Gene Regulatory Network Inference using Michaelis-Menten Kinetics

A.S.K. Youseph
Faculty of Information Technology
Monash University, Australia
Email: ahammed.youseph@monash.edu

Madhu Chetty
Faculty of Science and Technology
Federation University Australia
Email: madhu.chetty@federation.edu.au

Gour Karmakar
Faculty of Science and Technology
Federation University Australia
Email: gour.karmakar@federation.edu.au

Abstract*—A gene regulatory network (GRN) represents a collection of genes, connected via regulatory interactions. Reverse engineering GRNs is a challenging problem in systems biology. Various models have been proposed for modeling GRNs. However, many of these models lack the capability to explain the molecular mechanisms underlying the biological process. Michaelis-Menten kinetics can be used to model the biomolecular mechanisms and is a widely used non-linear approach to represent biochemical systems. However, the model in its current form is not suitable for reverse engineering biological systems. In this paper, based on Michaelis-Menten kinetics, we develop a new model to reverse engineer GRNs. The parameter estimation is formulated as an optimization problem which is solved by adapting trigonometric differential evolution (TDE), a variant of differential evolution (DE). The model is applied for reconstructing both *in silico* and *in vivo* networks. The results are promising and as the model is fully biologically relevant, it provides a new perspective for accurate GRN inference.

I. Introduction

A Gene Regulatory Network (GRN) is a set of genes interacting among themselves. The regulation of gene expression is predominantly via proteins, the gene products. The regulations can be either repression or activation, i.e., a product from a gene may repress or activate the expression of an another gene. The metabolic activities in a cell are catalyzed by enzymes. The enzymes required in metabolism are produced in a cell. The production of these enzymes is mainly driven by the regulatory mechanisms in the corresponding gene regulatory systems.

A wide range of models have been proposed in the existing literature for the reconstruction of gene regulatory networks. The systems biologists always have to face a trade-off between the modeling simplicity and biological relevance. There are several linear models like boolean networks [1], Bayesian networks [2], [3], etc. used in reconstruction of GRNs, the biological relevance of which is debatable. Among the very few non-linear models used in literature for reverse engineering GRNs, S-system models have been considered to be better fitting the biological nature of genetic networks [4]. Although the S-system models can represent the system more accurately than linear models, as the S-system equations use power-law functions, they fail to represent the sigmoidal behavior of biochemical processes precisely. Experimental studies have been carried out in the existing literature, to understand the variation in gene expression levels as a function of the gene's inducers. Such studies usually show the variation as sigmoidal functions [5]–[7]. The number of parameters to be estimated in an S-system model is $2N^2 + 2N$ for a network of N genes which makes the parameter estimation process computationally expensive [8], [9]. Incorporation of prior knowledge, an approach to enhance the speed of learning process, is not efficient in these models. For example, if a regulation is known, one parameter becomes known in our model whereas in S-system it just gives the sign of one parameter and the parameter is to be still estimated.

Modeling using Michaelis-Menten kinetics, a widely used nonlinear approach in modeling biochemical systems, is proven to be good for many biological processes. In spite of this, to the best of our knowledge, the model has not been applied for GRN inference. In this paper, we propose a novel model based on Michaelis-Menten kinetics suitable for reconstructing gene regulatory networks. Though the model is fully biologically relevant and hence highly nonlinear, the number of parameters to be estimated is less than that in S-system model. Moreover, the model enables us to incorporate prior knowledge in a very efficient way, as the known regulations reduce the number of unknown parameters to be estimated.

The problem of identification of parameters is formulated as an optimization problem. Evolutionary algorithms (EA) have been considered to be good in global optimization. Among the various evolutionary algorithms, differential evolution (DE) is considered to be very good in optimization of nonlinear and continuous systems [10]. We have used a variation of DE, trigonometric differential evolution (TDE) [11] for estimation of parameters. The proposed model is used to reconstruct *in silico* networks of small and medium size, from both noise free and noisy data. IRMA, an *in vivo* network of yeast, *S. cerevisiae* is also inferred using the model.

Rest of the paper is organized as follows: Section II details the background of the proposed approach. In Section III, we describe the proposed model and identification method in detail. In Section IV, we show and discuss the experimental results. In section V, we present the conclusions and future scope of the work.

II. Background

Biological systems are non-linear in nature. The gene regulations are the results of protein-protein interactions and protein-DNA interactions. These interactions are best modeled by non-linear equations. Michaelis-Menten kinetics are well accepted in modeling biochemical reactions. These are used extensively in engineering biological systems. Bulcke *et al.*

978-1-4799-7492-4/15/$31.00 ©2015 IEEE

developed a software which generates synthetic gene expression data where the interaction kinetics are modeled using Michaelis-Menten and Hill kinetics [12]. Michaelis-Menten equations can be represented as a subset of Hill equations, as we will see later in this section.

If the formation of a product, say P, is activated by an activator A, the rate of formation of the product, v can be expressed by Michaelis-Menten kinetics as

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

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

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

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 we have both inhibition and activation at the same time,

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

By Hill kinetics, the rate, v can be defined for the above case as

$$v = \frac{d[P]}{dt} = v_{max}\frac{K_I^n}{K_I^n + [I]^n}\frac{[A]^n}{K_A^n + [A]^n}f_3' \tag{4}$$

where, n is the Hill coefficient and it accounts for the steepness of sigmoidal curve, which represents a biological behavior termed ‘cooperativity’. A value of $n > 1$ indicates ‘positive cooperativity’ which means that binding of one molecule increases the binding affinity of other molecules. A value of $n < 1$ indicates ‘negative cooperativity’ which means that binding of one molecule decreases the binding affinity of other molecules. A value of $n = 1$ indicates ‘non-cooperativity’ which means that binding of one molecule does not influence the binding affinity of other molecules. Michaelis-Menten kinetics is a subset of Hill kinetics where Hill coefficient, $n = 1$.

Although ‘cooperativity’ is known in regulatory networks, it is not a prevalent phenomenon. This behavior is insignificant in most of the systems except in those which possess a ’switch-response’ to ‘shut down’ or ‘turn on’ the expression quickly. For simplicity of the modeling, we have not included the Hill coefficient in the model. In cases where ‘cooperativity’ is significant, one can easily extend the proposed model to include Hill kinetics. Opper and Sanguinetti have derived a different model for learning transcriptional dynamics from classical Michaelis-Menten model of transcription control [13], but it doesnt explicitly capture the underlying processes. In their model, the fractional term in the kinetic equation is replaced by a binary variable which completely ignores the sigmoid behavior. Moreover, a linear summation is used for modeling multiple interactions which is contradicting the non-linear behavior of the biological systems.

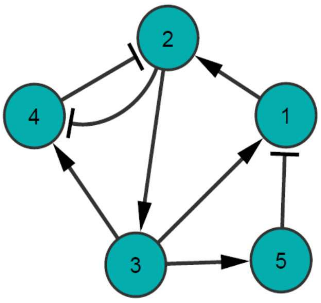


Fig. 1: A simple GRN of five genes. Arrow ended arcs indicate activation and block ended arcs indicate inhibition.

We will be able to model biochemical systems in the above described manner when the regulations are known a priori. However, in a reverse engineering problem, the regulations are unknown and applying this kinetics in reverse engineering biochemical systems is not straightforward, especially in gene regulatory network inference.

III. The Proposed Method

In this section, we propose a new model for reverse engineering GRNs based on Michaelis-Menten kinetics.

A. *The Model*

In reverse engineering GRNs, the task is to infer the regulatory interactions in the network from the gene expression data. Since Michaelis-Menten equation can be written only when we know the regulation, we introduce binary variables to account for the presence and absence of regulations. We introduce two binary variables as the regulation can be either activation or repression. For a system of N genes, the model equation can be defined as:

$$\begin{aligned}\dot{x}_i = \frac{dx}{dt} &= \alpha_i \prod_{j=1}^{N}\left(g_{ij}\frac{x_j}{K_j + x_j} + h_{ij}\frac{K_j}{K_j + x_j}\right) - \beta_i x_i \\ &= \alpha_i \prod_{j=1}^{N}\left(\frac{g_{ij}x_j + h_{ij}K_j}{K_j + x_j}\right) - \beta_i x_i\end{aligned} \tag{5}$$

where, α_i is the maximum rate of expression of gene i
$h_{ij}, g_{ij} \in \{0, 1\}, 1 \leq h_{ij} + g_{ij} \leq 2 \ \forall \ i, j = 1, \ldots, N$
$h_{ij} = 1, g_{ij} = 1$ 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.
β_i is the self decay rate of mRNA expressed by gene i.

Let us consider a simple example of a 5-gene network shown in Figure 1 to illustrate the model. Considering the node 4 where, gene-4 expression 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 general equation as

$$\begin{aligned}\frac{dx_4}{dt} =&\alpha_4 \prod_{j=1}^{5} \left[\frac{g_{ij}x_j + h_{ij}K_j}{K_j + x_j}\right] - \beta_4 x_4\\ =&\alpha_4 \left(\frac{g_{41}x_1 + h_{41}K_1}{K_1 + x_1}\right)\left(\frac{g_{42}x_2 + h_{42}K_2}{K_2 + x_2}\right)\\ &\times \left(\frac{g_{43}x_3 + h_{43}K_3}{K_3 + x_3}\right)\left(\frac{g_{44}x_4 + h_{44}K_4}{K_4 + x_4}\right)\\ &\times \left(\frac{g_{45}x_5 + h_{45}K_5}{K_5 + x_5}\right) - \beta_4 x_4\end{aligned}$$

As there is 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 for gene-3 activating gene-4, $g_{43} = 1$ 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).

$$\frac{dx_4}{dt} = \alpha_4 \left(\frac{K_2}{K_2 + x_2}\right)\left(\frac{x_3}{K_3 + x_3}\right) - \beta_4 x_4$$

h_{ij} and g_{ij} both cannot be zero for any i and j. In the estimation algorithm, instead of h_{ij} and g_{ij}, we can estimate a new variable $\delta_{ij} \in \{-1, 0, 1\}$ to reduce the computational cost as below.

$$\delta_{ij} = \begin{cases} 0, \text{if } g_{ij} = 1, h_{ij} = 1\\ +1, \text{if } g_{ij} = 1, h_{ij} = 0\\ -1, \text{if } g_{ij} = 0, h_{ij} = 1\end{cases} \tag{6}$$

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

$$\delta_{ij} = g_{ij} - h_{ij} \tag{7}$$

$$g_{ij} = \frac{\delta_{ij} + 1}{\mid \delta_{ij} \mid + 1} \tag{8}$$

$$h_{ij} = \frac{1 - \delta_{ij}}{\mid \delta_{ij} \mid + 1} \tag{9}$$

Substituting for g_{ij} and h_{ij} in equation (5), we get

$$\frac{dx}{dt} = \alpha_i \prod_{j=1}^{N} \left(\frac{\frac{\delta_{ij}+1}{|\delta_{ij}|+1}x_j + \frac{1-\delta_{ij}}{|\delta_{ij}|+1}K_j}{K_j + x_j}\right) - \beta_i x_i \tag{10}$$

For a system of N genes, the parameters are $\alpha \in \mathbb{R}^{N\times 1}$, $\beta \in \mathbb{R}^{N\times 1}$, $K \in \mathbb{R}^{N\times 1}$ and $\delta \in \mathbb{Z}^{N\times N}$. In effect, for N number of genes, equation (10) requires to estimate the values of N instances of α, K and β 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 our proposed model is $(N + N + N + N^2) = (3N + N^2)$ where, N^2 parameters are integer variables.

S-system model, a widely accepted non-linear ODE model used in reverse engineering GRNs, has the following equation.

$$\dot{x}_i = \frac{dx}{dt} = \alpha_i \prod_{j=1}^{N} x_j^{g_{ij}} - \beta_i \prod_{j=1}^{N} x_j^{h_{ij}} \tag{11}$$

The number of parameters to be estimated is $2N + 2N^2$. Although the order of complexity of the proposed model is same ($\sim N^2$) as that of S-system, the total number of parameters to be estimated is much less, the difference being $N^2 - N$. The increase in number of parameters to be estimated with the increase of number of genes for the proposed model is compared with that of S-system in Figure 2.

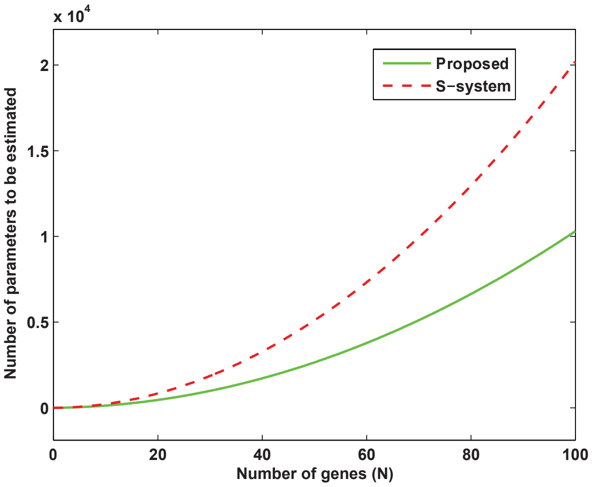


Fig. 2: Comparison of the number of parameters to be estimated in the proposed model with that of the S-system model

B. Parameter Estimation

Once the model is ready, the task is to identify the parameters representing the system. The parameter estimation is usually carried out using an optimization algorithm by which the modeling error, the deviation of the model prediction from the real data, is minimized. The data we have is the microarray data, $x \in \mathbb{R}^{N\times T}$ where, N is the number of genes and T is the number of time samples.

Mean squared error (MSE) and relative squared error are the widely considered objective functions to be minimized in most of these such problems. A penalty term on complexity, to maintain the trade off between structure complexity and prediction error, is usually added to the objective function in GRN inference algorithms.

$$MSE = \frac{1}{N\times T}\sum_{i=1}^{N}\sum_{j=1}^{T}\left\{x_{ij}^{cal} - x_{ij}^{exp}\right\}^2 \tag{12}$$

$$\text{Relative squared error} = \frac{1}{N\times T}\sum_{i=1}^{N}\sum_{j=1}^{T}\left\{\frac{x_{ij}^{cal} - x_{ij}^{exp}}{x_{ij}^{exp}}\right\}^2 \tag{13}$$

where, x_{ij}^{cal} and x_{ij}^{exp} are the gene expression levels of gene i at j^{th} time instant as inferred by the model and as obtained in the experiment, respectively.

One of the constraints being used in parameter estimation of GRN models is with respect to 'indegree' [14] or cardinality, which is the total number of genes involved in regulation (directly) of a given gene. A maximum indegree is usually set as a constraint in the optimization. To account for this structural complexity constraint, a penalty term can be added in the objective function [14]. We considered the fitness function as the sum of absolute square error and a penalty term which is analogous to the penalty term in [9]. The objective function minimized is the following.

$$f=\sum_{i=1}^{N}\sum_{j=1}^{T}\left\{\frac{x_{ij}^{cal}-x_{ij}^{exp}}{x_{ij}^{exp}}\right\}^{2}+\sum_{i=1}^{N}C_i\frac{N}{Z_i} \qquad (14)$$

where, N is the total number of genes and Z is the total number of non-regulations.

$$Z_i=N-\text{number of regulations}=N-\sum_{j=1}^{N}\mid\delta_{ij}\mid \qquad (15)$$

C_i is a constant as defined in [9].

$$C_i=\begin{cases}1,\ \text{if } J_i\le r_i\le I_i \text{ or } r_i=0\\ \frac{J_i}{r_i}2^{(J_i-r_i)},\ \text{if } r_i<J_i\\ \frac{r_i}{I_i}2^{(r_i-I_i)},\ \text{if } r_i>I_i \quad \text{where, } r_i=\sum_{j=1}^{N}\mid\delta_{ij}\mid\end{cases} \qquad (16)$$

where, J_i and I_i respectively, are the minimum and maximum cardinalities of gene i, r_i is the total number of regulations for gene i. Now, this problem gets formulated as a mixed integer nonlinear programming (MINLP) problem.

$$\begin{aligned}&\min_{\alpha,\beta,K,\delta}\ f(x,\delta)\\ s.t.\quad &\frac{dx}{dt}=f'(x,\alpha,\beta,K,\delta),\quad x(t_o)=x_0\\ &0\le J_i\le\sum_{j=1}^{N}\mid\delta_{ij}\mid\le I_i\le N\\ &\delta\in\mathbb{R}^{N\times N}:\delta_{ij}\in\{-1,0,1\}\\ &\alpha,\beta,K,x(t)\in\mathbb{R}^{N},\quad\alpha,\beta,K,x(t)\ge\overrightarrow{0}\end{aligned} \qquad (17)$$

where, f is the function defined in equation (14), f' is the R.H.S. of equation (10) and J_i and I_i respectively, are the minimum and maximum cardinalities of gene i and are made adaptive during the optimization as in [9]. If no constraint is set on the cardinality, the default values of J_i and I_i will be 0 and N, respectively.

The optimization algorithm, differential evolution, consists of mainly three steps; initialization, mutation-crossover and selection. The first step, initialization can be done as follows.

$$w_i^g=w_{min}+\rho_i(w_{max}-w_{min}),\ \rho_i\in[0,1]\qquad i=1,2,...,N_p \qquad (18)$$

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. The second step in DE is the mutation operation. The normal mutation operation is the following [10].

$$u_i^{g+1}=w_{r_1}^g+F_m(w_{r_2}^g-w_{r_3}^g),\ r_1,r_2,r_3\in\{1,2,\ldots,N_p\}\qquad i\ne r_1\ne r_2\ne r_3\ i=1,2,...,N_p \qquad (19)$$

where, F_m is the mutation factor and r_1, r_2 and r_3 are any three different random numbers from $\{1,2,\ldots,N_p\}$.
A random number, $rand1_j\in[0,1]$ is generated and if $rand1_j>M_t$ (where M_t is the trigonometric mutation ratio) the normal mutation is carried out and otherwise the trigonometric mutation is carried out. The trigonometric mutation operation is done as follows [11].

$$\begin{aligned}u_i^{g+1}=&\frac{w_{r_1}^g+w_{r_2}^g+w_{r_3}^g}{3}+(p_2-p_1)(w_{r_1}^g-w_{r_2}^g)\\&+(p_3-p_2)(w_{r_2}^g-w_{r_3}^g)+(p_1-p_3)(w_{r_3}^g-w_{r_1}^g)\end{aligned} \qquad (20)$$

where,

$$p_1=\frac{\mid f(w_{r_1}^g)\mid}{p'},p_2=\frac{\mid f(w_{r_2}^g)\mid}{p'},p_3=\frac{\mid f(w_{r_3}^g)\mid}{p'}$$
$$p'=\mid f(w_{r_1}^g)\mid+\mid f(w_{r_2}^g)\mid+\mid f(w_{r_3}^g)\mid \qquad (21)$$

The next step is a crossover operation. Here, the new members v_i^{g+1} of the next generation are created by randomly selecting elements $u_i^{g+1}(j)$ and $w_i^g(j)$ from the perturbed vector, u_i^{g+1} and the current generation, w_i^g, respectively using the following rule [10]:

$$v_i^{g+1}(j)=\begin{cases}u_i^{g+1}(j),\text{if } rand2_j\le CR,\ CR\in[0,1]\\ w_i^g(j),\text{otherwise}\end{cases} \qquad (22)$$

where, $rand2_j$ is a random number $\in[0,1]$ and CR is a user-defined crossover factor. The higher the value of CR, the more diverse will be the search. The next step is the evaluation and selection process. For the two generations, namely parent and children, if the constraints are satisfied, the objective functions are evaluated. From the parent and children generations, for each member, w_i^g and v_i^{g+1}, the new parent generation is selected as follows:

$$w_i^{g+1}=\begin{cases}v_i^{g+1},\text{if } f(v_i^{g+1}) \text{ is better than } f(w_i^g)\\ w_i^g,\text{otherwise}\end{cases} \qquad (23)$$

IV. Results and Discussion

We coded the program in Matlab and run using a PC with 3.1 GHz Intel(R) Core(TM) i5-2400 CPU and 4 GB of RAM. For the optimization algorithm, TDE, we set the parameters as in [9]; the mutation factor, $F=0.5$, the cross over ratio, $CR=0.8$, the trigonometric mutation ratio, $M_t=0.05$, the population size and $N_p=5\times$the number of parameters to be estimated. We executed the simulation for 800 generations.

The following are the metrics used to evaluate the inferred network.

1) **Sensitivity:**
$$S_n=\frac{TP}{TP+FN} \qquad (24)$$

2) **Specificity:**
$$S_p=\frac{TN}{TN+FP} \qquad (25)$$

3) **Precision:**
$$Pr=\frac{TP}{TP+FP} \qquad (26)$$

4) **F-score:**
$$\begin{aligned}F=&\frac{1}{\alpha\frac{1}{Pr}+(1-\alpha)\frac{1}{S_n}}\\=&\frac{2PrS_n}{Pr+S_n}\quad(\text{when } \alpha=0.5)\end{aligned} \qquad (27)$$

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 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 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.

A. *In silico Network*

We generated ten datasets each for a 5-gene and 20-gene synthetic networks having 8 and 22 arcs, respectively. We ran the algorithm for inference from both the datasets. We could infer all the arcs, correctly with direction and sign, i.e. we got the values of all the evaluation metrics as exactly 'one', for both the networks. We, then introduced different levels of noise in the data and inferred the network. For both the networks, the results remain the same for 5% noise. However, some false inferences were made for noise levels above 7.5%. As the noise is not modeled in the method, this inference is obvious and acceptable. Using noise modeling in our method, these false references can be reduced. The experimental results are presented in Table I.

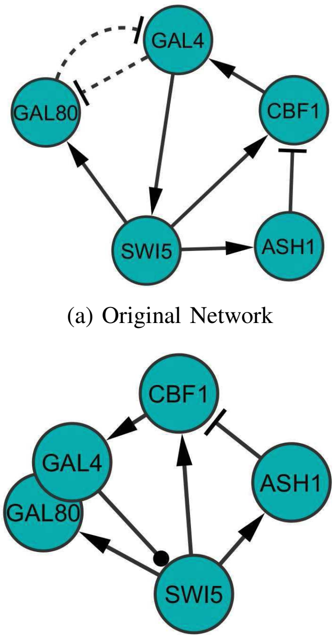
TABLE I: The best of the experimental results for *in silico* network

	5-Gene Network				**20-Gene Network**			
Input Data	$\mathbf{S}_n$	$\mathbf{S}_p$	$\mathbf{P}_r$	**F**	$\mathbf{S}_n$	$\mathbf{S}_p$	$\mathbf{P}_r$	**F**
0% Noise	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
5% Noise	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
7.5% Noise	1.0	0.94	0.89	0.94	0.95	0.98	0.69	0.80
10% Noise	1.0	0.82	0.73	0.84	0.90	0.97	0.62	0.73

B. *In vivo Network - IRMA*

We used the gene expression data of IRMA Network [15] for reconstruction. This is a 5-gene synthetic yeast network, shown in Figure 3a, for which the *in vivo* data is available. The interactions between *GAL4* and *GAL80* are protein-protein interactions and not the transcriptional regulations. Irene *et al.* [15] came out with a simplified representation of the network which shows only the transcriptional regulations as shown in Figure 3b.

A comparison of the experimental results, obtained using IRMA ON data, for our model with other models is tabulated in Table II. The methods based on Bayesian network models used in comparison include BANJO, TD-ARACNE and BIT-GRN. The other methods used in comparison are S-system based methods such as ALG and REGARD. The evaluation metrics, Sensitivity (S_n), Specificity (S_p), Precision (Pr) and F-score (F) are compared. Considering the simplified network, the sensitivity (which is =1.00) and F-score are the best among all the methods and other metrics are close to the best values. For the original network, although the performance metrics of our model are not the best among all the methods available, they are close to the best values.



(a) Original Network

(b) Simplified Network

Fig. 3: IRMA Network [15]. Solid edges indicate gene interactions and the dashed edges indicates protein-protein interaction. Arrow ended arcs indicate activation, block ended arcs indicate inhibition and the circle ended arc indicates either activation or inhibition.

Using IRMA OFF data as the input to the algorithm, the experimental results for our model are compared with that of other models in Table III. For both the original and simplified networks, the sensitivity (=1.00) and F-score are the best among all the methods and other metrics 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.

V. Conclusions

A broad variety of models are proposed in the past for GRN inference from microarray data. However, most of them are biologically irrelevant and none of them are capable of explaining the molecular mechanisms underlying the biochemical system. In this paper, we have proposed a novel method of reverse engineering, based on a fully biologically relevant formalism, namely Michaelis-Menten. We have tested the proposed method on both *in silico* and *in vivo* networks. The experimental results are comparable with that of the other methods and models available in literature. Our next goal is

TABLE II: The experimental results for IRMA network, reconstructed from ON dataset. The results for other methods were reported in [16].

	Original Network				Simplified Network			
Method	$\mathbf{S}_n$	$\mathbf{S}_p$	$\mathbf{P}_r$	$\mathbf{F}$	$\mathbf{S}_n$	$\mathbf{S}_p$	$\mathbf{P}_r$	$\mathbf{F}$
Proposed (best)	0.71	0.89	0.71	0.71	1.00	0.82	0.71	0.83
Proposed avg ± std	0.67± 0.06	0.87± 0.06	0.67± 0.08	0.67± 0.06	0.88± 0.10	0.85± 0.05	0.73± 0.05	0.80± 0.04
REGARD [9]	0.69	0.83	0.60	0.64	0.70	0.75	0.54	0.61
BITGRN [17]	0.63	1.00	1.00	0.77	0.67	1.00	1.00	0.80
TD-ARACNE [18]	0.63	0.88	0.71	0.67	0.67	0.90	0.80	0.73
BANJO [19]	0.24	0.76	0.33	0.29	0.50	0.70	0.50	0.50
ALG [20]	0.77	0.27	0.27	0.40	0.80	0.42	0.36	0.50

TABLE III: The experimental results for IRMA network, reconstructed from OFF dataset. The results for other methods were reported in [16].

	Original Network				Simplified Network			
Method	$\mathbf{S}_n$	$\mathbf{S}_p$	$\mathbf{P}_r$	$\mathbf{F}$	$\mathbf{S}_n$	$\mathbf{S}_p$	$\mathbf{P}_r$	$\mathbf{F}$
Proposed (best)	1.00	0.83	0.70	0.82	1.00	0.73	0.63	0.77
Proposed avg ± std	0.92± 0.07	0.80± 0.04	0.67± 0.05	0.91± 0.04	0.69± 0.08	0.75± 0.05	0.64± 0.05	0.76± 0.04
REGARD [9]	0.77	0.76	0.53	0.63	0.80	0.79	0.62	0.70
BITGRN [17]	0.50	0.94	0.80	0.62	0.50	0.90	0.75	0.60
TD-ARACNE [18]	0.60	-	0.37	0.46	0.75	-	0.50	0.60
BANJO [19]	0.38	0.88	0.60	0.46	0.33	0.90	0.67	0.44
ALG [20]	0.76	0.56	0.38	0.57	0.80	0.75	0.57	0.67

to implement the model and algorithm to infer medium and large scale GRNs.

Acknowledgment

In addition to Monash University scholarships, this work is supported by the ICT Centre of Excellence program, National ICT Australia (NICTA). The authors wish to acknowledge useful comments from Terry Caelli, University of Melbourne.

References

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

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

[3] R. Ram and M. Chetty, "A markov-blanket-based model for gene regulatory network inference," *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, vol. 8, no. 2, pp. 353–367, March 2011.

[4] Y. Maki, D. Tominaga, M. Okamoto, S. Watanabe, and Y. Eguchi, "Development of a system for the inference of large scale genetic networks," in *Pacific Symposium on Biocomputing*, vol. 6, 2001, pp. 446–458.

[5] G. Yagil and E. Yagil, "On the relation between effector concentration and the rate of induced enzyme synthesis," *Biophysical Journal*, vol. 11, no. 1, pp. 11 – 27, 1971.

[6] L. Bintu, N. E. Buchler, H. G. Garcia, U. Gerland, T. Hwa, J. Kondev, and R. Phillips, "Transcriptional regulation by the numbers: models," *Current Opinion in Genetics and Development*, vol. 15, no. 2, pp. 116 – 124, 2005.

[7] S. Kaplan, A. Bren, A. Zaslaver, E. Dekel, and U. Alon, "Diverse two-dimensional input functions control bacterial sugar genes," *Molecular Cell*, vol. 29, no. 6, pp. 786 – 792, 2008.

[8] A. Chowdhury and M. Chetty, "An improved method to infer gene regulatory network using s-system," in *Evolutionary Computation (CEC), IEEE Congress on*, 2011, pp. 1012–1019.

[9] A. Chowdhury, M. Chetty, and N. X. Vinh, "Adaptive regulatory genes cardinality for reconstructing genetic networks," in *Evolutionary Computation (CEC), IEEE Congress on*, 2012, pp. 1–8.

[10] R. Storn and K. Price, "Differential evolution a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[11] H.-Y. Fan and J. Lampinen, "A trigonometric mutation operation to differential evolution," *Journal of Global Optimization*, vol. 27, no. 1, pp. 105–129, 2003.

[12] T. Van den Bulcke, K. Van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor, and K. Marchal, "Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms," *BMC Bioinformatics*, vol. 7, no. 1, p. 43, 2006.

[13] M. Opper and G. Sanguinetti, "Learning combinatorial transcriptional dynamics from gene expression data," *Bioinformatics*, vol. 26, no. 13, pp. 1623–1629, 2010.

[14] 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, 2005.

[15] I. Cantone, L. Marucci, F. Iorio, M. A. Ricci, V. Belcastro, M. Bansal, S. Santini, M. di Bernardo, D. di Bernardo, and M. P. Cosma, "A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches," *Cell*, vol. 137, no. 1, pp. 172 – 181, 2009.

[16] A. Chowdhury, M. Chetty, and N. Vinh, "Incorporating time-delays in s-system model for reverse engineering genetic networks," *BMC Bioinformatics*, vol. 14, no. 1, p. 196, 2013.

[17] N. Morshed, M. Chetty, and N. Xuan Vinh, "Simultaneous learning of instantaneous and time-delayed genetic interactions using novel information theoretic scoring technique," *BMC Systems Biology*, vol. 6, no. 1, p. 62, 2012.

[18] P. Zoppoli, S. Morganella, and M. Ceccarelli, "Timedelay-aracne: Reverse engineering of gene networks from time-course data by an information theoretic approach," *BMC Bioinformatics*, vol. 11, no. 1, p. 154, 2010.

[19] J. Yu, V. A. Smith, P. P. Wang, A. J. Hartemink, and E. D. Jarvis, "Advances to bayesian network inference for generating causal networks from observational biological data," *Bioinformatics*, vol. 20, no. 18, pp. 3594–3603, 2004.

[20] N. Noman and H. Iba, "Inferring gene regulatory networks using differential evolution with local search heuristics," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 4, pp. 634–647, 2007.