A Simulation Framework for Modeling Combinatorial Control in Transcription Regulatory Networks

Sushmita Roy[†], Terran Lane[†], Margaret Werner-Washburne[*]
[†] Department of Computer Science, University of New Mexico
[†] Department of Biology, University of New Mexico

Abstract

With the increasing availability of genome scale data, a plethora of algorithms are being developed to infer genome scale regulatory networks. However, identifying the appropriate algorithm for a particular inference task is a critical and difficult modeling question. This is because for most real-world data, the true network is rarely known and different algorithms produce different networks, which may be due to assumptions inherrent to the algorithm or the objective the algorithm is designed to optimize. One approach to address this issue is to build artificial, but biologically realistic, gene networks for which the true topology is known. These networks can be sampled to generate data, which can then serve as a testbed to compare a suite of network inference algorithms.

Existing simulation systems either require highly detailed descriptions of network components making the generation of large-scale networks (> 100 nodes) infeasible, or, make simplifications that render the simulated networks very far from true models of gene regulation. We have developed a simulation system that builds a network of genes and proteins and models the rate of transcriptional activity as a non-linear function of a set of transciption factor complexes. Our simulator uses a system of differential equations and interfaces with Copasi, a differential equation solver, to produce steady-state and time-series expression datasets. Overall, our simulator has the following advantages: (i) it can produce expression datasets in an efficient and high-throughput manner, (ii) it is biologically more realistic since it models transcription rate as a function of transcription factor levels rather than gene expression levels, (iii) it incorporates the combinatorial control among different transcription factors, and (iii) it incorporates gene and protein expressions, thus allowing the comparison of algorithms that use different combinations of these data sources.

1 Introduction

Gene regulatory networks are networks of all the genes and transcription factors of an organism, which describe the combinatorial control of up or down-regulating genes in a context-specific manner. Because high-throughput assays measure only a subset of the network properties at a time we need network inference algorithms that can integrate different high-throughput assays and build a complete and high-confidence estimate of the true regulatory network. This raises an important issue of identifying the best algorithm from a repertoire of existing network inference algorithms. Because there is no genome-scale regulatory network that has been fully characterized under all possible environmental conditions that an organism can exist in, network inference is essentially an unsupervised learning task, making the comparison of results from different algorithms on real data extremely difficult.

One approach to address this issue is to build mathematical simulations of regulatory networks based on the known biology of the gene regulation process. These simulated networks can then be run forward in time to generate measurements of the nodes of the simulated network, which can be used as input data to an inference algorithm. Since the true structure of the network is known, we can gauge the merits of a network inference algorithm by comparing the inferred network with the true network.

Since the gene regulation process is essentially a dynamic process existing network simulations employ a system of ordinary differential equations (ODE) that describe the kinetics of gene (mRNA) and protein concentrations as a function of time. Some of these approaches [8, 7] require a lot of user-specified information to build highly detailed models, making them impratical for genome-scale networks with greater than ten genes. Other approaches [6], can generate large networks but disregard crucial information such as protein expression and the combinatorial control of the transcription factors.

We describe an ODE-based network simulation system that is integrated into an existing ODE solver, Copasi [4], for building large and biologically realistic regulatory networks. We represent genes and proteins as separate entities, incorporating crucial details of the role of protein expression in the regulation process. We also design kinetic equations of transcription that explicitly capture the combinatorial control of transcription factors. Our simulation system requires only the number of genes as user input and automatically generates a network topology and associated kinetic equations in a Copasi-compatible format. Copasi can then be used to generate time-series as well as steady-state data of hundreds of genes and proteins, which can then be leveraged to perform comparitive analysis of different network inference algorithms.

2 Related work

NetBuilder is a graphical user-interface enabled software that allows the description and simulation of gene regulatory networks [8]. The NetBuilder user interface provides logical gates to define relationships between the network components such as genes and proteins. These logical gates are associated with mathematical functions that are invoked during a simulation run. Although, NetBuilder supports complex, non-linear interactions between transcription factors, the model can be generated only via the user interface making the description of big networks (> 10 genes) very cumbersome.

Mendez *et al.* have generated a collection of artificial genetic networks (AGN) in Copasi compatible format, which can be used to generate time-series and steady-state data [6]. However, in these networks genes and proteins are treated as the same entity. The transcription rate of a gene is dependent upon the concentration of other genes rather than proteins, which inherrently assumes that gene expression is completely correlated with protein expression. However, in real biological systems, protein expression does not correlate with gene expression due to different translation and degradation rates of proteins [2, 3]. Finally, the transcription kinetics do not account for combinatorial control among multiple transcription factors.

Our simulation system combines the high throughput capabilities of the AGN system and the biological richness of the NetBuilder model.

3 Transcriptional regulatory network simulation system

3.1 Transcription

Transcription is the process of making messenger RNA (mRNA) from the DNA sequence of a gene. Transcriptional regulation of gene expression, that is, how much of mRNA is made, is a major driving force of an organism's growth and survival in healthy and perturbed conditions. Although the transcription process and its regulation is achieved by a complex, multi-level machinery of interacting bio-chemical components, we will focus only on the role of protein interactions and transcription factor proteins in the regulation process.

Transcriptional regulation at the protein level is achieved by trancription factors binding to different promoter regions of genes under different environmental conditions. Depending upon the set of transcription factors that are bound to the promoters, RNA polymerase II is recruited with different efficiency, which in turn influences the rate of transcription. The genes to which a transcription factor binds for transcriptional activation or repression are called the targets of the transcription factor. Several transcription factors may bind to the same gene in different combinations resulting in different rates of transcription. For example, if two transcription factors can bind to a gene, there are four possible combinations of the transcription factors, that may be present on the promoter region of the gene. In addition, when multiple transcription factors are bound simultaneously, the net effect on the transcription rate is not a linear function of the effects of the individual transcription factors. Different binding combinations may be induced by different environmental conditions, resulting in a complex, combinatorial and non-linear control on transcription. Our goal is to develop a simulation system that models this combinatorial control on gene regulation.

3.2 Modeling transcription

Our simulation framework is described in two parts: (a) the network topology connecting the genes and proteins, and (b) the set of differential equations specifying the kinetics of transcription as determined by the network topology.

3.2.1 Network topology

The regulatory networks that we generate are composed of genes and proteins as nodes in the network. We assume that every gene is coding and produces a unique protein. We assume that all the proteins are transcription factors and have activating and repressing actions on different sets of genes. The protein nodes are connected by an undirected network with scale-free topology [1]. Each gene has an outgoing arc to it's coding protein. Each transcription factor has outgoing arcs to its target genes.

We associate each gene to a set of transcription factors using an exponential distribution describing the in-degree distribution of the genes [5]. The in-degree of a gene may be zero implying that the gene is self-regulatory. Since we want to incorporate the role of the protein interaction network, we allow individual proteins as well as cliques of proteins to be in-neighbours of the genes. The cliques are generated by exhaustive enumeration in the protein network and represent transcription factors that must function as a single unit to produce the desired activating or repressing effects. See Section 3.3 for more details.

3.2.2 Differential equation model

The system of differential equations have been adapted and enhanced from the existing work from NetBuilder and AGN [6, 7, 8]. We denote $P_i(t)$ as the concentration of the i^{th} protein at time t, where $1 \leq i \leq n$, n being the number of genes in the network. The rate of change of $P_i(t)$, $\frac{dP_i(t)}{dt}$, is dependent upon its synthesis and degradation:

$$\frac{dP_i(t)}{dt} = U_i G_i(t) - u_i P_i(t)$$

Here $G_i(t)$ is the concentration of the gene that codes for the i^{th} protein at time t. The synthesis term, $U_i G_i(t)$ specifies that the amount of P_i being produced at time t is a linear function of the amount of G_i at that time. The degradation term, $u_i P_i(t)$ describes P_i degradation as a linear function of the concentraton, $P_i(t)$ at any point in time. The terms, U_i and u_i are the i^{th} protein's synthesis and degradation rate constants. We set U_i and u_i individually for each protein by drawing samples from a uniform distribution, $[0.01, 0.1]$.

The rate of change of $G_i(t)$, $\frac{dG_i(t)}{dt}$ is also written as a difference between the synthesis and degradation of G_i:

$$\frac{dG_i(t)}{dt} = V_i S_{G_i} - v_i G_i(t) \tag{1}$$

Similar to the proteins, degradation is a linear function of the gene concentration, where v_i is the degradation constant for the i^{th} gene. V_i is the basal rate of transcription for the i^{th} gene and assigned values drawn from a uniform distribution, $[0.01, 1]$. The term, S_{G_i} is a function of the concentrations of the transcription factors of G_i. The gene synthesis function is the crucial part of our model as it specifies the combinatorial control of the transcription factors and takes into the account the protein interaction topology. We describe it in the next section.

3.3 Modeling the combinatorial control of transcription factors

The combinatorial control of transcription factors is captured by the synthesis part of the right hand side of the differential equation defining the temporal dynamics of each gene concentration. We first identify cliques in the protein interaction network by using exhaustive enumeration. This is possible since we consider a few hundreds of proteins and the network is sparsely connected resulting in small cliques. We then use the set of all cliques, C, to determine the transcription factors of a gene.

Previous work has shown that the probability distribution of the in-degree, k, in the yeast regulatory network is an expoential distribution, $P(k) = \lambda e^{-\lambda k}$ [5]. So, we set the in-degree for each gene, k, to a sample drawn from the exponential distribution, $P(k) = (0.42)e^{-0.42k}$, where $\lambda = 0.42$ from Maslov *et al.*

We select k in-neighbours of a gene by drawing uniformly at random, k elements from the set C. We refer to the set of k cliques that are in-neigbours of the i^{th} gene as F_i. Each clique in F_i is assigned an activating or repressing role by flipping a bit at random. All the proteins in the clique are assigned the same repressing or activating role. Since the number of combinations of k elements grows exponentially ($2^k - 1$), we restrict the number of combinations for a gene to a hard threshold of $m = 10$. If the $2^k - 1 > 10$, then we select uniformly at random 10 combinations from the set of $2^k - 1$ combinations. We refer to each combination of cliques as a transcription factor complex. To summarize our process of creating transcription factor complexes, we first find cliques in the protein network. We then select k cliques for regulating a gene, and then use at most 10 different combinations of these k to determine the synthesis function of a gene.

From the above paragraph, we note that there are two levels of interactions occuring at the transcription factor level. The first level of interaction, *within-clique* interaction, captured in a clique is determined by the protein interaction network. The second level of interaction, *across-clique* interaction, is between cliques and therefore between the transcription factors, induced by the specific combination of cliques we pick. We now describe the mathematical functions that capture these two types of interactions.

3.3.1 Within clique interaction

The clique, c_j, $1 \leq j \leq |F_i|$ associated with the i^{th} gene, is a set of proteins with indices $\{r_1, \cdots, r_l\}$, where l is the number of proteins in c_j. Let A_{ij} denote the action of the clique, c_j on the i^{th} gene. If c_j is assigned an activating role then

$$A_{ij} = \frac{\prod_{p=1}^{l} P_p(t)}{\prod_{p=1}^{l} Ka_{ip} + \prod_{p=1}^{l} P_p(t)}$$

Here $P_p(t)$ is the concentration of the protein, P_p at time, t. Ka_{ip} is the equilibrium dissociation constant of the *a*ctivator P_p of the i^{th} gene.

If c_j is assigned a repressing role then

$$A_{ij} = \frac{\prod_{p=1}^{l} Ki_{ip}}{\prod_{p=1}^{l} Ki_{ip} + \prod_{p=1}^{l} P_p(t)}$$

where Ki_{ip} is the equilibrium dissociation constant of the *i*nhibitor P_p of the i^{th} gene.

3.3.2 Across clique interaction

The across clique interactions are captured in S_{G_i}, which was first introduced in equation 1. This S_{G_i} function is a weighted sum of contributions of the Q different transcription factor complexes, corresponding to the randomly selected clique combinations from the total $2^k - 1$ combinations. The contribution of the q^{th} transcription factor complex, x_q is simply the product of the action of each clique, c_j, A_{ij} in the q^{th} complex:

$$x_q = \prod_j A_{ij}$$

The weight of the q^{th} complex, w_q, determines how active the complex is in particular simulation setting. These weights can be changed to simulate the effect of different environmental conditions on the activity of the complex. S_{G_i} for the i^{th} gene is written as

$$S_{G_i} = \sum_{q=1}^{Q} w_q x_q \tag{2}$$

Thus, in this manner we combine the protein interaction network, protein expression as well as the combinatorial role of transcription factors into a mathematical model of gene regulation.

4 Results

We describe simulation results of a small network of $n = 5$ genes and $n = 5$ proteins with the topology shown in Fig 1. The gene synthesis functions are described

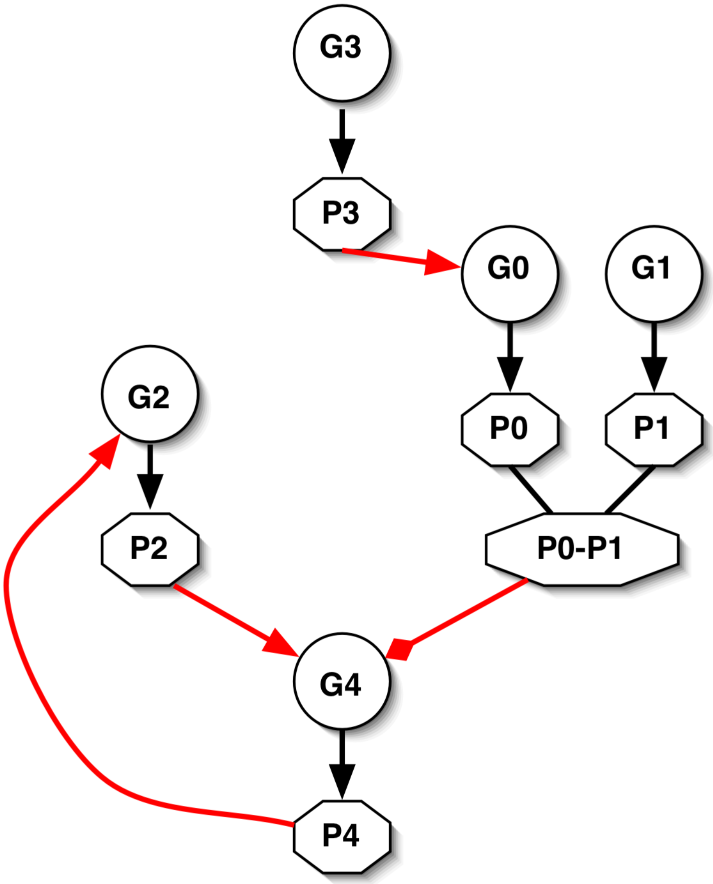


Figure 1: Example network of 5 genes and proteins. The red arrows denote the regulatory actions, black arrows denote transcription and the undirected arcs denote the protein interactions forming a complex. The arrow heads denote the type of regulation.

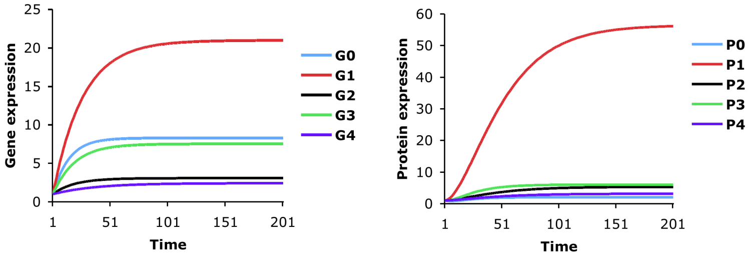
in Table 1. We first illustrate the importance of modeling protein expression as a separate entity and then we demonstrate the effect of perturbing different types of genes in the network. In particular, we show the effect of perturbing a protein clique that regulates another gene.

4.1 Combinatorial control of transcription

The network shown in figure 1 is composed of 5 genes and 5 proteins. The genes G_3 and G_1 are self-regulatory. The protein P_3 coded by gene G_3 is an activator for G_0. The proteins P_0 and P_1 coded by the genes G_0 and G_1 interact to form the two protein clique, $P_0 - P_1$, which has a repressive effect on G_4. The protein P_2 encoded by gene G_2 has an activating effect on G_4 which codes for P_4. This in

Gene	Synthesis function
$G0$	$V_0\left(\frac{P_3}{Ka_{0,3}+P_3}\right)$
$G1$	V_1
$G2$	$V_2\left(\frac{P_4}{Ka_{2,4}+P_4}\right)$
$G3$	V_3
$G4$	$V_4w_0\left(\frac{P_2}{Ka_{4,2}+P_2}\right)+V_4w_1\left(\frac{Ki_{4,0}Ki_{4,1}}{Ki_{4,0}Ki_{4,1}+P_0P_1}\right)$ $+V_4w_2\left(\frac{P_2}{Ka_{4,2}+P_2}\right)\left(\frac{Ki_{4,0}Ki_{4,1}}{Ki_{4,0}Ki_{4,1}+P_0P_1}\right)$

Table 1: Synthesis functions for the five genes



(a) (b)

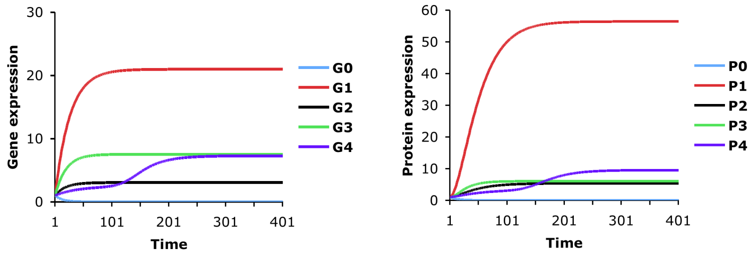
Figure 2: Timecourse of five genes and five proteins

turn activates G_2 resulting in a regulatory loop.

The only gene with combinatorial control is G_4, with cliques $\{P_2\}, \{P_0, P_1\}$ acting in 3 different possible combinations. We assume all the combinations have equal weights. The time-course of the five genes and protein under normal conditions (wild-type) is shown in Fig 2. From the time-course of the genes and proteins we see that although gene expression time-courses are different reaching in different steady-state expressions, their corresponding protein expression time-courses are quite similar with the exception of P_1. This shows that even if the individual gene and protein expression profiles are correlated to each other, the steady-state profiles of the overall set of genes and proteins are not correlated.

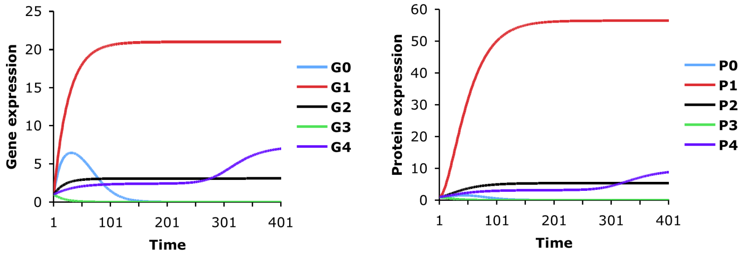
4.2 Perturbation effects in the network

We observed the effects of perturbing different genes in the network described in previous section. We first knocked out G_0, coding for P_0, which interacts with P_1 to form a repressive clique for G_4 (Fig 3). We observe that the gene and protein expressions of G_4 are much higher in this case than in wild-type (Fig 2). This indicates that on knocking out the single gene G_0 we also prevent the formation of the protein clique $P0 - P1$ and in turn remove the complete repressive effect on G_4. This illustrates a possible mechanism by which the protein network influences the regulation model. Similar to the wild-type expression, we observe that the gene



(a) (b)

Figure 3: Timecourse of five genes and five proteins after knocking out gene $G0$



(a) (b)

Figure 4: Timecourse of five genes and five proteins after knocking out gene $G3$

and protein expression time-courses are not completely correlated. For example the gene expression profiles of G_3 and G_4 look very similar in the later half of the time-course but the corresponding protein expression profiles, P_3 and P_4, are different in the later half, resulting in different steady-state expressions. This uderscores the importance of modeling protein expression separately from gene expression.

We show similar results of perturbing gene G_3 which is an activator for G_0 (Fig 4). Knocking out G_3 drives the the expression of G_0 and P_0 close to zero. We see that for G_2 there is no appreciable change in time-course and steady-state profiles as compared to the wild-type. This is because, G_4 which encodes an activator for G_2 is still active. For G_4 we observe that its time-course and as well as steady-state profiles are diferent from the wild-type, with the steady-state expression being higher for the perturbation than the wild-type. This is because G_3 codes an activator for G_0, which is absent, resulting in the disruption of the repressive complex, $P_0 - P_1$, and therefore an up-regulation of G_4. This illustrates a cascade of changes triggerred by a single perturbation that percolates down the chain of physical interactions, which is characteristic of true biological networks.

5 Future work

In this paper we have described a gene regulatory simulation framework that models the combinatorial control of transcription factors on gene regulation and does not assume that gene expression can explain everything about protein expression. Our framework requires minimal user input and automatically generates gene and protein time-series and steady-state expressions for several hundreds of nodes.

An important direction of future work we want to pursue is the incorporation of non-coding RNAs (ncRNA) which exercise a post-transcriptional control on gene expression. We will explore the cross-talk between the ncRNA-mediated regulatory control and the transcription factor-mediated control in our simulations. We also intend to incorporate genes or proteins which do not reach a single steady-state but have oscillatory behaviour. Such oscillations are quite frequent in nature and coupling these dynamics with the existing simple dynamics will enable us to generate networks that are closer to true regulatory networks. Finally, we will explore different types of perturbations including gene knockouts that re-wire the network, thus simulating the effect of environmental changes on biological networks. The data generated from these differentially re-wired networks can be used to compare and contrast different network inference algorithms in their ability to correctly infer the true network topology under different types of perturbations.

References

[1] Reka Albert and Albert-Laszlo Barabasi. Topology of evolving networks: local events and universality. *Phys. Rev. Lett.*, 85:5234–5237, 2000.

[2] Archana Belle, Amos Tanay, Ledion Bitincka, Ron Shamir, and Erin L. O'Shea. Quantification of protein half-lives in the budding yeast proteome. *Proceedings of the National Academy of Sciences*, 103, 2006.

[3] Andreas Beyer, Jens Hollunder, Heinz-Peter Nasheuer, and Thomas Wilhelm. Post-transcriptional expression regulation in the yeast saccharomyces cerevisiae on a genomic scale. *Molecular and Cellular Proteomics*, 3:1083–1092, 2004.

[4] S Hoops, S Sahle, R Gauges, C Lee, J Pahle, N Simus, M Singhal, L Xu, P Mendes, and U Kummer. Copasi – a complex pathway simulator. *Bioinformatics*, 22:3067–3074, 2006.

[5] Sergei Maslov and Kim Sneppen. Computational architecture of the yeast regulatory network. *Physical Biology*, 2:s94–s100, 2005.

[6] Pedro Mendes, Wei Sha, and Keying Ye. Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, 19:122–129, 2003.

[7] Maria J. Schilstra and Hamid Bolouri. Modelling the regulation of gene expression in genetic regulatory networks.

[8] Maria J. Schilstra and Hamid Bolouri. The logic of gene regulation. In *Poster abstract for Third International Conference on Systems Biology*, 2002.