAL4 activate SWI5, SWI5 activates CBF1, GAL80 and ASH1, ASH1 inhibits CBF1. The network has two protein-protein interactions, Gal4p and Gal80p inhibiting each other. Cantone et al. (2009) came out with a simplified representation of the network which shows only the transcriptional regulations as shown in Figure 7b. The original network consists of all eight regulatory arcs and a simplified network excludes the two protein-protein interactions.

Saccharomyces cerevisiae can grow on both glucose and galactose; the growth on galactose being more expensive than the growth on glucose. For this reason, the organism always prefers to use glucose for its growth if available (Bhat, 2008). However, if glucose is absent in the growth medium, yeast has to look for alternate sources of nutrition, for example, galactose. When galactose is the only available source for growth, yeast synthesizes enzymes that are necessary for the transport as well as metabolism of galactose, i.e., it switches ‘ON’ the system. Once the organism gets access to glucose, it switches ‘OFF’ the *GAL* genes. Cantone et al. (2009) generated the datasets for both the conditions. (i) Switch ’ON’: by growing the yeast *Saccharomyces cerevisiae* in a galactose-rich and glucose-absent medium and (ii) switch ‘OFF’: by transferring the cultures being grown on galactose into a glucose

(a) Original Network **(b)** Simplified Network

Figure 7 IRMA Network (Cantone et al., 2009). Solid edges indicate gene interactions and the dashed edges indicate protein-protein interactions. Arrow ended arcs indicate activation, block ended arcs indicate inhibition and the circle ended arc indicates either activation or inhibition.

Table 2 The experimental results of MMC-GRN and MMD-GRN for IRMA network, reconstructed from ON dataset. The results for other methods are reported by Chowdhury et al. (2013)

	Original Network				**Simplified Network**			
Method	S_n	S_p	P_r	F	S_n	S_p	P_r	F
MMD-GRN	0.71	0.89	0.71	0.71	1.00	0.82	0.71	0.83
MMC-GRN	0.71	0.89	0.71	0.71	1.00	0.82	0.71	0.83
REGARD (Chowdhury et al., 2012)	0.69	0.83	0.60	0.64	0.70	0.75	0.54	0.61
BITGRN (Morshed et al., 2012)	0.63	1.00	1.00	0.77	0.67	1.00	1.00	0.80
TD-ARACNE (Zoppoli et al., 2010)	0.63	0.88	0.71	0.67	0.67	0.90	0.80	0.73
BANJO (Yu et al., 2004)	0.24	0.76	0.33	0.29	0.50	0.70	0.50	0.50
ALG (Noman and Iba, 2007)	0.77	0.27	0.27	0.40	0.80	0.42	0.36	0.50

medium. Both of these datasets have been used for evaluating the proposed MMC-GRN model.

An evaluation of the proposed method with the existing methods for reconstructing IRMA network from ON dataset is tabulated in Table 2. The Bayesian inference methods used in comparison are BANJO (Bayesian network inference using java objects) (Yu et al., 2004), TD-ARACNE (Algorithm for the reconstruction of accurate cellular networks with time delay) (Zoppoli et al., 2010) and BITGRN (Bayesian information theoretic gene regulatory network reconstruction algorithm) (Morshed et al., 2012). The other methods used in comparison are S-system based methods such as ALG (Noman and Iba, 2007) and REGARD (Chowdhury et al., 2012).

Considering the simplified network, the sensitivity and F-score are the best for MMC-GRN and MMD-GRN among all the methods and other metrics are close to the best values. For the original

Table 3 The experimental results of MMC-GRN and MMD-GRN for IRMA network, reconstructed from OFF dataset. The results for other methods were reported by Chowdhury et al. (2013)

	Original Network				**Simplified Network**			
Method	S_n	S_p	P_r	F	S_n	S_p	P_r	F
MMD-GRN	1.00	0.84	0.67	0.80	1.00	0.82	0.71	0.83
MMC-GRN	1.00	0.83	0.70	0.82	1.00	0.73	0.63	0.77
REGARD (Chowdhury et al., 2012)	0.77	0.76	0.53	0.63	0.80	0.79	0.62	0.70
BITGRN (Morshed et al., 2012)	0.50	0.94	0.80	0.62	0.50	0.90	0.75	0.60
TD-ARACNE (Zoppoli et al., 2010)	0.60	-	0.37	0.46	0.75	-	0.50	0.60
BANJO (Yu et al., 2004)	0.38	0.88	0.60	0.46	0.33	0.90	0.67	0.44
ALG (Noman and Iba, 2007)	0.76	0.56	0.38	0.57	0.80	0.75	0.57	0.67

network, although the performance metrics for MMC-GRN and MMD-GRN are not the best among all the methods available, they are close to the best values. For 800 iterations and 200000 time steps of RK4 per function evaluation, MMC-GRN required 106 hours. The network with similar level of accuracy is achieved by MMD-GRN in 68 hours. In other words, MMD-GRN is 55.88% faster than MMC-GRN.

While MMD-GRN is more biologically relevant than MMC-GRN, the performance metrics for IRMA network for these two models is very similar. It may be noted that among the five genes of IRMA network, three genes, namely ASH1, CBF1 and GAL80 regulate only single genes. The other two genes, GAL4 and SWI5 have two outgoing arcs. Due to the fewer number of genes in the network and also because there are only two genes in the network that regulate multiple genes, we believe these to be the cause of observing similar performance metrics for IRMA network when inferred using MMC-GRN and MMD-GRN models.

Similar evaluations for IRMA OFF dataset is shown in Table 3. Considering both the original and simplified networks, the sensitivity (=1.00) is the best for both MMC-GRN and MMD-GRN. Similarly considering both the original and simplified networks, F-score is the best among all the methods for MMD-GRN and the second best for MMC-GRN. The other metrics for both MMC-GRN and MMD-GRN are close to the best values. It may be noted that the evaluation metric F which represents the trade-off between false positives and false negatives comes out to be the best among all in most of the cases. This is an indication of robustness and effectiveness of the proposed method. In summary, MMD-GRN

(a) Directed Unsigned **(b)** Undirected Unsigned

Figure 8 Unsigned forms of IRMA network used in GRN inference evaluations

shows the best performance in terms of sensitivity and F-score among all the methods compared for IRMA except for original network reconstructed using ON dataset.

Some of the recent GRN models have used both the datasets and showed the inference accuracies. We have made use of the results from both the datasets and the repeated regulations in both the results are considered for inference. For the purpose of comparison, we have considered the network in two forms; i.e., directed and undirected networks as shown in Figure 8. In both these forms, neither the sign of regulation (i.e. activation and inhibition) nor the possibility of self-regulations is considered. We have compared our results using both ON and OFF datasets of IRMA with various recent models shown in Table 4. It may be noted that considering the undirected and unsigned network, both MMC-GRN and MMD-GRN models outperform all other methods in terms of all performance metrics. Considering the directed and unsigned network, both the models show the best specificity and precision among all the approaches. Moreover, the sensitivity and F-score are close to the best values and the second best among all the methods.

4.2. SOS DNA Repair System in *Escherichia coli*

SOS DNA repair mechanism is a well-studied bacterial process, discovered by Miroslav Radman in mid-1970s (Friedberg, 2008). As this process is in response to the signal of DNA-damage, it is termed SOS response after the naval 'Save Our Souls' distress signal (Erill et al., 2007). The system represents the set of genes involved in repairing the DNA when any damage occurs. Although it has been shown that over a thousand genes are involved in SOS DNA damage response system of *Escherichia coli* (*E.*

Table 4 The experimental results of MMC-GRN and MMD-GRN for IRMA network using both ON and OFF datasets

	Undirected Unsigned				**Directed Unsigned**			
Method	S_n	S_p	P_r	F	S_n	S_p	P_r	F
MMD-GRN (best)	0.86	1.00	1.00	0.92	0.71	0.89	0.83	0.77
MMC-GRN	0.86	1.00	1.00	0.92	0.71	0.89	0.83	0.77
ELM-GRNNminer (Rubiolo et al., 2018)	1.00	0.00	0.78	0.88	0.75	0.82	0.67	0.71
BGL_prior (Fan et al., 2017)	-	-	-	-	0.63	0.75	0.63	0.63
MRMSn (Liu et al., 2016)	0.71	0.67	0.83	0.77	-	-	-	-
ProFoGRN (Ceccarelli et al., 2015)	0.86	0.67	0.86	0.86	0.71	0.86	0.83	0.78
MIDER (Villaverde et al., 2014)	0.71	0.67	0.83	0.77	0.38	0.75	0.50	0.43
BPDS (Vera-Licona et al., 2014)	0.71	0.67	0.83	0.77	0.63	0.83	0.71	0.67
OKVAR-Boost (Lim et al., 2013)	0.71	0.67	0.83	0.77	0.63	0.58	0.50	0.56
TDSS (Chowdhury et al., 2013)	0.57	1.00	1.00	0.73	0.50	1.00	1.00	0.67
DELDBN (Li et al., 2011)	0.86	0.33	0.75	0.80	0.88	0.75	0.70	0.78
BANJO (Yu et al., 2004)	-	-	-	-	0.63	0.75	0.63	0.63

coli) (Khil and Camerini-Otero, 2002), the significant genes in the network are considered to be less than 40 (Fernández de Henestrosa et al., 2002).

Among the significant genes of SOS response in *E. coli*, *lexA* and *recA* are considered as the primary genes of the entire network (Little and Mount, 1982). The most explored subnetwork of the system is an eight-gene network consisting of the primary genes, *lexA* and *recA*, and their target genes, *uvrD*, *umuD*, *uvrA*, *uvrY*, *ruvA* and *polB*. Ronen et al. (2002) conducted experiments by damaging DNA using ultraviolet (UV) radiation and reported the expression data of these eight genes. The data consists of four sets of gene expression data, with 50 samples each at intervals of 6 minutes.

A six gene subnetwork, which excludes the genes *uvrY* and *ruvA*, is also vastly studied for modeling and inference problems where the exclusion of those two genes are for not exhibiting adequate changes in the expression levels. Yang et al. (2012) came up with a gold standard of regulations in the six-gene-subnetwork of the SOS response network of *E. coli* from the literature. The gold standard is shown in Table 5.

For the sake of comparison, we have used the data from the experiment with UV intensity 20 J/m^2 (Yang et al., 2012). The best of the experimental results for the proposed MMD-GRN model is compared with other reported results. The evaluation metrics are calculated comparing the network

Table 5 The gold-standard for regulations in six-gene subnetwork of the SOS system in *E. coli* (Yang et al., 2012)

Gene	***uvrD***	***lexA***	***umuD***	***recA***	***uvrA***	***polB***
uvrD	0	-1	-1	+1	+1	0
lexA	0	-1	-1	+1	0	0
umuD	0	-1	-1	+1	0	0
recA	0	-1	-1	+1	0	0
uvrA	+1	-1	-1	+1	0	0
polB	0	-1	-1	+1	0	0

Table 6 The experimental results (best values) for SOS DNA repair system in *E. coli*

	Signed				**Unsigned**			
Method	S_n	S_p	P_r	F	S_n	S_p	P_r	F
MMD-GRN	0.38	0.70	0.50	0.43	0.47	0.82	0.75	0.58
ELM-GRNNminer (Rubiolo et al., 2018)	-	-	-	-	0.37	0.94	0.75	0.45
TDSS (Chowdhury et al., 2013)	0.32	0.88	0.75	0.44	0.35	0.94	0.88	0.50
REGARD (Chowdhury et al., 2012)	0.26	0.88	0.71	0.38	0.30	0.94	0.86	0.44
Kimura et al. (2009)	0.08	0.61	0.10	0.09	0.40	0.88	0.80	0.53
S-Tree (Cho et al., 2006)	0.30	0.94	0.86	0.44	0.30	0.94	0.86	0.44
NGnet (Kimura et al., 2008)	0.26	0.82	0.63	0.37	0.30	0.88	0.75	0.43

inferred with the gold-standard shown in Table 5. The results are compared with other methods in Table 6.

Considering a signed network, our proposed MMD-GRN model has performed better than NGnet, REGARD and Kimura *et al.* in terms of the most important metric, F-score. Considering an unsigned network, MMD-GRN outperforms all other methods mentioned in the table. In many real-life problems, the prediction of a directed network is critical to understand the genes influencing the network.

4.3. *In Silico* Networks

4.3.1. Five and Twenty Gene Networks

A small five-gene *in silico* network having eight regulatory arcs and a 20-gene synthetic network possessing 22 regulatory arcs are considered. Following DREAM4 settings of GeneNetWeaver, (Schaffter et al., 2011) an *in silico* benchmarking software, for each network, ten datasets having 21 time samples each are generated. The simulations are run for inference from both the datasets. All the arcs are

Table 7 Th experimental results (best values) for five gene *in silico* network

	MMC-GRN				MMD-GRN			
Input Data	S_n	S_p	F	Time (s)	S_n	S_p	F	Time (s)
0% Noise	1.0	1.0	1.0	851.2	1.0	1.0	1.0	805.4
5% Noise	1.0	1.0	1.0	851.6	1.0	1.0	1.0	806.8
7.5% Noise	1.0	0.94	0.94	860.4	1.0	0.94	0.94	810.1
10% Noise	1.0	0.82	0.84	862.3	0.86	0.89	0.80	812.2

Table 8 The experimental (best values) results for twenty gene *in silico* network

	MMC-GRN				MMD-GRN			
Input Data	S_n	S_p	P_r	F	S_n	S_p	P_r	F
0% Noise	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5% Noise	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
7.5% Noise	0.95	0.98	0.69	0.80	0.95	0.98	0.69	0.80
10% Noise	0.90	0.97	0.62	0.73	0.90	0.97	0.62	0.73

inferred correctly with direction and sign, i.e., the values of all the evaluation metrics are obtained as exactly 'one', for both the networks using both the models. Different levels of Gaussian noise are then introduced in the data and the network is inferred from noisy data by the proposed method. For both the networks, the results remain the same for 5% noise. However, some false regulations are inferred for the noise levels above 7.5%. However, the deterioration in inference is low. The experimental results are presented in Tables 7 and 8. The simulations required less time for MMD-GRN compared to MMC-GRN.

4.3.2. 50-Gene Subnetwork of Yeast

A subnetwork of size 50 is extracted from the gene network of yeast using the *in silico* benchmarking software, GeneNetWeaver (Schaffter et al., 2011). This is a highly sparse network, shown in Figure 9, having a total of 80 regulatory interactions among the genes.

The datasets are generated using the standard 'DREAM4 settings' providing 10 datasets of 21 samples each at a time interval of 50 units. We reconstructed the network from the generated datasets using the proposed MMD-GRN model. For the sake of comparison, we have used the BANJO (Bayesian Network Inference using Java Objects) software to infer the network from the same datasets. The results are shown in Table 9.

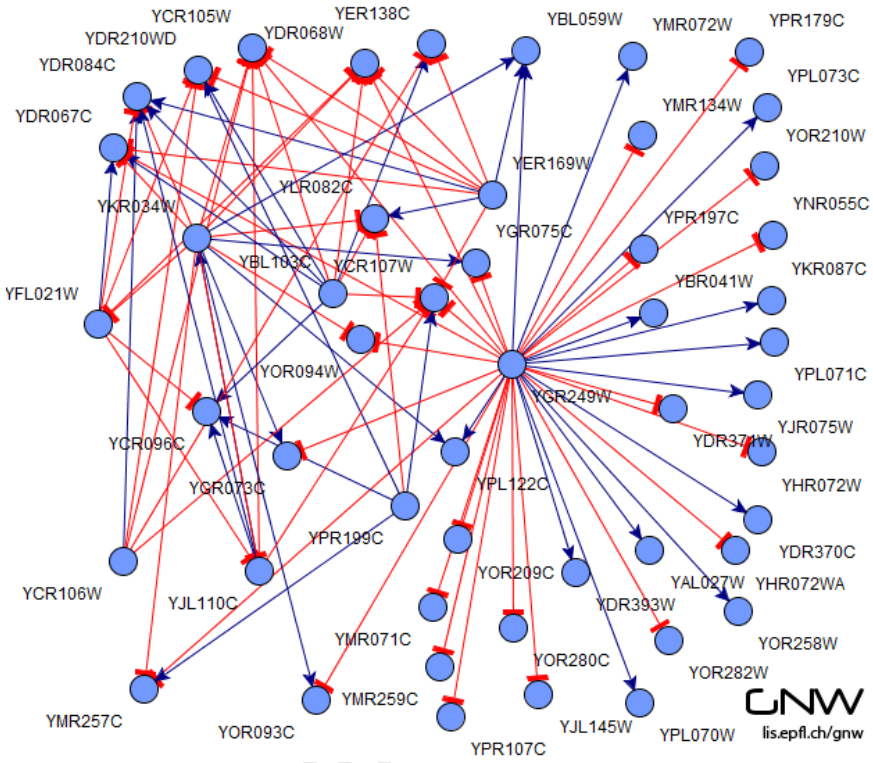

Figure 9 A 50-gene subnetwork of yeast generated by GeneNetWeaver (Schaffter et al., 2011)

Table 9 The experimental results (best values) for 50-gene subnetwork of yeast

Method	S_n	S_p	P_r	F
MMD-GRN	0.44	0.98	0.44	0.44
BANJO (Yu et al., 2004)	0.02	0.98	0.04	0.03

In terms of all performance metrics, BANJO shows very poor performance except for specificity which is the same for both the methods. The proposed MMD-GRN model performs much better than BANJO. The poor performance of BANJO for the network in Figure 9 may be due to its higher extent of sparsity. The scale-free topology of real-life genetic networks (Albert, 2005) illustrates that the GRNs are highly sparse in nature. The results presented for the 50-gene highly sparse network demonstrates the potential of MMD-GRN model for real-life problems of genetic network inference.

5. Conclusions

In this paper, we have proposed a novel model using Michaelis-Menten kinetics for inferring gene regulatory networks. The efficacy of the proposed approach is demonstrated by inferring not only small but also medium scale networks. The proposed coupled model MMC-GRN considers the strength of regulation by a gene to be the same for all the genes it regulates. To overcome this, we further propose the decoupled model, MMD-GRN. The decoupling makes the model more realistic by allowing regulatory strength to be different for different genes. Further, the decoupling reduces the computational complexity and makes the algorithm run faster. The effectiveness of decoupled model, MMD-GRN is is studied using different datasets of various sizes. The results show the superiority of Michaelis-Menten based models over other existing approaches for well known evaluation metrics, i.e., sensitivity, specificity, precision and F-score. The model provides a novel paradigm which can be extended for modeling and analysis of various biochemical systems such as signal transduction processes and various metabolic pathways.

It is being increasingly acknowledged that stochasticity is very important in gene regulatory processes. Recently, Pulkkinen and Metzler (2015) proposed a Variance-corrected Michaelis-Menten equation that predicts transient rates of single-enzyme reactions and response times in bacterial gene-regulation. This facilitates implementation of accurate rates of biochemical reactions even in the presence of large fluctuations. As future work, the Variance Corrected MM equation can be used to further underpin the proposed MM approach. Further, the model can be further studied for reconstruction of protein-protein interaction (PPI) networks. Another future work would be incorporation of cooperativity. Cooperative binding of transcription factors is best modeled by Hill kinetics. The

proposed model can be easily upgraded to include Hill kinetics.

Acknowledgments

This work was supported by Monash University and National ICT Australia (NICTA, currently Data61).

References

P. Xenitidis, I. Seimenis, S. Kakolyris, A. Adamopoulos, Evaluation of artificial time series microarray data for dynamic gene regulatory network inference, Journal of Theoretical Biology 426 (2017) 1 – 16. doi:10.1016/j.jtbi.2017.05.010.

S. A. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets, Journal of theoretical biology 22 (1969) 437–467. doi:10.1016/0022-5193(69)90015-0.

M. Grieb, A. Burkovski, J. E. Sträng, J. M. Kraus, A. Groß, G. Palm, M. Kühl, H. A. Kestler, Predicting variabilities in cardiac gene expression with a Boolean network incorporating uncertainty, PLoS One 10 (2015) e0131832. doi:10.1371/journal.pone.0131832.

N. Friedman, M. Linial, I. Nachman, D. Pe'er, Using Bayesian networks to analyze expression data, Journal of Computational Biology 7 (2000a) 601–620. doi:10.1089/106652700750050961.

N. Friedman, M. Linial, I. Nachman, D. Pe'er, Using Bayesian networks to analyze expression data, in: Proceedings of the Fourth Annual International Conference on Computational Molecular Biology, RECOMB '00, ACM, New York, NY, USA, 2000b, pp. 127–135. doi:10.1145/332306.332355.

B. Yu, J.-M. Xu, S. Li, C. Chen, R.-X. Chen, L. Wang, Y. Zhang, M.-H. Wang, Inference of time-delayed gene regulatory networks based on dynamic Bayesian network hybrid learning method, Oncotarget 8 (2017) 80373–80392. doi:10.18632/oncotarget.21268.

V. A. Huynh-Thu, G. Sanguinetti, Combining tree-based and dynamical systems for the inference of gene regulatory networks, Bioinformatics 31 (2015) 1614–1622. doi:10.1093/bioinformatics/btu863.

V. A. Huynh-Thu, P. Geurts, dynGENIE3: dynamical GENIE3 for the inference of gene networks from time series expression data, Scientific Reports 8 (2018) 3384. doi:10.1038/s41598-018-21715-0.

P. Nguyen, R. Braun, Time-lagged ordered lasso for network inference, BMC Bioinformatics 19 (2018) 545. doi:10.1186/s12859-018-2558-7.

T. Lu, H. Liang, H. Li, H. Wu, High-dimensional ODEs coupled with mixed-effects modeling techniques for dynamic gene regulatory network identification, Journal of the American Statistical Association 106 (2011) 1242–1258. doi:10.1198/jasa.2011.ap10194.

H. Zhu, R. S. P. Rao, T. Zeng, L. Chen, Reconstructing dynamic gene regulatory networks from sample-based transcriptional data, Nucleic Acids Research 40 (2012) 10657–10667. doi:10.1093/nar/gks860.

M. Zheng, J.-n. Wu, Y.-x. Huang, G.-x. Liu, Y. Zhou, C.-g. Zhou, Inferring gene regulatory networks by singular value decomposition and gravitation field algorithm, PLoS ONE 7 (2012) e51141. doi:10.1371/journal.pone.0051141.

O. Hirose, R. Yoshida, S. Imoto, R. Yamaguchi, T. Higuchi, D. S. Charnock-Jones, C. Print, S. Miyano, Statistical inference of transcriptional module-based gene networks from time course gene expression profiles by using state space models, Bioinformatics 24 (2008) 932–942. doi:10.1093/bioinformatics/btm639.

Y. Tamada, R. Yamaguchi, S. Imoto, O. Hirose, R. Yoshida, M. Nagasaki, S. Miyano, SiGN-SSM: open source parallel software for estimating gene networks with state space models, Bioinformatics 27 (2011) 1172–1173. doi:10.1093/bioinformatics/btr078.

M. Rubiolo, D. H. Milone, G. Stegmayer, Extreme learning machines for reverse engineering of gene regulatory networks from expression time series, Bioinformatics 34 (2018) 1253–1260. doi:10.1093/bioinformatics/btx730.

M. M. Kordmahalleh, M. G. Sefidmazgi, S. H. Harrison, A. Homaifar, Identifying time-delayed gene regulatory networks via an evolvable hierarchical recurrent neural network, BioData Mining 10 (2017) 29. doi:10.1186/s13040-017-0146-4.

C.-K. Chen, Reconstructing genetic regulatory networks using two-step algorithms with the differential equation models of neural networks, Interdisciplinary Sciences: Computational Life Sciences (2017). doi:10.1007/s12539-017-0254-3.

A. Khan, G. Saha, R. K. Pal, An approach for reduction of false predictions in reverse engineering of gene regulatory networks, Journal of Theoretical Biology 445 (2018) 9 – 30. doi:10.1016/j.jtbi.2018.02.015.

S. Barbosa, B. Niebel, S. Wolf, K. Mauch, R. Takors, A guide to gene regulatory network inference for obtaining predictive solutions: Underlying assumptions and fundamental biological and data constraints, Biosystems 174 (2018) 37 – 48. doi:10.1016/j.biosystems.2018.10.008.

A. Noor, E. Serpedin, M. Nounou, H. Nounou, Inferring gene regulatory networks via nonlinear state-space models and exploiting sparsity, IEEE/ACM Transactions on Computational Biology and Bioinformatics 9 (2012) 1203–1211. doi:10.1109/TCBB.2012.32.

M. Wahde, J. Hertz, Coarse-grained reverse engineering of genetic regulatory networks, Biosystems 55 (2000) 129–136. doi:10.1016/S0303-2647(99)00090-8.

W. Pan, Y. Yuan, J. GonÃğalves, G. Stan, A sparse bayesian approach to the identification of nonlinear state-space systems, IEEE Transactions on Automatic Control 61 (2016) 182–187. doi:10.1109/TAC.2015.2426291.

Y. Maki, D. Tominaga, M. Okamoto, S. Watanabe, Y. Eguchi, Development of a system for the inference of large scale genetic networks, in: Pacific Symposium on Biocomputing, volume 6, 2001, pp. 446–458. doi:10.1142/9789814447362_0044.

Y. Chen, D. Chen, X. Zou, Inference of biochemical S-systems via mixed-variable multiobjective evolutionary optimization, Computational and Mathematical Methods in Medicine 2017 (2017) 3020326. doi:10.1155/2017/3020326.

G. Yagil, E. Yagil, On the relation between effector concentration and the rate of induced enzyme synthesis, Biophysical Journal 11 (1971) 11 – 27. doi:10.1016/S0006-3495(71)86192-1.

L. Bintu, N. E. Buchler, H. G. Garcia, U. Gerland, T. Hwa, J. Kondev, R. Phillips, Transcriptional regulation by the numbers: models, Current Opinion in Genetics and Development 15 (2005) 116 – 124. doi:10.1016/j.gde.2005.02.007.

S. Kaplan, A. Bren, A. Zaslaver, E. Dekel, U. Alon, Diverse two-dimensional input functions control bacterial sugar genes, Molecular Cell 29 (2008) 786 – 792. doi:10.1016/j.molcel.2008.01.021.

B. Choi, G. A. Rempala, J. K. Kim, Beyond the Michaelis-Menten equation: Accurate and efficient estimation of enzyme kinetic parameters, Scientific Reports 7 (2017) 17018. doi:10.1038/s41598-017-17072-z.

T. Van den Bulcke, K. Van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor, K. Marchal, SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms, BMC Bioinformatics 7 (2006) 43. doi:10.1186/1471-2105-7-43.

N. Noman, H. Iba, Inferring gene regulatory networks using differential evolution with local search heuristics, IEEE/ACM Transactions on Computational Biology and Bioinformatics 4 (2007) 634–647. doi:10.1109/TCBB.2007.1058.

A. Chowdhury, M. Chetty, An improved method to infer gene regulatory network using S-system, in: Evolutionary Computation (CEC), IEEE Congress on, 2011, pp. 1012–1019. doi:10.1109/CEC.2011.5949728.

Y. Maki, T. Ueda, M. Okamoto, N. Uematsu, K. Inamura, K. Uchida, Y. Takahashi, Y. Eguchi, Inference of genetic network using the expression profile time course data of mouse P19 cells, Genome Informatics 13 (2002) 382–383. doi:10.11234/gi1990.13.382.

A. S. K. Youseph, M. Chetty, G. Karmakar, Gene regulatory network inference using Michaelis-Menten kinetics, in: Evolutionary Computation (CEC), IEEE Congress on, 2015a, pp. 2392–2397. doi:10.1109/CEC.2015.7257181.

A. S. K. Youseph, M. Chetty, G. Karmakar, Decoupled modeling of gene regulatory networks using Michaelis-Menten kinetics, in: S. Arik, T. Huang, K. W. Lai, Q. Liu (Eds.), Neural Information Processing, volume 9491 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2015b, pp. 497–505. doi:10.1007/978-3-319-26555-1_56.

V. Henri, Lois Générales de l'action des Diastases, Ph.D. thesis, Université de Paris (Sorbonne), 1903.

L. Michaelis, M. Menten, Kinetik der invertinwirkung, Biochem. Zeitung 49 (1913) 333–369.

A. Cornish-Bowden, One hundred years of Michaelis-Menten kinetics, Perspectives in Science 4 (2015) 3 – 9. doi:10.1016/j.pisc.2014.12.002, proceedings of the Beilstein *ESCEC* Symposium - Celebrating the 100th Anniversary of Michaelis Menten-Kinetics.

M. L. Shuler, F. Kargi, Bioprocess Engineering: Basic Concepts, 2nd ed., Prentice Hall PTR, 2002.

A. Arkin, J. Ross, H. H. McAdams, Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected *Escherichia coli* cells, Genetics 149 (1998) 1633–1648. URL: http://www.genetics.org/content/149/4/1633.

K. C. Chen, L. Calzone, A. Csikasz-Nagy, F. R. Cross, B. Novak, J. J. Tyson, Integrative analysis of cell cycle control in budding yeast, Molecular Biology of the Cell 15 (2004) 3841–3862. doi:10.1091/mbc.e03-11-0794.

A. R. Chowdhury, M. Chetty, N. X. Vinh, Incorporating time-delays in S-system model for reverse engineering genetic networks, BMC Bioinformatics 14 (2013) 196. doi:10.1186/1471-2105-14-196.

I. Otero-Muras, D. Henriques, J. R. Banga, SYNBADm: a tool for optimization-based automated design of synthetic gene circuits, Bioinformatics 32 (2016) 3360–3362. doi:10.1093/bioinformatics/btw415.

R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359.

X.-S. Yang, Nature-Inspired Optimization Algorithms, 1st ed., Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 2014.

H.-Y. Fan, J. Lampinen, A trigonometric mutation operation to differential evolution, Journal of Global Optimization 27 (2003) 105–129. doi:10.1023/A:1024653025686.

N. Noman, H. Iba, On the reconstruction of gene regulatory networks from noisy expression profiles, in: Evolutionary Computation (CEC), IEEE Congress on, 2006, pp. 2543–2550. doi:10.1109/CEC.2006.1688625.

A. R. Chowdhury, Gene Regulatory Network Reconstruction using Time-delayed S-system Model, Ph.D. thesis, Gippsland School of Information Technology, Monash University, Australia, 2014.

I. Cantone, L. Marucci, F. Iorio, M. A. Ricci, V. Belcastro, M. Bansal, S. Santini, M. di Bernardo, D. di Bernardo, M. P. Cosma, A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches, Cell 137 (2009) 172 – 181. doi:10.1016/j.cell.2009.01.055.

P. J. Bhat, Galactose Regulon of Yeast, Springer-Verlag, Berlin, Heidelberg, 2008. doi:10.1007/978-3-540-74015-5.

J. Yu, V. A. Smith, P. P. Wang, A. J. Hartemink, E. D. Jarvis, Advances to Bayesian network inference for generating causal networks from observational biological data, Bioinformatics 20 (2004) 3594–3603. doi:10.1093/bioinformatics/bth448.

P. Zoppoli, S. Morganella, M. Ceccarelli, TimeDelay-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach, BMC Bioinformatics 11 (2010) 154. doi:10.1186/1471-2105-11-154.

N. Morshed, M. Chetty, N. Xuan Vinh, Simultaneous learning of instantaneous and time-delayed genetic interactions using novel information theoretic scoring technique, BMC Systems Biology 6 (2012) 62. doi:10.1186/1752-0509-6-62.

A. Chowdhury, M. Chetty, N. X. Vinh, Adaptive regulatory genes cardinality for reconstructing genetic networks, in: Evolutionary Computation (CEC), IEEE Congress on, 2012, pp. 1–8. doi:10.1109/CEC.2012.6256462.

Y. Fan, X. Wang, Q. Peng, Inference of gene regulatory networks using bayesian nonparametric regression and topology information, Computational and Mathematical Methods in Medicine 2017 (2017) 8 pages. doi:10.1155/2017/8307530.

W. Liu, W. Zhu, B. Liao, X. Chen, Gene regulatory network inferences using a maximum-relevance and maximum-significance strategy, PLOS ONE 11 (2016) 1–19. doi:10.1371/journal.pone.0166115.

M. Ceccarelli, L. Cerulo, G. D. Ruvo, V. Nardone, A. Santone, Infer gene regulatory networks from time series data with probabilistic model checking, in: 2015 IEEE/ACM 3rd FME Workshop on Formal Methods in Software Engineering, 2015, pp. 26–32. doi:10.1109/FormaliSE.2015.12.

A. F. Villaverde, J. Ross, F. MorØčØŇn, J. R. Banga, MIDER: Network inference with mutual information distance and entropy reduction, PLOS ONE 9 (2014) 1–15. doi:10.1371/journal.pone.0096732.

P. Vera-Licona, A. Jarrah, L. D. Garcia-Puente, J. McGee, R. Laubenbacher, An algebra-based method for inferring gene regulatory networks, BMC Systems Biology 8 (2014) 37. doi:10.1186/1752-0509-8-37.

N. Lim, Y. Şenbabaoğlu, G. Michailidis, F. d'Alché Buc, OKVAR-Boost: a novel boosting algorithm to infer nonlinear dynamics and interactions in gene regulatory networks, Bioinformatics 29 (2013) 1416. doi:10.1093/bioinformatics/btt167.

Z. Li, P. Li, A. Krishnan, J. Liu, Large-scale dynamic gene regulatory network inference combining differential equation models with local dynamic bayesian network analysis, Bioinformatics 27 (2011) 2686–2691. doi:10.1093/bioinformatics/btr454.

E. C. Friedberg, A brief history of the DNA repair field, Cell Research 18 (2008) 3–7. doi:10.1038/cr.2007.113.

I. Erill, S. Campoy, J. Barbé, Aeons of distress: an evolutionary perspective on the bacterial SOS response, FEMS Microbiology Reviews 31 (2007) 637–656. doi:10.1111/j.1574-6976.2007.00082.x.

P. P. Khil, R. D. Camerini-Otero, Over 1000 genes are involved in the DNA damage response of *Escherichia coli*, Molecular Microbiology 44 (2002) 89–105. doi:10.1046/j.1365-2958.2002.02878.x.

A. R. Fernández de Henestrosa, T. Ogi, S. Aoyagi, D. Chafin, J. J. Hayes, H. Ohmori, R. Woodgate, Identification of additional genes belonging to the LexA regulon in *Escherichia coli*, Molecular Microbiology 35 (2002) 1560–1572. doi:10.1046/j.1365-2958.2000.01826.x.

J. W. Little, D. W. Mount, The sos regulatory system of *Escherichia coli*, Cell 29 (1982) 11 – 22. doi:10.1016/0092-8674(82)90085-X.

M. Ronen, R. Rosenberg, B. I. Shraiman, U. Alon, Assigning numbers to the arrows: Parameterizing a gene regulation network by using accurate expression kinetics, Proceedings of the National Academy of Sciences 99 (2002) 10555–10560. doi:10.1073/pnas.152046799.

X. Yang, J. E. Dent, C. Nardini, An S-system parameter estimation method (SPEM) for biological networks, Journal of Computational Biology 19 (2012) 175–187. doi:10.1089/cmb.2011.0269.

S. Kimura, S. Nakayama, M. Hatakeyama, Genetic network inference as a series of discrimination tasks, Bioinformatics 25 (2009) 918–925. doi:10.1093/bioinformatics/btp072.

D. Y. Cho, K. H. Cho, B. T. Zhang, Identification of biochemical networks by s-tree based genetic programming, Bioinformatics 22 (2006) 1631–1640. doi:10.1093/bioinformatics/btl122.

S. Kimura, K. Sonoda, S. Yamane, H. Maeda, K. Matsumura, M. Hatakeyama, Function approximation approach to the inference of reduced ngnet models of genetic networks, BMC Bioinformatics 9 (2008) 23. doi:10.1186/1471-2105-9-23.

T. Schaffter, D. Marbach, D. Floreano, GeneNetWeaver: In silico benchmark generation and performance profiling of network inference methods, Bioinformatics 27 (2011) 2263–2270. doi:10.1093/bioinformatics/btr373.

R. Albert, Scale-free networks in cell biology, Journal of Cell Science 118 (2005) 4947–4957. doi:10.1242/jcs.02714.

O. Pulkkinen, R. Metzler, Variance-corrected Michaelis-Menten equation predicts transient rates of single-enzyme reactions and response times in bacterial gene-regulation, Scientific Reports 5 (2015) 17820. doi:10.1038/srep17820.