iopscience.iop.org

Home Search Collections Journals About Contact us My IOPscience

A simple method for reverse engineering causal networks

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

2006 J. Phys. A: Math. Gen. 39 L647

(http://iopscience.iop.org/0305-4470/39/46/L01)

View the table of contents for this issue, or go to the journal homepage for more

Download details:
IP Address: 171.66.16.108
The article was downloaded on 03/06/2010 at 04:56

Please note that terms and conditions apply.

INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF PHYSICS A: MATHEMATICAL AND GENERAL

J. Phys. A: Math. Gen. **39** (2006) L647–L655 doi:10.1088/0305-4470/39/46/L01

LETTER TO THE EDITOR

A simple method for reverse engineering causal networks

M Andrecut and S A Kauffman

Institute for Biocomplexity and Informatics, University of Calgary, 2500 University Drive NW, Calgary Alberta, T2N 1N4, Canada

Received 8 September 2006, in final form 2 October 2006
Published 1 November 2006
Online at stacks.iop.org/JPhysA/39/L647

Abstract
We present a simple method for 'reverse engineering' causal networks, based on mutual information, as a correlation measure. The goal of our method is not to recover all the causal interactions in a network but rather to recover some causal interactions with a very high confidence. For this purpose, we derive an 'exact' theoretical result for the statistical significance of mutual information. Also, we give some numerical simulation results, obtained for random Boolean networks, as an idealized model of genetic regulatory networks.

PACS numbers: 89.75.−k, 02.50.Tt, 87.16.Yc

(Some figures in this article are in colour only in the electronic version)

The development of microarray technology gives the possibility of studying the gene expression of a large number of genes from a variety of organisms [1–3]. The analysis of microarray data by clustering methods, where patterns of gene expressions are grouped into clusters, has become one of the most important approaches [4]. A step beyond clustering is to reconstruct gene regulation networks, i.e. the topology, the causal interactions and the functional correlation of the genes. There are a number of approaches to 'reverse engineering' genetic regulatory networks from gene expression data [5–19]. All available approaches suffer to various degrees from problems such as an overfitting and underconstrained regression analysis, high computational complexity or reliance on non-realistic models. None of the existing computational methods for 'reverse engineering' (including ours) is likely to allow the inference of the full genetic network and its logic.

Here we introduce a simple method for 'reverse engineering', based on mutual information [20], as a causal correlation measure. The goal of our method is not to recover all the interactions in a network but rather to recover some interactions with high confidence. For this purpose, we derive an exact result for the statistical significance of mutual information. Most of the available methods are constructed to infer non-directed connections among genes, from temporally static data such as gene expression patterns across tissues, cell types, or cancer samples. The connection inference is acausal, and these methods cannot discriminate whether a pair of connected genes, A and B, has A as a regulatory input to B, or B to A. Our approach tries to overcome this difficulty by using dynamical data, and it requires the knowledge of gene expression levels at successive time points, before and after a network transition. Also, our method is general enough such that it can be applied to almost any kind of networks (chemical, biological, ecological, social, economic etc).

We consider that a network is a directed graph together with a function that assigns a label to each edge. We denote a network by $\Gamma = (V, E, F)$, where E stands for the directed set of edges connecting the set of vertices V, while F corresponds to the labelling function. Therefore, we assume that the network Γ has N labelled nodes $V = \{i | i = 1, \ldots, N\}$. Also, by $[c_{ij}]$ we denote the adjacency matrix of the network Γ:

$$c_{ij} = \begin{cases} 1 & \text{if} \quad i \text{ is connected to } j \\ 0 & \text{otherwise} \end{cases} . \tag{1}$$

In general, the number of 1's in the ith row corresponds to the number of edges entering the vertex i, and the number of 1's in the jth column corresponds to the number of edges leaving the vertex j.

We assume that the variables x_i, describing the state of the elements (vertices) of the network, take values in the alphabet (symbolic set):

$$A = \{\alpha | \alpha = 0, 1, \ldots, \Omega\}. \tag{2}$$

Also, the state of each node depends on the information received from its input nodes:

$$x_i = f_i\left(x_{i_1}, \ldots, x_{i_{k_i}}\right) \in A. \tag{3}$$

Here, $x_{i_1}, \ldots, x_{i_{k_i}}$ are the states of the input nodes of the node i; k_i is the connectivity of the node i (the number of edges entering the vertex i); f_i is the unknown transfer function corresponding to the node i.

The inference problem consists in finding the adjacency matrix $[c_{ij}]$ and the transfer functions f_i of a network $\Gamma = (V, E, F)$, given a labelling function F and a set of experimental data.

The state of the network is denoted by

$$X = \{x_i \in A | i = 1, \ldots, N\}. \tag{4}$$

By evolving the network one step forward, the state changes to

$$Y = \{y_i \in A | i = 1, \ldots, N\}. \tag{5}$$

We say that the pair

$$(X, Y) \equiv \{(x_i, y_i) \in A \times A | i = 1, \ldots, N\} \tag{6}$$

describes a causal transition of the network (Y is the output for the input X). We assume that the following set of causal state transitions:

$$\left\{\left(x_i^m, y_i^m\right) \in A \times A | i = 1, \ldots, N; m = 1, \ldots, M\right\} \tag{7}$$

is obtained via some experiments. For example, in the case of gene network, a causal pair may correspond to two successive measurements of the genome of the same biological organism etc.

We define the causal mutual information between the nodes i and j as following:

$$I_{ij} = \sum_{\alpha=0}^{\Omega} \sum_{\beta=0}^{\Omega} p_{\alpha\beta}\left(x_i^m, y_j^m\right) \log_2 \left(\frac{p_{\alpha\beta}\left(x_i^m, y_j^m\right)}{p_\alpha\left(x_i^m\right) p_\beta\left(y_j^m\right)} \right). \tag{8}$$

Here, $p_\alpha\left(x_i^m\right)$ and $p_\beta\left(y_j^m\right)$ are the probabilities of the symbols $\alpha \in A$ and $\beta \in A$, in $\left\{x_i^m \middle| m = 1, \ldots, M\right\}$ and respectively $\left\{y_j^m \middle| m = 1, \ldots, M\right\}$, and $p_{\alpha\beta}\left(x_i^m, y_j^m\right)$ is the probability of the pair (α, β) in $\left\{\left(x_i^m, y_j^m\right) \middle| m = 1, \ldots, M\right\}$. The above probabilities can be easily estimated as following:

$$p_{\alpha\beta}\left(x_i^m, y_j^m\right) = \frac{\sum_{m=1}^{M} \delta\left(x_i^m; \alpha\right)\delta\left(y_j^m; \beta\right)}{\sum_{\alpha=0}^{\Omega}\sum_{\beta=0}^{\Omega}\sum_{m=1}^{M} \delta\left(x_i^m; \alpha\right)\delta\left(y_j^m; \beta\right)}, \tag{9}$$

$$p_{\alpha}\left(x_i^m\right) = \sum_{\beta=0}^{\Omega} p_{\alpha\beta}\left(x_i^m, y_j^m\right), \tag{10}$$

$$p_{\beta}\left(y_j^m\right) = \sum_{\alpha=0}^{\Omega} p_{\alpha\beta}\left(x_i^m, y_j^m\right), \tag{11}$$

where

$$\delta(x; y) = \begin{cases} 1 & \text{if} \quad x = y \\ 0 & \text{if} \quad x \neq y \end{cases}. \tag{12}$$

The causal mutual information I_{ij} measures the causal correlation between the node i (input) and the node j (output). Therefore, I_{ij} measures the strength of a possible oriented link (edge): $i \rightarrow j$.

The algorithm has the following steps:

1. Calculate the causal mutual information I_{ij} for all pairs $i, j = 1, \ldots, N$.
2. Calculate the statistical threshold for mutual information: I^*.
3. Calculate the approximation of the adjacency matrix:

$$c'_{ij} = \begin{cases} 1 & \text{if} \quad I_{ij} \geqslant I^* \\ 0 & \text{if} \quad I_{ij} < I^*, \end{cases} \qquad i, j = 1, \ldots, N. \tag{13}$$

4. Create the tables for partial transfer functions. After establishing the adjacency matrix $[c'_{ij}]$ (which contains the inputs for each node), for each node one can partially infer the transfer functions by building the tables with the values of the input nodes and the values of the output node. These values can be taken directly from the set of causal transitions.

Now, let us derive an analytical result for the value of statistical threshold I^*. Let us consider two strings of length M, defined over the alphabet A:

$$x \equiv \{x^1, \ldots, x^M\} \in A^M, \tag{14}$$

$$y \equiv \{y^1, \ldots, y^M\} \in A^M, \tag{15}$$

we would like to calculate the relationship between the mutual information $I(x, y)$ and the probability $P(x, y)$ that the strings are arranged such that

$$\xi_\alpha = \sum_{m=1}^{M} \delta(x^m; \alpha), \qquad \alpha \in A, \tag{16}$$

$$\eta_\beta = \sum_{m=1}^{M} \delta(y^m; \beta), \qquad \beta \in A, \tag{17}$$

$$\gamma_{\alpha\beta} = \sum_{m=1}^{M} \delta(x^m; \alpha)\delta(y^m; \beta), \qquad \alpha, \beta \in A. \tag{18}$$

Let us consider the 'new' string:

$$z = (x, y) \equiv \{(x^1, y^1), \dots, (x^M, y^M)\} \in (A \times A)^M. \tag{19}$$

The number of strings $x \in A^M$, $y \in A^M$ and $z \in (A \times A)^M$, satisfying the above conditions are given by the multinomial coefficients:

$$(\xi_\alpha | \alpha \in A)! = \frac{\left(\sum_{\alpha=0}^{\Omega} \xi_\alpha\right)!}{\prod_{\alpha=0}^{\Omega} \xi_\alpha!} = \frac{M!}{\prod_{\alpha=0}^{\Omega} \xi_\alpha!}, \tag{20}$$

$$(\eta_\beta | \beta \in A)! = \frac{\left(\sum_{\beta=0}^{\Omega} \eta_\beta\right)!}{\prod_{\beta=0}^{\Omega} \eta_\beta!} = \frac{M!}{\prod_{\beta=0}^{\Omega} \eta_\beta!}, \tag{21}$$

$$(\gamma_{\alpha\beta} | \alpha, \beta \in A)! = \frac{\left(\sum_{\alpha=0}^{\Omega} \sum_{\beta=0}^{\Omega} \gamma_{\alpha\beta}\right)!}{\prod_{\alpha=0}^{\Omega} \prod_{\beta=0}^{\Omega} \gamma_{\alpha\beta}!} = \frac{M!}{\prod_{\alpha=0}^{\Omega} \prod_{\beta=0}^{\Omega} \gamma_{\alpha\beta}!}. \tag{22}$$

Therefore, the probability $P(x, y)$ is given by

$$P(x, y) = \frac{(\gamma_{\alpha\beta} | \alpha, \beta \in A)!}{(\xi_\alpha | \alpha \in A)!(\eta_\beta | \beta \in A)!} = \frac{\prod_{\alpha=0}^{\Omega} \xi_\alpha! \prod_{\beta=0}^{\Omega} \eta_\beta!}{M! \prod_{\alpha=0}^{\Omega} \prod_{\beta=0}^{\Omega} \gamma_{\alpha\beta}!}. \tag{23}$$

Now, taking the natural logarithm of $P(x, y)$ we obtain

$$\ln P(x, y) = \sum_{\alpha=0}^{\Omega} \ln \xi_\alpha! + \sum_{\beta=0}^{\Omega} \ln \eta_\beta! - \sum_{\alpha=0}^{\Omega} \sum_{\beta=0}^{\Omega} \ln \gamma_{\alpha\beta}! - \ln M!. \tag{24}$$

The logarithm of factorials can be calculated using Stirling's approximation:

$$\ln n! = \ln \prod_{k=1}^{n} k = \sum_{k=1}^{n} \ln k \approx \int_1^n \ln x \, \mathrm{d}x \approx n \ln n - n \tag{25}$$

Consequently, we have

$$\begin{aligned} \ln P(x, y) &= \sum_{\alpha=0}^{\Omega} \xi_\alpha \ln \xi_\alpha - \sum_{\alpha=0}^{\Omega} \xi_\alpha + \sum_{\beta=0}^{\Omega} \eta_\beta \ln \eta_\beta - \sum_{\beta=0}^{\Omega} \eta_\beta \\ &\quad - \sum_{\alpha=0}^{\Omega} \sum_{\beta=0}^{\Omega} \gamma_{\alpha\beta} \ln \gamma_{\alpha\beta} + \sum_{\alpha=0}^{\Omega} \sum_{\beta=0}^{\Omega} \gamma_{\alpha\beta} - M \ln M + M \\ &= \sum_{\alpha=0}^{\Omega} \xi_\alpha \ln \xi_\alpha + \sum_{\beta=0}^{\Omega} \eta_\beta \ln \eta_\beta - \sum_{\alpha=0}^{\Omega} \sum_{\beta=0}^{\Omega} \gamma_{\alpha\beta} \ln \gamma_{\alpha\beta} - M \ln M. \end{aligned} \tag{26}$$

Now, by introducing the probabilities $p_{\xi_\alpha} = \xi_\alpha / M$, $p_{\eta_\beta} = \eta_\beta / M$, $p_{\gamma_{\alpha\beta}} = \gamma_{\alpha\beta} / M$, we obtain

$$\begin{aligned} \frac{1}{M} \ln P(x, y) &= \sum_{\alpha=0}^{\Omega} p_{\xi_\alpha} \ln \left(M p_{\xi_\alpha}\right) + \sum_{\beta=0}^{\Omega} p_{\eta_\beta} \ln \left(M p_{\eta_\beta}\right) - \sum_{\alpha=0}^{\Omega} \sum_{\beta=0}^{\Omega} p_{\gamma_{\alpha\beta}} \ln \left(M p_{\gamma_{\alpha\beta}}\right) - \ln M \\ &= \sum_{\alpha=0}^{\Omega} p_{\xi_\alpha} \ln \left(p_{\xi_\alpha}\right) + \ln(M) \sum_{\alpha=0}^{\Omega} p_{\xi_\alpha} + \sum_{\beta=0}^{\Omega} p_{\eta_\beta} \ln \left(p_{\eta_\beta}\right) + \ln(M) \sum_{\beta=0}^{\Omega} p_{\eta_\beta} \\ &\quad - \sum_{\alpha=0}^{\Omega} \sum_{\beta=0}^{\Omega} p_{\gamma_{\alpha\beta}} \ln \left(p_{\gamma_{\alpha\beta}}\right) - \ln(M) \sum_{\alpha=0}^{\Omega} \sum_{\beta=0}^{\Omega} p_{\gamma_{\alpha\beta}} - \ln M \end{aligned}$$

$$= \sum_{\alpha=0}^{\Omega} p_{\xi_\alpha} \ln\left(p_{\xi_\alpha}\right) + \sum_{\beta=0}^{\Omega} p_{\eta_\beta} \ln\left(p_{\eta_\beta}\right) - \sum_{\alpha=0}^{\Omega}\sum_{\beta=0}^{\Omega} p_{\gamma_{\alpha\beta}} \ln\left(p_{\gamma_{\alpha\beta}}\right). \tag{27}$$

Switching to base 2 logarithm we have

$$\frac{1}{M} \log_2 P(x, y) = -H(x) - H(y) + H(x, y), \tag{28}$$

where $H(x)$, $H(y)$ and $H(x, y)$ are the entropies of x, y and $z = (x, y)$. Taking into account that

$$I(x, y) = H(x) + H(y) - H(x, y), \tag{29}$$

from the above equation we obtain

$$I(x, y) = \frac{1}{M} \log_2 \frac{1}{P(x, y)}. \tag{30}$$

The two strings x and y are correlated if $I(x, y) \geqslant I^*$, where the threshold is given by

$$I^* = \frac{1}{M} \log_2 \frac{1}{P^*}. \tag{31}$$

Here, P^* is some small value, which for a network with N nodes can be set as $P^* = N^{-2}$. This is also the probability that only 1 among the N^2 possible connections is a false positive. Therefore, this value gives a very high precision $\sim(1 - P^*)$ in deciding if there is a correlation between two elements of the network.

In order to evaluate the limitations of the proposed inference method we use the random Boolean networks (RBNs), as an idealized model of genetic regulatory networks [21]. Also, the RBN model can be considered as a prototype of a generic dynamical system, as they present chaotic as well as regular behaviour and many other typical structures of dynamical systems. From this point of view, the RBNs model is of potential interest in several different fields, ranging from gene regulation and control, to modelling biochemical, neurophysiological, social and physical spin systems. The model consists of N binary variables, corresponding to the two states of gene expression (off and on). In this binary setting, each gene is assigned a logical function on its inputs showing its next activity. While clearly an idealization, much has been learned from this class of large Boolean networks, and major features generalize to a class of piecewise linear differential equations [22, 23], a family of polynomial maps [24] and a set of nonlinear differential equations [25].

In the case of RBNs, the alphabet has only two symbols $A = \{\alpha|\alpha = 0, 1\}$. Each variable $x_i \in A, i = 1, \ldots, N$ has associated other k variables that act as arguments for a Boolean function f_i, that is used to update the state at each time step. Therefore, the variable x_i will change its state at each time step according to the rule:

$$x_i(t + 1) = f_i(x_{i_1}(t), \ldots, x_{i_k}(t)). \tag{32}$$

Both the function f_i and its inputs are initially randomly assigned to each variable, but maintained afterwards through the evolution of the system. The Boolean functions f_i are created as following:

$$f_i(x_{i_1}(t), \ldots, x_{i_K}(t)) = \begin{cases} 1 & \text{with probability } p \\ 0 & \text{with probability } 1 - p \end{cases}. \tag{33}$$

The parameter p is called the bias of the network. The inputs are randomly assigned such that for a given k and N, the total number of connections in the network is kN. Thus, k represents the mean connectivity of the network.

It is well known that, for a given value of the bias p, RBNs show a transition between ordered and chaotic phases, separated by a critical value k_c of the connectivity [25]. In the ordered phase ($k < k_c$) the networks freeze in a pattern after a short transient, while in the chaotic phase ($k > k_c$) all initial patterns are lost and the systems appear to be in a completely chaotic state. At the critical point ($k = k_c$) a small number of attractors is observed, which show high stability against minimal perturbations in single elements or Boolean functions and low reachability among different attractors. As we change the value of p the critical value at which the transition takes place also changes as $k_c = [2p(1-p)]^{-1}$.

In our simulations we have used a fixed bias $p = 0.5$ and we have varied k in the range $k \in [0, 8]$. Therefore, the critical point is at $k_c = 2$. For every k we have averaged the results over 100 networks. The causal state transition pairs have been generated by starting the network in a random state and evolving it one step forward in time. The number of state transition pairs M is also a varying parameter taking the following values $M = 250, 500, 750, 10^3$ and 10^4. The number of elements in the network is fixed at $N = 10^4$.

The precision Φ and the recall Ψ of the method are estimated as following:

$$\Phi = \frac{TP}{TP + FP} \in [0, 1], \tag{34}$$

$$\Psi = \frac{TP}{TP + FN} \in [0, 1], \tag{35}$$

where

$$TP = \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(c_{ij}; 1)\delta(c'_{ij}; 1) \tag{36}$$

is the number of true positives,

$$FP = \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(c_{ij}; 0)\delta(c'_{ij}; 1) \tag{37}$$

is the number of false positives, and

$$FN = \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(c_{ij}; 1)\delta(c'_{ij}; 0) \tag{38}$$

is the number of false negatives in the approximation c'_{ij} of the real adjacency matrix c_{ij}.

In figure 1 we give the precision $\Phi(k)$ and the recall $\Psi(k)$ obtained for different values of M. One can see that the precision stays at $\Phi(k) = 1$ for any values of k and M. This means that the method does not generate false positive connections. The recall reaches its maximum for $k \in (1, 2)$ after which it drops monotonously. By increasing the number of state transition pairs (M) the performance of the method increases, reaching a saturation value. For example, by increasing M from 10^3 to 10^4 one gains only about 2–3% in recall.

For a large interval $k \in (0, 7)$, the method is able to recall between 20% and 65% of the network, which is not too bad considering the simplicity of the approach. Assuming that the average connectivity of a real genetic network with $N \sim 10^4$ genes is somewhere in this range we conclude that by using this simple method one could recover a good per cent of it just by using about $M \sim 10^2$–10^3 state transitions. Of course, this depends also on the quality of the experimental data (i.e., the effect of noise).

Let us now analyse the effect of noise on the precision and recall of the proposed method. It is obvious that the precision of the method should not suffer, because the threshold (31) is independent of the level of noise. So, by introducing noise we are only expecting to see a

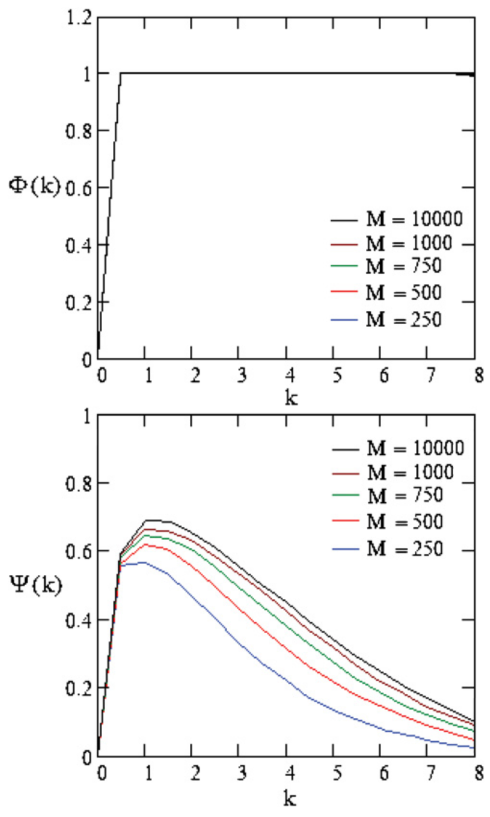

Figure 1. The precision and the recall of the algorithm applied to RBNs.

decrease in recall. In order to measure quantitatively the effect of noise we consider that the Boolean functions (32) are 'misbehaving' with a probability ε. This means that the update in the network is made according to the following rule:

$$x_i(t+1) = \begin{cases} f_i(x_{i_1}(t), \ldots, x_{i_k}(t)) & \text{if} \quad \text{rnd}() \geqslant \varepsilon \\ [f_i(x_{i_1}(t), \ldots, x_{i_k}(t)) + 1] \text{ modulo}(2) & \text{if} \quad \text{rnd}() < \varepsilon, \end{cases} \tag{39}$$

where rnd() $\in [0, 1]$ returns an uniform random number. In figure 2 we give the precision $\Phi(k)$ and the recall $\Psi(k)$ obtained for different values of $\varepsilon \in \{0.0, 0.1, 0.2, 0.3\}$, $N = 10^3$ and $M = 500$. One can see that the precision stays at $\Phi(k) = 1$ for any values of ε and k. However, the recall drops proportionally with the noise level ε.

In conclusion, we have presented a simple method for 'reverse engineering' causal networks. There are a number of characteristics that make our method significantly different from other approaches. For example, Liang *et al* [5] and Akutsu *et al* [6] have calculated a lower bound and an upper bound of the number of expression patterns required to identify the genetic network using the Boolean network model. However, their results require perfect data and therefore, is not robust with respect to noise. The Bayesian network approach [10, 13] uses heuristic procedures to search for locally optimal graph structures. Learning the most likely network requires the exploration of the entire graph space, which is an NP-complete problem. The relevance networks method, proposed by Butte *et al* [8, 9], and the ARACNE algorithm developed by the Califano group [18, 19], compute mutual information for all gene pairs and infer that two genes are biologically related if their mutual information is above

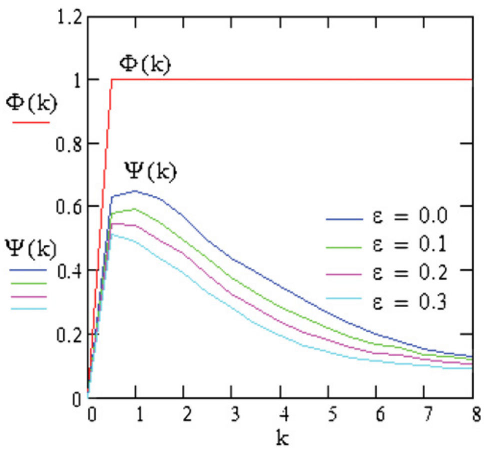

Figure 2. The precision $\Phi(k)$ and the recall $\Psi(k)$ as a function of the noise level $\varepsilon \in \{0.0, 0.1, 0.2, 0.3\}$.

a certain threshold. This method tries to overcome the hard computational problem raised by the Bayesian Network approach. However, the threshold is calculated only by using some semi-empirical methods. There is no proof that by setting the threshold according to these procedures, the connections will be inferred with a high precision. A high precision of $\Phi(k) \sim 1$ is a must for any inference method. For example, let us assume that $\sim 10^8$ connections are inferred with a precision of only $\Phi(k) \sim 0.95$. This is translated in a huge number of false positives $FP \sim 10^6$ which is quite difficult and expensive to be tested in the lab.

Our method builds upon the algorithms described in [8, 9, 18, 19] and adds several major improvements. These algorithms were constructed to infer only non-directed connections among genes, from temporally static data such as gene expression patterns across tissues, cell types, or cancer samples. Our approach tries to overcome this difficulty by using dynamical data, and it requires the knowledge of gene expression levels at successive time points, before and after a network transition. Also, the goal of our method is not to recover all the causal interactions in a network but rather to recover some causal interactions with high confidence. For this purpose, we have derived an analytical result for the threshold of mutual information (31). This result gives a generalization of the methods used in previous approaches. The derived threshold gives a high precision of $\sim(1 - P^*)$, where P^* is an arbitrary small number. Within its scope, our approach overcomes several limitations of previous methods. It has a low computational complexity and it does not rely on unrealistic network models or a priori assumptions. Also, the method can be applied to arbitrarily complex networks, without reliance on heuristic search procedures.

Acknowledgment

This research was supported by iCORE under Grant No. RT732223.

References

[1] DeRisi J L, Lyer V R and Brown P O 1997 *Science* **277** 1275
[2] Wen X, Fuhrman S, Michaels G S, Carr D B, Smith S, Barker J L and Somogyi R 1998 *Proc. Natl. Acad. Sci. USA* **95** 334

[3] Yuh C-H, Bolouri H and Davidson E H 1998 *Science* **279** 1896
[4] Eisen M B, Spellman P T, Brown P O and Botstein D 1998 *Proc. Natl. Acad. Sci. USA* **95** 14863
[5] Liang S, Fuhrman S and Somogyi R 1998 *Pac. Symp. Biocomput.* **1998** 18
[6] Akutsu T, Miyano S and Kuhara S 1999 *Pac. Symp. Biocomput.* **1999** 17
[7] Friedman N, Linial M, Nachman I and Pe'er D 2000 *J. Comput. Biol.* **7** 601
[8] Butte A J and Kohane I S 2000 *Pac. Symp. Biocomput.* **2000** 418
[9] Butte A J, Tamayo P, Slonim D, Golub T R and Kohane I S 2000 *Proc. Natl. Acad. Sci. USA* **97** 12182
[10] Hartemink A J, Gifford D K, Jaakkola T S and Young R A 2001 *Pac. Symp. Biocomput.* **2001** 422
[11] van Someren E P, Wessels L F, Backer E and Reinders M J 2002 *Pharmacogenomics* **3** 507
[12] Yeung M K, Tegner J and Collins J J 2002 *Proc. Natl. Acad. Sci. USA* **99** 6163–8
[13] Smith V A, Jarvis E D and Hartemink A J 2002 *Bioinformatics* **18** S216
[14] Lukashin A V, Lukashev M E and Fuchs R 2003 *Bioinformatics* **19** 1909
[15] Gat-Viks I and Shamir R 2003 *Bioinformatics* **19** (Suppl. 1) i108
[16] Gardner T S, di Bernardo D, Lorenz D and Collins J J 2003 *Science* **301** 102
[17] Friedman N 2004 *Science* **303** 799
[18] Basso K, Margolin A A, Stolovitzky G, Klein U, Dalla-Favera R and Califano A 2005 *Nat. Genet.* **37** 382
[19] Margolin A A, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera R Dalla and Califano A 2006 *BMC Bioinformatics* **7** (Suppl. 1) S7
[20] Cover T M and Thomas J A 1991 *Elements of Information Theory* (New York: Wiley)
[21] Kauffman S A 1969 *J. Theor. Biol.* **22** 437
Aldana M, Coopersmith S and Kadanoff L P 2003 *Perspectives and Problems in Nonlinear Science (Springer Applied Mathematical Sciences Series)* ed E Kaplan, J E Marsden and K R Sreenivasan (New York: Springer) pp 23–89
[22] Glass L and Kauffman S A 1973 *J. Theor. Biol.* **39** 103
[23] Glass L 1975 *J. Chem. Phys.* **63** 1325
[24] Andrecut M 2005 Mean field dynamics of random Boolean networks *J. Stat. Mech.* P02003
[25] Andrecut M and Kauffman S A 2006 *New J. Phys.* **8** 148
[26] Derrida B and Pomeau Y 1986 *Europhys. Lett.* **1** 45