

Reconstruction of Gene Regulatory Networks under the Finite State Linear Model

Dace Ruklisa^{1,3} Alvis Brazma² Juris Viksna^{1,4}
 Dace.Ruklisa@mii.lu.lv brazma@ebi.ac.uk jviksna@cclu.lv

¹ Institute of Mathematics and Computer Science, University of Latvia, Rainis boulevard 29, Riga LV-1459, Latvia

² European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, CB10 1SD, UK

³ supported by ESF (European Social Fund) project 2004/0001/VPD1/ESF/PIAA/04/NP/3.2.3.1./0001/0063 and Latvian Council of Science grant 05.1535

⁴ supported by EPSRC grant EP/C00373X/1 and Latvian Council of Science grant 05.1535

Abstract

We study the Finite State Linear Model (FSLM) for modelling gene regulatory networks proposed by A. Brazma and T. Schlitt in [4]. The model incorporates biologically intuitive gene regulatory mechanism similar to that in Boolean networks, and can describe also the continuous changes in protein levels. We consider several theoretical properties of this model; in particular we show that the problem whether a particular gene will reach an active state is algorithmically unsolvable. This imposes some practical difficulties in simulation and reverse engineering of FSLM networks. Nevertheless, our simulation experiments show that sufficiently many of FSLM networks exhibit a regular behaviour and that the model is still quite adequate to describe biological reality.

We also propose a comparatively efficient $O(2^K n^{K+1} M^{2K} m \log m)$ time algorithm for reconstruction of FSLM networks from experimental data. Experiments on reconstruction of random networks are performed to estimate the running time of the algorithm in practice, as well as the number of measurements needed for successful network reconstruction.

Keywords: gene regulatory networks, network reconstruction

1 Introduction

Various mathematical models describing gene regulatory networks as well as algorithms for network reconstruction from experimental data have been a subject of intense studies during the recent years. This is largely motivated by the availability of high-throughput experimental data. Experimental data are obtained mainly from RNA microarray experiments, which can give information about the expression level (transcription activity) of many genes at a time; very recently protein arrays have also become available. It is possible, at least in principle, to reconstruct gene regulatory networks just from these measurements.

Still, one cannot say that currently we are capable to infer gene networks from experimental data successfully. The largest obstacle to achieve this still seems to be the quality/quantity of available data — the error rate in measurements is quite high, there is no widely accepted method to compare the data from different experiments, some network models require hopelessly large amount of data for network reconstruction.

However, the problem of network inference is not completely solved also from the perspective of computer science. First, we need to find suitable models to describe gene regulation. These models should be sufficiently accurate in representing biological reality. Second, we need to provide algorithms

for reverse engineering of networks from experimental data. Unfortunately, these requirements can be considered as somewhat contradictory - the more subtle the image of biological reality in the model, the harder is the problem of network reconstruction.

Different models describe gene regulation at various levels of detail and simplicity. Boolean network models [1, 12] emphasize discrete aspects of regulatory interactions. The model includes only one regulatory parameter: the expression level of gene that can be in two states (expressed/not expressed). For each gene the dependence of its expression from expression of other genes is defined by a Boolean function. Despite the simplicity of this model, dynamics of several biological networks have been successfully modelled inside this framework [7]. Boolean networks have a significant advantage, i.e. the reconstruction of genetic network from experimental measurements is relatively simple and fast [2, 9, 11, 15].

Another class of network models is based on differential equations [5, 7, 16]. These models successfully depict continuous behaviour of genetic networks. Typically they include continuous variables that describe the expression level of gene, but the interaction between genes is defined by differential or difference equation. Probably the most popular model of this class is linear model, where types of interaction are constrained to those definable by linear equations [7, 16]. This model has an algorithm for network reconstruction from gene expression data. However, network reconstruction problem in general is quite complex for this model class, as it involves the solving of systems of differential equations.

To overcome the limits of the expressiveness of previous models, several models that unite discrete and continuous components are proposed [4, 6]. These models have continuous parameters for protein concentrations and discrete (Boolean) parameters for expression level of gene. Thus continuous changes in environment, as well as fast changes in the expression level of gene can be reflected. These properties make this model class particularly attractive, because biological mechanisms can be captured quite adequately. Currently qualitative behaviour of gene networks is widely studied with these models [3, 6]. The main problem within this class is the lack of any network reconstruction algorithm. It can be explained with large amount of different parameters that are used to define a genetic network.

In this paper we consider Finite State Linear Model (FSLM) proposed in [4]. This model combines discrete and continuous aspects of gene regulation and is consistent with current knowledge of biological gene regulation. The continuous part of the model consists of the state of protein concentrations. States of promotor regions are modelled with discrete components. Promotor region determines the expression level of gene that can be in a finite number of states. Expression level defines the change rate of protein's concentration. Proteins attach to or detach from a promotor region, as they reach certain thresholds, thus influencing the state of promotor.

The FSLM has several advantages. Firstly, it incorporates a "believable" Boolean-type model of gene regulation, as well as continuous changes of protein concentrations (those lacking pure Boolean models). In [4] the authors have succeeded to construct a biologically adequate network for lambda-phage using FSLM. Secondly, as we show in this paper, it is still possible to have an efficient network reconstruction algorithm that requires the number of data-points similar to what is required for Boolean networks.

In this paper we consider several theoretical properties of FSLM that were left unsolved in [4]. These properties might help to understand the biological significance of this model better. Some of the questions one might wish to answer are as follows.

What kind of network dynamics could be modelled inside this framework? Does this model incorporate only networks with periodic behaviour? Is it possible to describe chaotic network dynamics? These questions are connected with biological meaning and interpretation of networks defined in this model. Networks in real life show certain order and regularity. How close model instances resemble biological networks? In what sense the model is wider or narrower than the genetic network domain?

Somewhat surprisingly, but we were able to show that the problem of FSLM network behaviour is algorithmically undecidable in the general case (in particular, the problem whether a given gene will

ever reach an active state is algorithmically unsolvable). The first impression might be that networks generally will be chaotic. However, simulation experiments show that a sufficiently large amount of random networks exhibit periodic behaviour. Thus FSLM is still a candidate for biologically meaningful model. At the same time, undecidability result implies that there are no general algorithms for several practical problems, such as periodicity of a network or equivalence of two networks. This leaves network simulation as the only tool for getting “believable” answers to above mentioned questions.

We also address the problem of network reconstruction from experimental data in FSLM. In [4] it has been proven that there exists reverse engineering algorithm for this model. However the given algorithm is utterly impractical and is based on the enumeration of all possible candidate networks. Here we propose a more realistic algorithm with time complexity and requirements for the number of measurements similar to the known algorithms for Boolean networks.

The paper is organised as follows. In the next chapter we define Finite State Linear Model and state (without detailed proofs) some of its properties. In chapter 3 we introduce reverse engineering algorithm of FSLM networks. Chapter 4 describes several simulation experiments that assess periodicity of random FSLM networks as well as performance of network reconstruction algorithm.

2 The Finite State Linear Model

The Finite State Linear Model is based on several assumptions about the biological machinery of gene regulation [4]:

1. The state of transcription factor binding sites in the promotor region determines the gene activity,
2. Each binding site can be in a finite number of states (binding site is either free or it is occupied by a protein from a predefined set),
3. Gene activity can be in a finite number of levels, level depending of the state of promotor region,
4. If a gene is active, the concentration of protein produced by the gene is growing with rate dependent on the activity level, otherwise concentration is decreasing,
5. The state of a binding site depends on the concentration of the transcription factor.

We define the formal model consistent with these assumptions. We restrict it to a case where each binding site has two possible states: it is either free or occupied by a predefined protein.

Let $\Gamma = \{0, \dots, n-1\}$ be a set of n genes. With each pair of genes (i, j) , $i, j \in \Gamma$, we associate a Boolean variable $v_{ij} \in \{\text{true}, \text{false}\}$, which characterize the state of binding site near gene i that can be occupied by protein j . By \mathbf{R}_+ we denote the set of non-negative real numbers, and by \mathbf{R}_- the set of non-positive real numbers. 2^S is a set of all subsets of S .

Definition. A gene net work is a 7-tuple $N = \langle \Gamma, r_1, r_2, B, L_a, L_d, F \rangle$, where:

- $r_1 : \Gamma \rightarrow \mathbf{R}_+, r_2 : \Gamma \rightarrow \mathbf{R}_-$ (for each gene i the values $r_1(i)$ and $r_2(i)$ are correspondingly the growth rate of protein, when gene i is expressed, and the degradation rate of protein, when gene i is non-expressed),
- $B : \Gamma \rightarrow 2^\Gamma$ (for each gene i the set $B(i)$ can be considered as a set of *binding sites*, i.e. it contains genes that regulate i),
- $\forall i \in \Gamma : l_{a,i} : B(i) \rightarrow \mathbf{R}_+, l_{d,i} : B(i) \rightarrow \mathbf{R}_+$ and $L_a = (l_{a,0}, \dots, l_{a,n-1}), L_d = (l_{d,0}, \dots, l_{d,n-1})$ (for each binding site there are two constants: association constant $l_{a,i}(j)$ for protein j at site near gene i and dissociation constant $l_{d,i}(j)$ for protein j at site near gene i ; without loss of generality one can assume that $l_{d,i}(j) < l_{a,i}(j)$, otherwise the site will not be functional),

- $F = (F_0, \dots, F_{n-1})$, where for each i function F_i is a Boolean function with variables from the set $\{v_{ij} : j \in B(i)\}$ (F_i describes how the activity of gene i depends on the states of its binding sites; gene i is active when $F_i = \text{true}$, inactive if $F_i = \text{false}$).

The behaviour of gene network can be described by functions $c_i : \mathbf{R}_+ \rightarrow \mathbf{R}_+$, where $c_i(t)$ denotes the concentration of gene i product (i.e. protein) at time instant t , as well as by functions $v_{ij} : \mathbf{R}_+ \rightarrow \{\text{true}, \text{false}\}$, where $v_{ij}(t)$ denotes the state of binding site at time instant t . By $F_i(t)$ we denote the value of function F_i at time instant t , where functions $v_{ij}(t)$ are substituted for variables v_{ij} . We also have to assume that at time instant $t = 0$ there are initial concentrations $c_i(0)$ and *consistent* initial states of binding sites $v_{ij}(0)$ are given. (We will say that the state of binding site $v_{ij}(t)$ is *consistent*, if either $v_{ij}(t) = \text{true}$ and $c_i(t) > l_{a,j}(i)$, or $v_{ij}(t) = \text{false}$ and $c_i(t) < l_{a,j}(i)$). We will refer to the set of initial concentrations $c_i(0)$ together with the set of consistent initial states of binding sites $v_{ij}(0)$ as an *initial state* of network.

Now, let $t_0 < t_1 < t_2 < \dots$ be a sequence of time points, such that $t_0 = 0$ and set $\{t_1, t_2, \dots\}$ contains all moments t at which $c_i(t) = l_{a,j}(i)$ or $c_i(t) = l_{d,j}(i)$ for some genes i and j . (That is, the sequence will contain all time points at which the state of some binding site changes.) Usually this sequence will be infinite, although in special situations it can be finite, in which case we assume that the sequence ends with $t_k = +\infty$.

Then the concentration $c_i(t)$ at time instant $t \in (t_k, t_{k+1}]$ can be defined as follows:

$$c_i(t) = \begin{cases} c_i(t_k) + r_1(i)(t - t_k) & \text{if } F_i(t_k) = \text{true}, \\ \max(c_i(t_k) + r_2(i)(t - t_k), 0) & \text{if } F_i(t_k) = \text{false}. \end{cases}$$

And the state $v_{ij}(t)$ of binding site at time instant $t \in (t_k, t_{k+1}]$ can be defined as follows:

$$v_{ij}(t) = \begin{cases} \text{true} & \text{if } c_i(t_k) \geq l_{a,j}(i) \text{ or } v_{ij}(t_k) = \text{true} \text{ and } c_i(t_k) > l_{d,j}(i), \\ \text{false} & \text{if } c_i(t_k) \leq l_{d,j}(i) \text{ or } v_{ij}(t_k) = \text{false} \text{ and } c_i(t_k) < l_{a,j}(i). \end{cases}$$

Less formally the model can be described like this. Each gene i can be in one of two states: active or inactive. If gene is active, it produces some product (protein) with the rate $r_1(i)$, if gene is inactive, the protein degrades with the rate $r_2(i)$. The concentration of protein at a time instant t is given by $c_i(t)$. There is also a set of binding sites associated with each gene. Each binding site reacts to a product of particular gene j and becomes active, when the concentration of protein j reaches at least $l_{a,j}(i)$, and becomes inactive when the concentration of protein j decreases to $l_{d,j}(i)$. The activity of gene i is governed by some Boolean function F_i , the value of F_i depends on the activity/inactivity of the gene's i binding sites. Figure 1 shows the behaviour of a simple network with 3 genes.

There are some important properties of the model, which could be useful to know. Is it powerful/adequate enough to model biological reality? The results from [4] imply that the model could be appropriate in this context. Also, the model reasonably corresponds to our knowledge of biological machinery. The most noticeable objection could be that we do not place any explicit bounds on protein concentrations - arbitrarily large concentrations certainly are not possible in biological systems. However, then one can argue that there is some biological mechanism to keep concentrations down, which can be described by the same model. Namely, we can restrict our attention only to networks where concentrations do not exceed some upper bound (or only to genes having this property).

Then, networks in real life usually show certain order and regularity. How regular/ordered FSLM networks are? For example, are their behaviour periodic, or is it possible to construct a network where states never repeat? Another important question (especially, when one is trying to reconstruct networks from experimental data) is network equivalence problem. Is there an algorithm to decide whether two networks are equivalent?

Somewhat surprisingly for us, but we found that the answer to the last two questions is negative. Certain analogy with timed automata and hybrid systems can be drawn. Hybrid systems are more

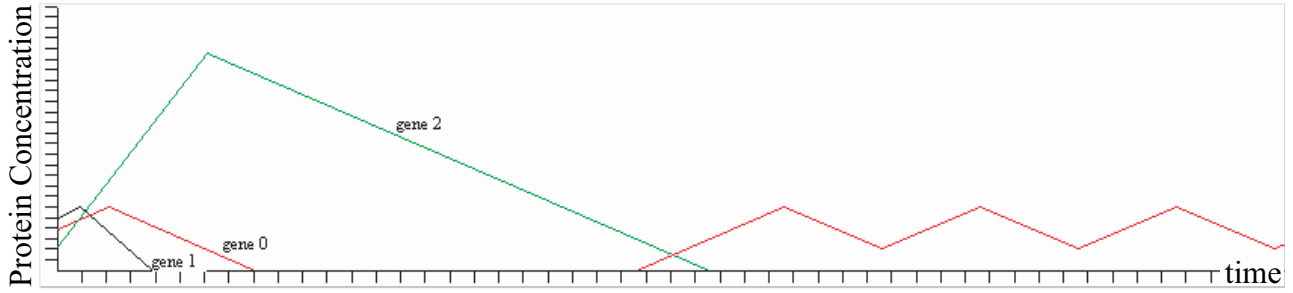


Figure 1: Behaviour of a simple network with 3 genes. Regulatory functions: $F_0 = v_{00} \ \& \ (\neg v_{01} \vee v_{02})$, $F_1 = v_{00} \vee v_{01}$, $F_2 = \neg v_{01} \vee v_{02}$; growth/degradation rates: $r_1(0) = 1$, $r_2(0) = 1$, $r_1(1) = 1$, $r_2(1) = 3$, $r_1(2) = 2$, $r_2(2) = 1$; association/dissociation constants: $l_{a,0}(0) = 6$, $l_{a,1}(0) = 4$, $l_{a,2}(0) = 5$, $l_{d,0}(0) = 2$, $l_{d,1}(0) = 3$, $l_{d,2}(0) = 1$, $l_{a,0}(1) = 7$, $l_{a,1}(1) = 4$, $l_{d,0}(1) = 2$, $l_{d,1}(1) = 2$, $l_{a,1}(2) = 5$, $l_{a,2}(2) = 3$, $l_{d,1}(2) = 1$, $l_{d,2}(2) = 2$; initial concentrations: $c_0(0) = 3.9$, $c_1(0) = 2.2$, $c_2(0) = 5$; initial states of binding sites: all sites for genes 0 and 2 inactive, both sites for gene 1 active.

powerful in capturing real-time processes than timed automata. But wideness of the model introduces problems that are not algorithmically solvable (for example, dividing real-time systems in equivalence classes).

Formally the equivalence of two networks can be defined as follows.

Definition. Two networks $N^{(1)}$ and $N^{(2)}$ with n genes each are *equivalent*, if there exists an initial state for each network, such that $c_i^{(1)}(t) = c_i^{(2)}(t)$ for all $i \in \{0, n-1\}$ and $t \geq 0$.

Definition. Two networks $N^{(1)}$ and $N^{(2)}$ with n genes each with given initial states are *behaviourally t -equivalent* for some time moment t , if $c_i^{(1)}(t') = c_i^{(2)}(t')$ for all $i \in \{0, n-1\}$ and $t' \in [0, t]$.

Let us define the network periodicity. To avoid restriction only to networks with bounded concentrations, we define periodicity taking into account only genes which affect network behaviour at the time considered.

Definition. A network N with n genes and given initial state is *periodic*, if there exists two time moments $t_1 < t_2$, such that for all $i, j \in \{0, n-1\}$ we have $v_{ij}(t_1) = v_{ij}(t_2)$ and either $c_i(t_1) = c_i(t_2)$, or $c_i(t_2) \geq c_i(t_1) > l$ and $c_i(t) > l$ for all $t \in [t_1, t_2]$ where $l = \max_{i,j} \{l_{a,j}(i)\}$.

Our main result of this section is given by the following theorem, which states that we cannot algorithmically decide whether a particular gene will ever become active (proof, although not particularly complicated, is technically complex; here we give just the general idea of it).

Theorem 1. For a given network $N = \langle \Gamma, r_1, r_2, B, L_a, L_d, F \rangle$ with given initial state and for a given gene $i \in \Gamma$ it is algorithmically undecidable whether there exists t such that $F_i(t) = \text{true}$.

Idea of proof. Every Minsky machine (push-down automaton with two counters) can be reduced to a gene network in FSLM, where network simulates the behaviour of Minsky machine. The reduction can be done in such a way that a specific gene becomes active only if Minsky machine stops with counter values at 0. It is known that the vertex reachability problem is undecidable for Minsky machines. Therefore, from reduction follows that the problem whether a particular gene will become active is undecidable for FSLM.

The proof assumes that concentrations are not bounded by some upper limit. As we already mentioned, the assumption is not true in biological networks. However, the situation might be analogous with using infinite Turing machines to model real computers (which are finite and can be described by finite automata). Similarly, restricted network models with a finite number of concentration levels could be “closer to reality”, but inappropriate for practical use.

Following undecidability results are either simple consequences of this theorem, or can be obtained by slightly modified proof.

Corollary 1. The problem to decide whether two networks are equivalent is algorithmically unsolvable. Also, the problem to decide whether for two networks with given initial states there exists time moment t such that networks are not behaviourally t -equivalent is algorithmically unsolvable.

Here follows the rather unpleasant conclusion that, when studying network reconstruction, in general it will be impossible to tell whether the reconstruction has succeeded (even if reconstructed network perfectly describes cell processes up to some measurement point, it can become completely chaotic just after that!). The best one can do, is to continue simulation of reconstructed network after that point, and, if it still conforms to the measurements long enough, assume that “very likely” it is correct. Also, if one notices that the network is periodic (it is often the case with biological data), then one can be sure about its future behaviour.

Corollary 2. For a given network with given initial state the problem to decide whether it is periodic is algorithmically unsolvable.

However, if one simulates a periodic network for sufficiently long time, the periodicity will become apparent. Also, the equivalence problems are easily solvable for periodic networks.

Theorem and corollaries imply that the model includes networks with acyclic dynamics. Nevertheless, networks in nature show certain regularity. This regularity might be achieved with some patterns of network architecture or with a restricted class of Boolean functions. For some other network models [13, 14], the authors have searched for the most probable patterns of architecture that enhance the stability of a system. Some authors suggest that most of the regulatory influences can be described with canalizing functions, which are quite stable to random change in some inputs [8, 10]. Periodicity and stability of networks is further analysed in the experimental part of the work. We show that a noticeable share of random networks is periodic, which could be selected by nature as “useful” one.

3 Reverse Engineering of Networks

From the previous chapter we can conclude that in principle we cannot be certain that our reconstructed network will be consistent with all additional measurements that may be made in future. On the other hand, this problem could be solvable for “regular” networks and is solvable, if “regular” is interpreted as periodic. As a positive result there is a comparatively efficient algorithm which reconstructs networks that are compatible with the given observations. The proposed algorithm has an additional requirement that measurements are made at all time points where activity of some gene changes.

For a network with n genes and some time moment t by $C(t)$ we denote the n -tuple $(c_0(t), c_1(t), \dots, c_{n-1}(t))$ of all protein concentrations at time moment t .

Definition. We say that a finite set of time points $T = \{t_0, \dots, t_{m-1}\}$ is *representative* for a network N , if T contains all time points t between t_0 and t_{m-1} at which activity of some gene changes, that is $\{t | t_0 \leq t < t_{m-1} \& \exists i \exists \delta > 0 : \forall \varepsilon \in (0, \delta) F_i(t) \neq F_i(t - \varepsilon)\} \subseteq T$.

Definition. For a given set of time points $T = \{t_0, \dots, t_{m-1}\}$ and a given network N a *measurement series* $M(T)$ is a set of the pairs of time points t and the corresponding concentration values $C(t)$, i.e. $M(T) = \{(t_0, C(t_0)), (t_1, C(t_1)), \dots, (t_{m-1}, C(t_{m-1}))\}$.

Definition. The network N with a given initial state is *compatible* with measurement series $M(T) = \{(t_0, C(t_0)), (t_1, C(t_1)), \dots, (t_{m-1}, C(t_{m-1}))\}$, if N produces concentrations $C(t_0), C(t_1), \dots, C(t_{m-1})$ at time points t_0, t_1, \dots, t_{m-1} .

The notion of compatibility is largely introduced for technical convenience and is closely related with behavioural t -equivalence. If measurement series $M(T)$ are obtained from network N at time points from some representative set $T = \{t_0 = 0, \dots, t_{m-1}\}$, then network N' is compatible with $M(T)$ if and only if N' and N are behaviourally t_{m-1} -equivalent.

The network reconstruction problem. Given a representative set of time points T for a network N and measurement series $M(T)$, find a network N' and its initial state that is compatible with $M(T)$.

In [4] it has been proven that the problem of network reconstruction is algorithmically solvable. Yet the proposed algorithm is based on the enumeration of all networks with fixed bounds on gene count and connectivity and testing the compatibility with measurements for each enumerated network. As a result the algorithm is of theoretical value only. Here we propose a new algorithm that can be used for practical purposes already and has a complexity similar to algorithms for Boolean network reconstruction.

The basic structure of the algorithm resembles that of algorithms for Boolean network reconstruction. However, instead of just inferring a Boolean formula for each gene, here we must also reconstruct the association and dissociation constants, protein growth and degradation rates and initial state of the network. We assume that the network has n genes and the constant *maxConnectivity* exceeds the maximum number of regulators for any gene. It is widely assumed that this value can vary between 4 and 8 in different networks [9].

Algorithm *ReconstructGeneNetwork*

Input: Representative set of time points T and measurement series $M(t)$
Output: Network N and its initial state S that is compatible with $M(t)$
begin **for** $i = 1, \dots, n$ **do**
 for $k = 1, \dots, \text{maxConnectivity}$ **do**
 for each set B of k genes **do**
 $r_1, r_2 \leftarrow \text{ComputeGrowthRates}(M(T), i)$
 $\Lambda_a, \Lambda_d \leftarrow \text{ComputePotentialThresholds}(M(T), i, B)$
 if $F, L_a, L_b, s \leftarrow \text{ConstructRegulatoryFunction}(M(T), i, B, \Lambda_a, \Lambda_d)$ **then**
 add gene i together with the corresponding values r_1, r_2, B, L_a, L_b, F
 to network N
 add the part of initial state s to initial state S
 go to next gene
 return network N and its initial state S
 end

The task of network reconstruction is reduced to the search of influence function for each gene i . The argument count of function is bound by a constant *maxConnectivity*. When potential regulator genes are chosen, the construction of growth and degradation rates, association and dissociation constants and regulatory function can start.

The construction of growth and degradation rates done by function *ComputeGrowthRates*($M(T), i$) is straightforward. We consider all possible pairs of values $c_i(t_x)$ and $c_i(t_{x+1})$. If we have $c_i(t_x) < c_i(t_{x+1})$, the value $r_1(i) = (c_i(t_{x+1}) - c_i(t_x)) / (t_{x+1} - t_x)$ is a growth rate. If we have $c_i(t_x) > c_i(t_{x+1})$, the value $r_2(i) = (c_i(t_x) - c_i(t_{x+1})) / (t_{x+1} - t_x)$ is a degradation rate. All these values should be consistent. If any of them cannot be computed due to unavailable data, we can assign to it an arbitrary value.

The non-trivial parts of this process are the search for potential thresholds and the search for a consistent regulatory function. Building of a consistent function includes both the choice of a threshold pair for each regulatory protein from the potential threshold set and creating a truth-table for regulatory function. The truth-table should be without contradictions (different values of function for the same values of arguments) and ambiguities (the value of the function should be determined for each combination of argument values).

Sets of potential thresholds Λ_a and Λ_d , computed by function *ComputePotentialThresholds*($M(T), i, S$), are determined after the protein concentrations at local extremes of the concentration graphs. We consider a concentration graph of protein i and graphs of potential regulators for i . Let us denote the sequence of time moments, when concentration of a protein h reaches a local extreme (i.e. time

moments when activity of gene h changes; we will exclude moments when the protein falls to 0 level and stays here for some time) with t'_1, \dots, t'_m . We look up the concentrations $c_j(t'_x)$ for all regulator proteins j at each moment t'_x . If the protein j grows before t'_x , then $c_j(t'_x)$ is a potential association threshold, and we add it to Λ_a , otherwise $c_j(t'_x)$ is a potential dissociation threshold, and we add it to Λ_d . Constants $+\infty$ and -1 are added to potential upper and lower thresholds.

In principle, we cannot guarantee that in this way we will find all the thresholds used while regulating gene i in initial network N . However, the following technical result (given here without proof) shows that we can choose unknown values from the sets Λ_a and Λ_d , and still obtain a network that is compatible with measurements $M(T)$.

Theorem 2. If it is possible to construct a consistent regulatory function F_i for a given combination of regulators $B(i)$ and a given gene i , then it is possible to choose a threshold pair for each regulator from the sets of potential thresholds Λ_a and Λ_d , such that a consistent regulatory function F'_i exists for this pair of thresholds.

Regulatory function, association and dissociation constants and a part of initial state corresponding to that particular gene are computed by function *ConstructRegulatoryFunction* ($M(T), i, B, \Lambda_a, \Lambda_d$). In general, this function searches through all threshold pairs from the sets Λ_a and Λ_d for each regulator protein and all consistent states of binding sites in B at time point t_0 (part of the initial state s can be derived from these values). When some combination of threshold pairs and binding site states is chosen, we try to reconstruct regulatory function F_i that is consistent with measurements $M(T)$. To construct a function we need to determine all time points when some binding site changes its state. Let us denote the sequence of these points with t_0, \dots, t_m . At each moment t_k we will compute the state of all binding sites in B and determine the expression of gene i (this can be done according to definitions). Consistent function for given thresholds and initial states exists if there are no time moments t_k and t_j , where states of binding sites are equal, but the expression differs. It can be tested by sorting binding site states by value and then checking whether neighbouring elements with equal binding site states have equal expressions of the gene i . If a consistent function exists, truth-table of F_i can be built by treating the states of binding sites as values of arguments and the expression of gene as a value of function. If some function values in truth-table remain undetermined, they can be filled with arbitrary bits.

If we succeed in constructing a consistent regulatory function, we return this function, the values of association and dissociation constants that are used, as well as the used initial state.

Some heuristics is used in search, which usually limit the number of pairs from the sets Λ_a and Λ_d that need to be considered, although in the worst case the exhaustive search still has to be performed.

To evaluate the complexity of the algorithm, at first we can notice that the number of sets of binding sites considered for each gene is of order C_n^k , where k is the number of binding sites for this gene. Since n is usually much larger than k , we can assume that this value is simply n^k .

Then, the growth rates can be computed in $O(m)$ time, where m is the number of time points, where a local extreme is reached. The time needed to compute the potential thresholds will depend on a particular network, but will not exceed $O(km \log m)$ (there can be up to m local extreme points, each giving up to m potential thresholds for each of up to k genes considered). Since this value affects the further complexity of the algorithm we denote it by M .

To construct a regulatory function, we need to consider up to M^{2k} threshold pairs and up to 2^k initial states, giving $2^k M^{2k}$ iterations for the regulatory function construction. This function can be constructed in time $O(m(\log m + k))$, or, since k is small, this can be assumed to be $O(m \log m)$.

Thus, in the worst case the time needed to process one gene will be $O(2^k n^k M^{2k} m \log m)$, and the reconstruction of the whole network will take time $O(2^K n^{K+1} M^{2K} m \log m)$, where K is the maximal number of binding sites for genes in the network.

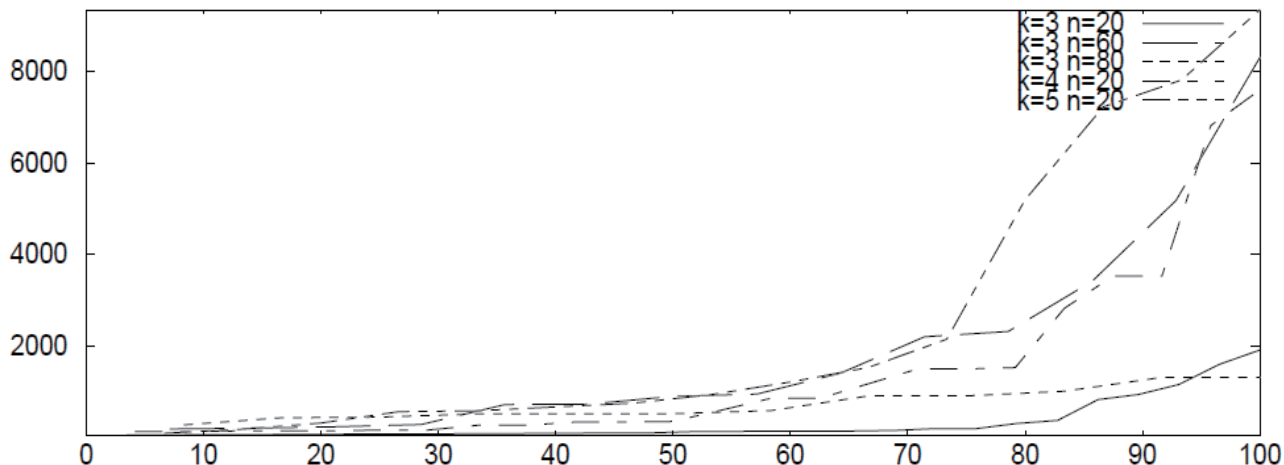


Figure 2: Distribution of steps at which networks become periodic for random networks with n genes and connectivity k . y axis shows the first step of period, x axis shows the percentage of networks with the first step of period not exceeding the corresponding y value.

4 Simulation and Reconstruction Experiments

In this section we include two types of experimental results: simulation experiments that are used to estimate periodicity of random FSLM networks and experiments, in which we use the algorithm from the previous chapter to reconstruct randomly generated FSLM networks to estimate practical running time of the algorithm as well as the number of time points needed for successful reconstruction of networks.

A series of experiments were performed to estimate the regularity of behaviour of random networks. In principle this task poses some challenges: the notion of periodicity might be too strong to cover all behaviours that we might want to consider as regular (even a network consisting of several periodic, but unrelated components could be hard to detect), also the use of real constants for growing rates, binding thresholds, etc does not correspond to biological reality. Non-periodic networks are very easily obtained by choosing growing rates with irrational ratios. However, such effects are impossible in biological networks due to the fact that all rates and thresholds will be approximate. Therefore, even to study periodicity one should be careful when selecting sets of rates and thresholds that will be used to generate random networks. In our experiments we used a set of 100 thresholds and initial values, and a set of 10 growth/degradation rates, such that the ratios between all these numbers are rational. Networks were simulated for 10000 steps (steps correspond to time points when the state of some binding site changes) and then tested for periodicity. 50 random networks were considered for several combinations of values of n (number of genes) and k (number of regulators for gene). Up to 56% of networks turned out to be periodic, although this value decreases for larger n and k . For some values of k and n the distribution of steps at which periodicity has been observed is shown in Figure 2. In general the experiments leave an impression that periodic networks are sufficiently common to consider FSLM model suitable for describing biological reality.

The connectivity of genes has the largest impact on time when a network reaches the period. When the connectivity is greater, networks reach the period much later. The best example is graphs for $k = 5, n = 20$ and for $k = 3, n = 80$. It means that networks with larger connectivity have a much greater complexity than networks with smaller k , and even large number of genes can hardly compensate this effect. Probably the period increases exponentially with an increase in k . The exponential influence of k is consistent with the reconstruction of gene networks. The number of genes

in a network leaves a smaller impact on the periodicity. A little growth of the period which coincides with the growth in the number of genes can be observed. The hypothesis is that this growth might be linear.

The undecidability results imply that in general there is no number of measurements that can guarantee a successful reconstruction of network. However, it is not hard to show that for the reconstruction of periodic networks it is sufficient for this number to exceed the number of steps at which periodicity can be observed. Thus, Figure 2 gives also some estimate of the number of measurements that might be needed for the network reconstruction (however, due to the issues mentioned above, these values still should be interpreted with caution).

Another set of experiments was performed to estimate practical running times of the reconstruction algorithm. These experiments were performed on networks with 20 to 100 genes using series of 100, 200 and 300 measurements. 50 networks with each parameter set were reconstructed. The running times are summarized in Table 1.

Table 1 shows a very strange feature that running times are decreasing with a growing number of genes. This is explained by the fact that the number of measurements is insufficient for the reconstruction of these networks and the data can be easily explained with networks having very small connectivities (regulators for a gene with a small connectivity can be found very quickly).

Other data that were obtained from these experiments are the number of genes with a particular connectivity in reconstructed networks (see Table 2). An interesting feature is that even in the case of mostly successful network reconstruction (20 genes, $k = 3$, $M = 100$) in the reconstructed networks in average there are 14.7 genes with connectivity 1, 4.6 genes with connectivity 2 and 0.7 genes with connectivity 3. The proportion of connectivity 1 seems quite notable. It can be explained by the fact that many proteins go to 0 or ∞ level immediately from the initial concentration (the expression of the corresponding gene never changes). To infer the actual regulatory mechanism of such genes we would need measurements of protein concentrations starting from a different initial state. Besides, many initial states lead to some attractor (in fact, attractor is not reached only when the network is not periodic). The attractor contains information about few states (states present in the period). Thus the network reconstruction algorithm sees only a small part of possible network behaviours and explains the given behaviour with simple functions and a small amount of regulators.

The connectivities of genes in the reconstructed networks give us clues about the number of time points needed to reconstruct a network. When $M = 100$ and $k = 3$, genes with connectivity 3 are present if $n = 20$, but their amount completely diminishes at $n = 80$ or $n = 100$. It means that $M = 100$ is not enough to reconstruct a network with, say, 100 genes.

The larger the number of genes in network, the larger number of measurements is needed for the reconstruction of this network. The number of necessary measurements is linearly dependent on the gene count, and is closely related to the average step when the period is reached. This is not very surprising, if we think that M is the total number of time points when the state of some binding site changes. If we divide M with a number of genes, we obtain the average amount of represented changes in the binding sites for one gene. It means that for $M = 100$ and $n = 100$ each protein reaches a threshold in average once. Clearly, the information obtainable from measurements, when $M = 100$ and $n = 100$, is not enough to reconstruct a network.

5 Conclusions

The FSLM model for gene regulatory networks is promising, and we are continuing network reconstruction experiments on simulated data. Of course, a real challenge will be to try to apply the method on some sets of real data. However, to achieve this we still need to develop a modified version of the algorithm that can treat inaccurate data (the source of inaccuracies will be measurement errors, as well as the fact that biological systems are very unlikely to behave exactly in accord with the FSLM model).

Table 1: Maximal and average (in brackets) running times for the network reconstruction algorithm for a given number of genes, connectivity k and number of time points M (in seconds).

	$k = 3, M = 100$	$k = 5, M = 100$	$k = 3, M = 200$	$k = 5, M = 200$	$k = 3, M = 300$
20 genes	118 (9.4)	687 (28.8)			
40 genes	18 (0.9)	341 (13.1)	462 (35.7)		
60 genes	4 (0.6)	11 (0.6)	391 (18.1)	2764 (142.8)	
80 genes	3 (0.3)	4 (0.4)	324 (26.2)	458 (19.2)	
100 genes	3 (0.2)	4 (0.3)	8 (4.4)	17 (3.3)	465 (46)

Table 2: Average number of genes with connectivity 1, 2 and 3 (separated by “/”) in reconstructed networks where original network’s $k = 3$ and with connectivity 1, 2, 3 and 4 where original network’s $k = 5$ (no genes with connectivity 5 were reconstructed).

	$k = 3, M = 100$	$k = 5, M = 100$	$k = 3, M = 200$	$k = 5, M = 200$	$k = 3, M = 300$
20 genes	14.7/4.6/0.7	11.5/7/1.4/0.1			
40 genes	34.6/5.3/0.1	34/5.8/0.2/0	27.7/11.1/1.2		
60 genes	56.7/3.2/0.1	56.4/3.6/0/0	47.8/11.6/0.6	44.6/13.9/1.5/0	
80 genes	77.9/2.1/0	77.1/2.9/0/0	70.2/9.2/0.6	67/12.3/0.7/0	
100 genes	98.6/1.4/0	98.2/1.8/0/0	91.4/8.4/0.2	89.1/8.7/0.2	81.7/17.1/1.2

Another important (and probably related) problem is the network reconstruction from measurement series at non-representative sets of time points. It is easy to show that, if the time points are sufficiently frequent (at least two measurements are done between each pair of time points from a representative set), the method described here can still be used for network reconstruction. In this case the given measurements will easily allow the reconstruction of a representative set of time points together with measurements at these points. However, network reconstruction from measurement series at sparser non-representative sets of time points is still an open problem.

References

- [1] Akutsu, T., Miyano, S., and Kuhara, S., Identification of genetic networks from a small number of gene expression patterns under the Boolean network model, *Pacific Symposium of Biocomputing*, 4:17–28, 1999.
- [2] Akutsu, T., Miyano, S., and Kuhara, S., Inferring qualitative relations in genetic networks and metabolic pathways, *Bioinformatics*, 16:727–734, 2000.
- [3] Batt, G., Ropers, D., de Jong, H., Geiselman, J., Mateescu, R., Page, M., and Schneider, D., Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in *Escherichia coli*, *Bioinformatics*, 21:i19–i28, 2005.
- [4] Brazma, A. and Schlitt, T., Reverse engineering of gene regulatory networks: a finite state linear model, *Genome Biol.*, 4:P5:1–31, 2003.
- [5] Chen, T. E., He, H. L., and Church, G. M., Modeling gene expression with differential equations, *Pacific Symposium of Biocomputing*, 4:29–40, 1999.

- [6] de Jong, H., Page, M., Hernandez, C., and Geiselman, J., Qualitative simulation of genetic regulatory networks: method and application, In Nebel, B., ed., *Proc. IJCAI-01*, Morgan-Kaufmann, 67–73, 2001.
- [7] D’Haeseleer, P., Liang, S., and Somogyi, R., Genetic network inference: from co-expression clustering to reverse engineering, *Bioinformatics*, 16:707–726, 2000.
- [8] Harris, S. E., Sawhill, B. K., Wuensche, A., and Kauffman, S., A model of transcriptional regulatory networks based on biases in the observed regulation rules, *Complexity*, 7:23–40, 2002.
- [9] Ideker, T. E., Thorsson, V., and Karp, R., Discovery of regulatory interactions through perturbation: inference and experimental design, *Pacific Symposium of Biocomputing*, 5:302–313, 2000.
- [10] Kauffman, S., Peterson, C., Samuelsson, B., and Troein, C., Random Boolean network models and the yeast transcriptional network, *Proc. Natl. Acad. Sci. USA*, 100:14796–14799, 2003.
- [11] Kyoda, K. M., Morohashi, M., Onami, S., and Kitano, H., A gene network inference method from continuous-value gene expression data of wild-type and mutants, *Genome Informatics*, 11:196–204, 2000.
- [12] Liang, S., Fuhrman, S., and Somogyi, R., REVEAL, a general reverse engineering algorithm for inference of genetic network architectures, *Pacific Symposium of Biocomputing*, 3:18–29, 1998.
- [13] McAdams, H. and Arkin, A., It’s a noisy business! Genetic regulation at the nanomolar scale, *Trends Genet.*, 15:65–69, 1999.
- [14] Savageau, M. A., Rules for the evolution of gene circuitry, *Pacific Symposium of Biocomputing*, 3:54–65, 1998.
- [15] Shmulevich, I., Yli-Harja, O., and Astola, J., Inference of genetic regulatory networks under the best-fit extension paradigm, In *Proceedings of the IEEE-EURASIP Workshop of Nonlinear Signal and Image Processing (NSIP-01)*, June 3–6, Maryland, Baltimore, 2001.
- [16] Weaver, D. C., Workman, C. T., and Stormo, G. D., Modeling regulatory networks with weight matrices, *Pacific Symposium of Biocomputing*, 4:112–123, 1999.