BIOINFORMATICS ORIGINAL PAPER
Vol. 27 no. 11 2011, pages 1529–1536
doi:10.1093/bioinformatics/btr166

Systems biology
Advance Access publication April 5, 2011

Inferring Boolean network structure via correlation

Markus Maucher[1,†], Barbara Kracher[2,†], Michael Kühl[2] and Hans A. Kestler[1,3,*]

[1]Research group Bioinformatics and Systems Biology, Clinic for Internal Medicine I, University Medical Center Ulm, [2]Institute for Biochemistry and Molecular Biology, Ulm University, and [3]Research Group Bioinformatics and Systems Biology, Institute for Neural Information Processing, Ulm University, Ulm, Germany

Associate Editor: Olga Troyanskaya

ABSTRACT

Motivation: Accurate, context-specific regulation of gene expression is essential for all organisms. Accordingly, it is very important to understand the complex relations within cellular gene regulatory networks. A tool to describe and analyze the behavior of such networks are Boolean models. The reconstruction of a Boolean network from biological data requires identification of dependencies within the network. This task becomes increasingly computationally demanding with large amounts of data created by recent high-throughput technologies. Thus, we developed a method that is especially suited for network structure reconstruction from large-scale data. In our approach, we took advantage of the fact that a specific transcription factor often will consistently either activate or inhibit a specific target gene, and this kind of regulatory behavior can be modeled using monotone functions.

Results: To detect regulatory dependencies in a network, we examined how the expression of different genes correlates to successive network states. For this purpose, we used Pearson correlation as an elementary correlation measure. Given a Boolean network containing only monotone Boolean functions, we prove that the correlation of successive states can identify the dependencies in the network. This method not only finds dependencies in randomly created artificial networks to very high percentage, but also reconstructed large fractions of both a published *Escherichia coli* regulatory network from simulated data and a yeast cell cycle network from real microarray data.

Contact: hans.kestler@uni-ulm.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on August 20, 2010; revised on February 28, 2011; accepted on March 25, 2011

1 INTRODUCTION

Boolean networks were popularized by Stuart Kauffman as models for genetic regulatory networks (Kauffman, 1969). In this kind of model, only two states are discriminated (active/inactive) for each gene. The dynamics of the network can then be described by Boolean functions. This model works with a very small set of parameters and thus represents a very stringent application of Occam's Razor, which makes it especially suitable for modeling large genetic networks (Bornholdt, 2005). Nonetheless, it is powerful enough to model the structure of network motifs, basic network components frequently found in gene regulatory networks (Alon, 2006; Babu *et al.*, 2004).

Gene transcription in eukaryotic cells has to be tightly regulated to ensure proper cell function, i.e. according to a particular cellular context (cell cycle phase, cell type, environmental conditions, developmental stage), a specific subset of genes is expressed while the expression of other genes has to be actively repressed. This regulation is generally mediated via certain proteins, called transcription factors, that bind to specific sequence motifs in the promoter region of a gene and either enhance or inhibit the transcription of this gene [reviewed e.g. in (Orphanides and Reinberg, 2002; Venters and Pugh, 2009)]. If a promoter region contains binding motifs for different transcription factors, these factors can either cooperate in the regulation of a certain target gene or counteract each other. Moreover, in some cases, specific cofactors determine whether a transcription factor acts as activator or repressor. Despite the variety in the modes of transcriptional regulation, most transcriptional regulators will be either activators or inhibitors of a certain gene in a specific cell type. In this case, the activating or repressing effect of a transcription factor monotonically depends on its cellular concentration. In other words, an increase in the concentration of an activator will increase but never decrease transcription of its target, while an increase in the concentration of a repressor will decrease but never increase transcription of its target. This kind of transcriptional regulation can be modeled mathematically in a very simplistic manner by the use of monotone Boolean functions which describe exactly this monotonic relation. Besides the monotone Boolean functions applied in this work, there exist further related classes of functions describing such monotonic relations. Among them are, for example, nested canalyzing functions (Kauffman *et al.*, 2003), single-layer perceptrons (Rani *et al.*, 2007) and multilinear functions (Tsukimoto and Hatano, 2003).

Gene expression data can be analyzed and visualized on the basis of correlations identified in the observed expression patterns of the analyzed genes. Models from information theory correlate expression values from two genes directly and predict an interaction between two genes to be present if the correlation coefficient is exceeding a certain threshold. Algorithms based on this principle are for example CLR (Faith *et al.*, 2007) or ARACNE (Margolin *et al.*, 2006). However, these correlations generally reflect co-expression of genes and can, under some restrictions to the underlying network like e.g. absence of cyclic dependencies, also give information on direct causal relationships [cf. (Pearl, 2000; Shipley, 2000; Spirtes *et al.*, 2000) and references therein]. Moreover, Opgen-Rhein and Strimmer (2007) and Zoppoli *et al.* (2010) established algorithms

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

to develop networks depicting dynamic relations from correlation data. Likewise, there exist algorithms for the inference of dynamic Boolean networks from gene expression data, like REVEAL (Liang *et al.*, 1998) or the best fit extension algorithm (Lähdesmäki *et al.*, 2003). These algorithms, however, perform reasonably well only if the datasets are not too large.

In contrast to existing approaches, we do not correlate expression data of different genes directly, but instead correlate the states of particular genes with the successive states of potential target genes, thus assuming and examining directed causal dependencies.

This article is organized as follows: we first prove properties of interaction reconstruction via correlation, assuming monotone Boolean functions. This is followed by a series of experiments successively advancing from entirely artificial through to real data.

2 METHODS—CORRELATION AND MONOTONE FUNCTIONS

Boolean networks: a Boolean network consists of n nodes, numbered from 1 to n and n functions $f_1,\dots,f_n:\{0,1\}^n\rightarrow\{0,1\}$. The state of the network can be described by a Boolean vector $x\in\{0,1\}^n$, where x_i describes the state of the i-th node. The n functions describe the dynamics of the network: If the network is in state x at time t, it transforms into state $(f_1(x),f_2(x),\dots,f_n(x))$ at time $t+1$. We can also combine $f_1,\dots,f_n$ into one function $F:\{0,1\}^n\rightarrow\{0,1\}^n$ such that $F(x)_i=f_i(x)$. State x then transforms into state $F(x)$. When considering successive states $x^{(1)},x^{(2)},\dots$, the term $x_i^{(j)}$ will denote the state of the i-th node in $x^{(j)}$.

Relevance of a variable: a function $f:\{0,1\}^n\rightarrow\{0,1\}$ depends on the i-th variable if there exist $x_1,\dots,x_{i-1},x_{i+1},\dots,x_n$ such that

$$\begin{aligned}&f(x_1,\dots,x_{i-1},0,x_{i+1},\dots,x_n)\\&\qquad\neq f(x_1,\dots,x_{i-1},1,x_{i+1},\dots,x_n).\end{aligned}$$

If f depends on the i-th input variable, we also say that the i-th variable is *relevant* for f. The set of all relevant variables of f is denoted as rel(f).

Reconstructing the dynamics of a Boolean network: in order to reconstruct the functions of a Boolean network, we are given a sequence of m states $x^{(1)},x^{(2)},\dots,x^{(m)}$ along with the corresponding successor states $F(x^{(1)}),F(x^{(2)}),\dots,F(x^{(m)})$. A tuple $(x^{(i)},F(x^{(i)}))$ is also called an *example*. From these examples, the task is to reconstruct the dependencies in the Boolean network, i.e. the set of variables each of the functions depends on.

Monotonicity of functions: a Boolean function $f:\{0,1\}^n\rightarrow\{0,1\}$ is monotonically increasing in the i-th variable if for all $x_1,\dots,x_{i-1},x_{i+1},\dots,x_n$

$$\begin{aligned}&f(x_1,\dots,x_{i-1},0,x_{i+1},\dots,x_n)\\&\qquad\leq f(x_1,\dots,x_{i-1},1,x_{i+1},\dots,x_n),\end{aligned}$$

the function f is monotonically decreasing in the i-th variable if for all variables $x_1,\dots,x_{i-1},x_{i+1},\dots,x_n$

$$\begin{aligned}&f(x_1,\dots,x_{i-1},0,x_{i+1},\dots,x_n)\\&\qquad\geq f(x_1,\dots,x_{i-1},1,x_{i+1},\dots,x_n).\end{aligned}$$

A function f is monotone if for every variable f is either monotonically increasing or monotonically decreasing in that variable. For example, the Boolean *AND* function is monotone, while the *XOR* function is not.

Influence of a variable: a probability distribution D on $\{0,1\}^n$ is called product distribution if for any D-distributed random variable X the property $P[X=(x_1,\dots,x_n)]=\prod_{i=1}^{n}P[X_i=x_i]$ holds, i.e. the X_i are independent.

Let D be a product distribution and X be a D-distributed random variable. The *influence* $I_{D,i}(f)$ of a variable x_i on a function $f:\{0,1\}^n\rightarrow\{0,1\}$ is defined as the probability that a change in x_i will also lead to a change in $f(X)$, i.e.

$$I_{D,i}(f)=P_D\left[f(X)|_{x_i=0}\neq f(X)|_{x_i=1}\right] \ .$$

Here, $f(X)|_{x_i=0}$ denotes a partial assignment, i.e.

$$f(X_1,\dots,X_n)|_{x_i=b}=f(X_1,\dots,X_{i-1},b,X_{i+1},\dots,X_n) \ .$$

Pearson correlation: given two random variables X and Y, their Pearson correlation is defined as

$$\rho(X,Y)=\frac{E[(X-E[X])(Y-E[Y])]}{\sigma_X\sigma_Y}=\frac{\mathrm{Cov}(X,Y)}{\sigma_X\sigma_Y},$$

where σ_Z denotes the SD of a random variable Z.

THEOREM 1. *Given a monotone function $f:\{0,1\}^n\rightarrow\{0,1\}$ and a random variable $X\in\{0,1\}^n$ that is distributed according to a product distribution D. Then the function f depends on the i-th variable if and only if the Pearson correlation of X_i and $f(X)$ is non-zero for non-zero variances.*

PROOF. Let D_i denote the product distribution D on all variables except x_i. Then

$$\begin{aligned}&\frac{E_D[(X_i-E[X_i])(f(X)-E[f(X)])]}{\sigma_{X_i}\sigma_{f(X)}}\\
&=\frac{1}{\sigma_{X_i}\sigma_{f(X)}}E_{D_i}E_{X_i}[(X_i-E[X_i])(f(X)-E[f(X)])]\\
&=\frac{1}{\sigma_{X_i}\sigma_{f(X)}}E_{D_i}\Big[(1-\mu_i)(-\mu_i)\Big(f(X)|_{x_i=0}-\overline{f(X)}\Big)\\
&\qquad\qquad+\mu_i(1-\mu_i)\Big(f(X)|_{x_i=1}-\overline{f(X)}\Big)\Big]\\
&=\frac{\mu_i(1-\mu_i)}{\sigma_{X_i}\sigma_{f(X)}}E_{D_i}\left[f(X)|_{x_i=1}-f(X)|_{x_i=0}\right]\\
&=\frac{\sigma_{X_i}}{\sigma_{f(X)}}E_{D_i}\left[f(X)|_{x_i=1}-f(X)|_{x_i=0}\right] \ .\end{aligned}$$

The theorem then follows from the fact that f is monotone in x_i. ■

As derived in Theorem 1, the influence of a variable can be estimated via a modified version of the Pearson correlation

$$\tilde{\rho}(X,Y)=\frac{E[(X-E[X])(Y-E[Y])]}{\sigma_X^2}=\frac{\mathrm{Cov}(X,Y)}{\sigma_X^2}.$$

The Chernoff-Hoeffding bound (Hoeffding, 1963): given independent random variables $X_1,\dots,X_m$ with $a\leq X_i\leq b$ for all i, the Chernoff-Hoeffding bound can be stated as follows:

$$P\left[\left|\sum_{i=1}^{m}X_i-\sum_{i=1}^{m}E[X_i]\right|\geq\epsilon m\right]\leq e^{\frac{-2m\epsilon^2}{(b-a)^2}} \ .$$

Estimating the correlation coefficient: we will use the Chernoff-Hoeffding bound for the estimation of a variable's influence:

THEOREM 2. *When estimating the influence of a variable on a function, the estimation error shrinks exponentially in the number of samples.*

PROOF. As shown in Theorem 1, the influence of a variable X_i on a function f can be estimated via a modified correlation $\tilde{\rho}(X_i,f(X))=\frac{\sigma_{f(X)}}{\sigma_X}\rho(X_i,f(X))$. We will show that when estimating this term the error probability shrinks exponentially in the number of samples.
Via the Chernoff-Hoeffding bound, we can prove that

$$P[|\overline{x}-\mu_x|\geq\epsilon_x]\leq\exp(-2m\epsilon_x^2)$$

and

$$P\left[\left|\overline{y}-\mu_y\right|\geq\epsilon_y\right]\leq\exp(-2m\epsilon_y^2) \ ,$$

which implies

$$\begin{aligned} &P[|\overline{xy}-\mu_x\mu_y|\geq \epsilon_x\mu_y+\epsilon_y\mu_x+\epsilon_x\epsilon_y] \\ &\leq \exp(-2m\epsilon_x^2)+\exp(-2m\epsilon_y^2) \ , \end{aligned}$$

and we can also show

$$P\left[\left|\frac{1}{m}\sum_{i=1}^{m} x_i y_i - E[XY]\right| \geq \epsilon_r\right] \leq \exp(-2m\epsilon_r^2) \ .$$

We use the fact that the x_i and y_i are all Boolean to estimate the sample variance $s_x^2 = \bar{x}(1-\bar{x})$. Let δ_x denote the estimation error for μ_x, i.e. $\bar{x} = \mu_x + \delta_x$. That way,

$$\begin{aligned} \bar{x}(1-\bar{x}) &= (\mu_x+\delta_x)(1-\mu_x-\delta_x) \\ &= \mu_x(1-\mu_x)+(1-2\mu_x-\delta_x)\delta_x \ . \end{aligned}$$

Since $|(1-2\mu_x-\delta_x)| \leq 1$, we can conclude that

$$|\bar{x}(1-\bar{x})-\mu_x(1-\mu_x)| \leq |\delta_x|.$$

This means that if the error for estimating μ_x is at most ϵ_x then the error for estimating σ_x^2 is at most ϵ_x, too. We can now combine the estimation errors above: estimating the modified correlation coefficient $\tilde{\rho}$ via

$$\tilde{r}(X,Y) = \frac{\frac{1}{m}\sum_{i=1}^{m} x_i y_i - \overline{xy}}{\bar{x}(1-\bar{x})}$$

and setting $\epsilon := (\epsilon_r+\epsilon_x\mu_y+\epsilon_y\mu_x+\epsilon_x\epsilon_y+\epsilon_x)/(\sigma_x^2-\epsilon_x)$, we can use Lemma 1 below to conclude that

$$\begin{aligned} &P[|\tilde{r}(X,Y)-\tilde{\rho}(X,Y)|\geq \epsilon] \\ &\leq \exp(-2m\epsilon_x^2)+\exp(-2m\epsilon_y^2)+\exp(-2m\epsilon_r^2) \ . \end{aligned}$$

In particular, the error probability shrinks exponentially in the number of samples m. ■

LEMMA 1. *Let $A, \tilde{A}, B, \tilde{B} \in [-1,1]$ with $B > |A|$, $|A-\tilde{A}| < \epsilon_1$ and $|B-\tilde{B}| < \epsilon_2$. Then*

$$\left|\frac{A}{B}-\frac{\tilde{A}}{\tilde{B}}\right| \leq \frac{\epsilon_1+\epsilon_2}{B-\epsilon_2} \ .$$

PROOF. It holds that

$$\left|\frac{A}{B}-\frac{\tilde{A}}{\tilde{B}}\right| \leq \max\left\{\frac{A}{B}-\frac{A-\epsilon_1}{B+\epsilon_2}, \frac{A+\epsilon_1}{B-\epsilon_2}-\frac{A}{B}\right\} \ .$$

With

$$\frac{A}{B}-\frac{A-\epsilon_1}{B+\epsilon_2} = \frac{A\epsilon_2+B\epsilon_1}{B(B+\epsilon_2)} \leq \frac{\epsilon_1+\epsilon_2}{B+\epsilon_2}$$

and

$$\frac{A+\epsilon_1}{B-\epsilon_2}-\frac{A}{B} = \frac{A\epsilon_2+B\epsilon_1}{B(B-\epsilon_2)} \leq \frac{\epsilon_1+\epsilon_2}{B-\epsilon_2}$$

the lemma follows. ■

Inferring the structure of a Boolean network: combining the results above, we can infer the dependency relations within a Boolean graph with a fast algorithm: Algorithm 1 finds the relevant variables of a monotone function with a chosen minimum influence, given a sequence of independent examples, by calculating the correlation value of each variable and identifying all variables with this correlation exceeding a given value. Running that algorithm for the output function of each node of a Boolean network reveals the structure of the entire network.

Algorithm 1 Finding all relevant variables of f with influence exceeding a threshold T

```
Input: $m$ examples $(x^{(1)}, f(x^{(1)})), \ldots, (x^{(m)}, f(x^{(m)}))$ of a function $f$,
    drawn from a product distribution, threshold $T$
Output: Approximation of $rel(f)$
  $rel \leftarrow \emptyset$
  $\bar{y} \leftarrow \frac{1}{m}\sum_{j=1}^{m} f(x^{(j)})$
  for $i \in \{1, \ldots, n\}$ do
    $\bar{x} \leftarrow \frac{1}{m}\sum_{j=1}^{m} x_i^{(j)}$
    if $\bar{x}(1-\bar{x}) > 0$ then
      {ensures non zero sample variance}
      if $\frac{\frac{1}{m}\sum_{j=1}^{m} x_i^{(j)} f(x^{(j)}) - \overline{xy}}{\bar{x}(1-\bar{x})} \geq T$ then
        $rel \leftarrow rel \cup \{i\}$
      end if
    end if
  end for
  return rel
```

3 SIMULATION RESULTS

Error rates in artificial networks with fixed in-degree: to investigate the performance of our method for concrete values and for controlled noise levels, we analyzed the error probability of our inference algorithm in an artificial network consisting of monotone Boolean functions. For this purpose, we created a set of 50 random Boolean networks each containing 80 nodes. For every node in these networks, a monotone function with three input variables was constructed as follows: first, we created a random truth table and determined randomly for each input variable if the function should be monotonically increasing or decreasing in that variable. We then altered the initial truth table by correcting all inconsistencies. If this resulted in a function that did not depend on all of its input variables, we created a new random truth table and repeated the process until every function depended on all of its input variables. Additionally, for each of the networks generated we analyzed attractors, i.e. single states or sequences of states toward which the networks evolve. The network generation and attractor analysis was performed with the R package BoolNet (Müssel *et al.*, 2010), with the described modification to create monotone functions. We determined the number of attractors for each of these artificial networks via a random sampling of 10 000 starting states and their resulting attractors (function getAttractors of BoolNet). The number of attractors ranged from 1 to 59 (with a median of 5.5), with attractor lengths ranging from 1 to 258 (with median 6). From each of the 50 random networks, we subsequently generated a batch of 1000 simulated time-series datasets, with each dataset covering 2 time points. In this process, for the generation of each dataset a new starting configuration of the network was drawn from a uniform distribution. The datasets generated in this way for each of the 50 random networks represent synthetic 'gene expression' data of the 80 'genes', measured at two successive time points in an unperturbed system. These datasets were then used to reconstruct the dependencies in the underlying artificial networks. If a Boolean function depends on three variables, changing one of these variables from 0 to 1 must have an effect on f for at least one combination of the other two variables. Since the probability of such a combination of two variables is 0.25 under a uniform distribution, each of these variables has an influence of at least 0.25. A correlation value was

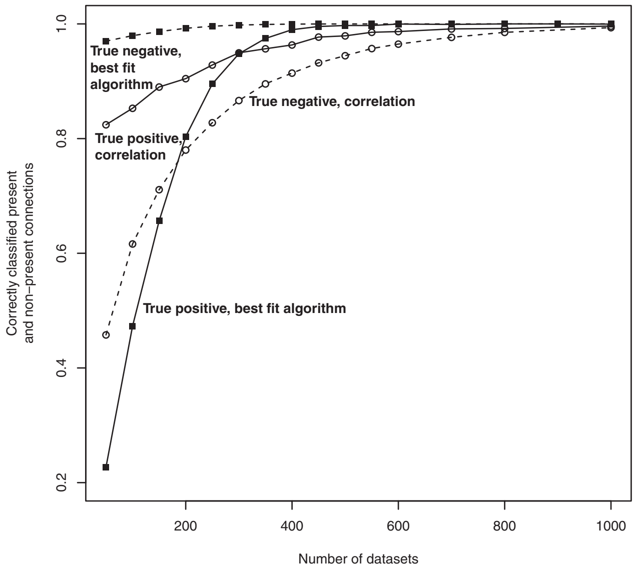

Fig. 1. True positive and true negative rates of an experiment with artificial data and Gaussian noise with SD $\sigma = 0.4$. The correlation algorithm and the best fit extension algorithm were used to infer artificially created networks consisting of 80 nodes with an in-degree of 3. The plot shows average true positive and true negative rates for the reconstruction of 50 artificial networks from 50 to 1000 datasets each.

assumed to signify a dependency when it exceeded a threshold of 0.125, i.e. half the minimum influence.

Figure 1 shows the average identification rates of our method and Lähdesmäki and co-workers' best fit extension algorithm (Lähdesmäki *et al.*, 2003) for the reconstruction of 50 artificial Boolean networks. The network size of 80 nodes was chosen to be able to compare the performance of our approach with the best fit algorithm. Moreover, to obtain data that is closer to experimental results, we added a Gaussian noise term with mean 0 and SD 0.4 to the generated data, then rounded to the nearest value in $\{0,1\}$. This corresponds to a probability of about 0.1 for each bit to be changed to the wrong value. The effect of different noise levels on network reconstruction is shown in Supplementary Figure S1. For the best fit algorithm, we used the implementation in the BoolNet R-package. As seen in Figure 1, for a threshold of 0.125, the correlation approach detected 80% and more of the dependencies in the artificial network already when less than 200 datasets were used for network reconstruction, while the best fit algorithm identified a comparable portion of dependencies only when more than 200 datasets were used. However, for this threshold, the correlation approach also retrieved a fraction of false-positive dependencies, that were not actually present in the underlying networks. For example, when using 200 datasets, 25% of the non-present dependencies were wrongly classified as present. Thus, the rate of correctly classified non-present connections was lower than the true-positive rate, unless more than 400 datasets were used for network inference (Fig. 1). We further analyzed precision, i.e. the fraction of true dependencies among all predicted dependencies, and recall, i.e. the fraction of true dependencies that was correctly predicted, for our reconstruction approach and the best fit algorithm. As seen in Supplementary Figure S5, precision and recall increase for both algorithms with the number of datasets used for network inference. In the correlation approach, moreover, the threshold for the correlation can be varied to determine whether the algorithm should preferably identify dependencies with a high precision, at the expense of a lower recall, or the other way round (see Supplementary Fig. S5). As additional measure for the reliability of network reconstruction, we further analyzed the F-score, which can be regarded as weighted average of precision and recall (see Supplementary Fig. S6).

Table 1. Runtimes of the correlation method and the best fit extension algorithm, averaged over 20 runs

Network size	Correlation (s)	Best Fit (max $k = 3$) (s)	Best fit (max $k = 4$)
50 nodes	0.98	0.79	10.2 s
100 nodes	3.86	11.2	324 s
200 nodes	15.4	176	2 h 50 min

The Correlation approach is independent of setting a max k.

Runtime: in a second experiment, we measured the running time of the correlation method and the best fit algorithm. The average single-core runtimes on a 3.0 GHz Xeon CPU are given in Table 1. The random networks for the reconstruction consisted of 50, 100 and 200 nodes, respectively. In each case, we measured the running times for reconstruction of the random network from 50 synthetic datasets. The values in the table show the running times averaged over 20 runs, where we created a new random Boolean network for each of these runs. As seen in Table 1, a key advantage of the correlation method are the considerably shorter running times compared to the best fit algorithm. Thus, our correlation approach is especially useful for large networks in which the nodes have a relatively large in-degree. For example, the runtime of the best fit algorithm for the reconstruction of a network of 200 nodes with a maximum in-degree of 4 (i.e. each Boolean function depends on no more than four input variables) is almost 3 h, whereas the correlation approach needs only about 15 s for this task.

Reconstruction of interactions from an E. coli regulatory network: we next tested the method on a real biological network. For that purpose, we chose the integrated *Escherichia coli* gene regulatory and metabolic network published by Covert *et al.* (2004). Based on previously published information, extracted from literature and databases, the authors of this study constructed a network model that describes the transcriptional regulation of genes involved in *E.coli* metabolism by transcription factors and environmental stimuli. The complete model includes a total of 1010 genes and 102 environmental stimuli. Of these, 906 genes code for proteins in the *E.coli* metabolic network and 104 genes code for transcriptional regulators. The expression of 479 of the genes is controlled in the metabolic network, 427 genes were not included in our investigation. So we used a total 685 $(= 479 + 104 + 102)$ variables in the network, i.e. the regulated 583 genes and the 102 additional inputs. To reconstruct the regulatory network, including the environmental stimuli, we used the Boolean functions provided by Covert *et al.* (2004) to generate 20 synthetic time-series datasets covering 3 time points each. For the generation of each of the 20 datasets, a random starting configuration of the network was drawn from a uniform distribution. To every value in all datasets, Gaussian noise with mean 0 and SD 0.1 was added, to resemble the noise generally

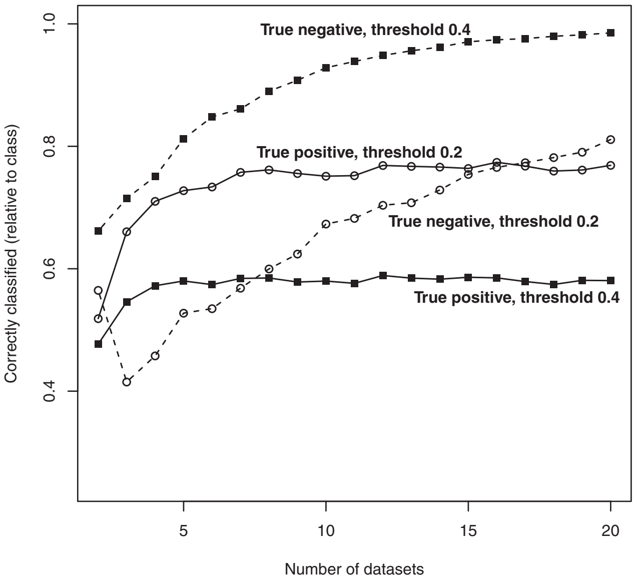

Fig. 2. True positive (solid lines) and true negative (dashed lines) rates of the correlation approach for reconstruction of a published *E.coli* regulatory network with 685 nodes. The network was reconstructed from 2 to 20 synthetic time-series datasets (with 3 time steps), that had been generated from the network by Covert *et al.* (2004). Correlations signified a regulatory dependency if they exceeded a predefined threshold of 0.2 (circles) or 0.4 (squares).

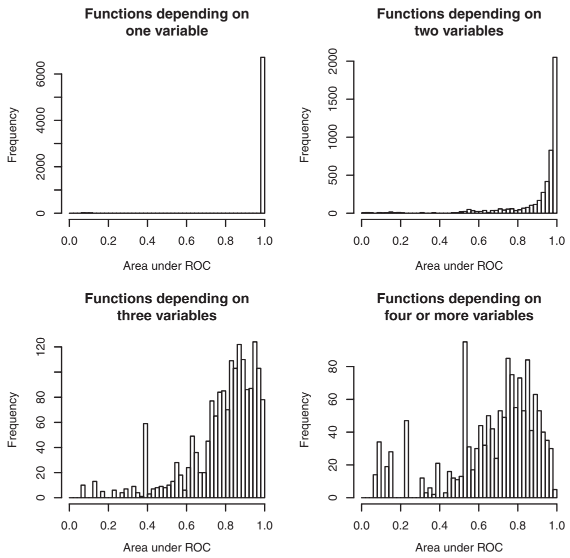

Fig. 3. Histograms for the areas under the receiver operator curves (AUC). Correlation values were computed from 10 time-series datasets each covering three discrete time steps. In total, 25 reconstruction runs were performed, each time creating new time-series. That way, every dependency in the *E.coli* network contributes 25 AUC values to the corresponding histogram. The AUC frequency distributions for functions that depend on 1, 2, 3, or more than 3 input variables are shown in four separate histograms.

observed in biological data. Binarization was then performed by rounding to the nearest Boolean value 0 or 1. We then reconstructed the dependencies of the published *E.coli* network from the synthetic datasets using the described correlation method. To investigate the influence of the number of datasets on the reconstruction, we varied the number of used datasets in the range of 2 to the available 20 (Fig. 2). By comparing the predicted dependencies with the published network, finally, the fraction of correctly classified present (true positive) and non-present (true negative) dependencies was determined. Additionally, we analyzed precision and recall and F-scores for different thresholds (see Supplementary Figs S7 and S8). As seen in Figure 2, for a threshold value of 0.2, the true positive rate was close to 75% for network reconstruction from at least five datasets. The true negative rate, for this threshold, ranged between 50% and 80% depending on the number of datasets used for reconstruction. Thus, at this threshold a large fraction (∼75%) of the actual dependencies was found. To increase precision of the correlation approach, a higher threshold can be applied. So, for a threshold of 0.4, the true positive rate still was close to 60%, and in this case the true negative rate was higher than 80% already when five datasets were used for reconstruction (Fig. 2). Examples of reconstructed interactions for different dataset sizes and thresholds are given as Supplementary Material II.

To further evaluate how reliably the correlation of successive states identified true relevant variables of the examined Boolean functions, we additionally computed the area under the receiver operator curve (AUC; Fawcett, 2006). The AUC is equal to the probability that a randomly chosen present dependency has a higher absolute correlation value than a randomly chosen non-existing dependency. For computation of the AUC values, we performed 25 independent network reconstruction runs. For each reconstruction run, we generated a fixed number of 10 datasets of length 3 from the Boolean network of Covert *et al.* (2004) by choosing starting states randomly. In each run, we calculated the respective correlation values for all potential dependencies. This generates $25\times583(=14575)$ AUC values (583 genes are dependent). One AUC value is determined by calculating all correlations of one of the 583 dependent genes to the 685 variables that are potentially influencing them and utilizing all possible thresholds. From these values, we generated histograms separated by the number of either 1, 2, 3 or more than 3 input variables (Fig. 3). For Boolean functions with only one input variable (representing genes which are regulated by just one factor), the correlation method identified the relevant variables with very high reliability. For functions with several input variables (representing genes which are regulated by several factors), the fraction of smaller AUC values increased, but a large fraction is still close to 1.

Interactions of the yeast cell cycle transcriptional network: finally, we used the correlation method to reconstruct a biological network from microarray data. For that purpose, we chose the cell cycle transcriptional network of budding yeast, as suggested by Orlando *et al.* (2008), and used published microarray results from three groups (Cho *et al.*, 1998; Pramila *et al.*, 2006; Spellman *et al.*, 1998) for network reconstruction. All three studies represent genome-wide analyses of gene expression during the cell cycle in synchronized yeast cells, but they differ in the applied synchronization methods and the time intervals at which transcript levels were measured. In a first step, we binarized each of these microarray datasets with the 2-means algorithm, the version of the k-means clustering algorithm that generates $k=2$ clusters. Next, we created a set of

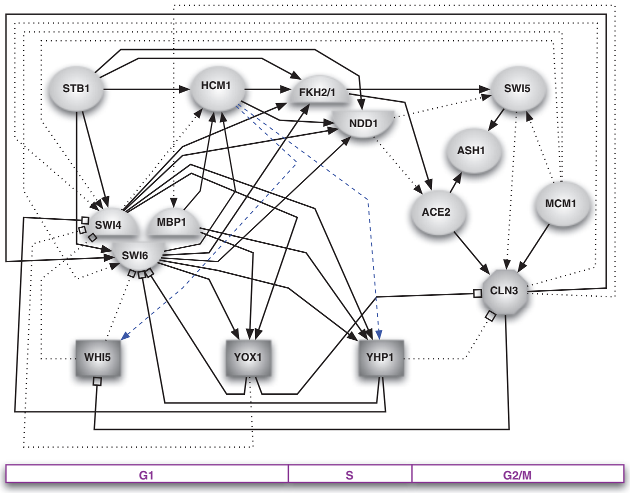

Fig. 4. Yeast cell cycle transcription factor network. The diagram shows the yeast transcription factor network as suggested by Orlando *et al.* (2008) with the transcription factors arranged on the cell cycle time line (G1 →S→G2/M) on the basis of their peak transcript levels. Interactions shown as dashed lines are based on a publication by Pramila *et al.* (2006). Transcriptional activators are depicted as circles, repressors as rectangles and the cyclin Cln3 as octagon; activating interactions end in arrowheads and inhibitory interactions in squares. The interactions shown as solid and dashed lines were correctly identified by our approach, while interactions shown as dotted lines were not found.

example pairs, i.e. each example contained the state of the network at a certain time point along with a succeeding state measured after a specific time period. As the time lag between regulator expression and target expression varies considerably in the data, we grouped the sample pairs according to the intervals: the short time lag group contained all examples where the following states were measured after a period of 5–10 min, the intermediate time lag group contained the examples with the following states measured after 14–21 min, and the long time lag group contained the examples with the following states measured after 25–30 min. Thus, for network reconstruction we regarded subsequent states with a time lag of up to about one fourth of the yeast's cell cycle, which is assumed to span a time of about 135 min (Orlando *et al.*, 2008). For each of the 16 genes with known dependencies in the published cell cycle transcriptional network, we computed the correlation to the other genes in the cell cycle network and also added 16 randomly chosen additional genes contained in all of the microarray datasets. This was done to distort the reconstruction process and to evaluate the detection of non-present dependencies. A computed correlation was assumed to represent a dependency, if it exceeded the mean of all correlations either by at least 1 SD for at least one of the time lag groups or by at least half the SD for at least two of the time lag groups. This evaluation was repeated for a total of 25 times, each time choosing a new set of 16 additional genes at random. For comparison, we also applied the best fit extension algorithm to the same data. This algorithm was run on the data corresponding to each of the three time lag groups, and for maximum in-degrees of 3, 4 and 5.

Figure 4 shows the correctly identified dependencies in the yeast transcription factor network (Orlando *et al.*, 2008). Assuming that this published network correctly and fully represents the real biological situation, our correlation inference approach correctly identified an average of 74.7% of the present dependencies and 59.1% of non-present dependencies for the described threshold. In comparison, Lähdesmäki and co-workers' best fit algorithm found 61.1% of the present dependencies and 38.8% of the non-present dependencies. This best result was achieved with a maximum in-degree of 5 and interpreting a dependency as present when it was found for at least one of the three time lag groups. The identification rates of the best fit algorithm for different maximum in-degrees are given in the Supplementary Material I. Additionally, we analyzed precision and recall as well as F-scores for correlation and best fit algorithm, applying different thresholds for the correlation. In these analyses we found, that the initially chosen threshold of 'mean ± 1SD' provided a reasonable balance between recall and precision (see Supplementary Figs S9 and S10). An example of the interactions found by the described analysis are given in Supplementary Material III. Here, an interaction was included if it occurred in the majority of the 25 simulations. Finally, we examined differences in precision and recall, when each of the three time lag groups was used separately for network reconstruction (see Supplementary Fig. S12).

4 DISCUSSION

Under the assumption that a Boolean network consists of monotone functions, we have shown that its dependencies can be reconstructed via Pearson correlation of subsequent states. Compared for instance to the best fit extension algorithm, correlations have the important advantage that they can be computed very quickly—for each pair of nodes, only two correlations for the two possible directed dependencies have to be computed, where the running time for the computation of a single correlation is linear in the number of samples. This leads to an overall running time of the order $O(n^2m)$, where n is the number of nodes and m the number of samples. In contrast, an exhaustive search algorithm like the best fit algorithm considers all subsets of genes up to a given size, assuming the functions of the network have an input degree of k, for each node of the network $\binom{n}{k}$ combinations of input nodes have to be considered, which leads to a running time of the order $O(n^{k+1}m)$. This is also reflected in Table 1: for $k=3$, doubling the number of nodes in the network increases the runtime of the best fit algorithm by a factor of 16, while it increases the runtime of the correlation method only by a factor of 4. So while the exhaustive search approach is only feasible for networks with a small number of nodes or low input degrees, the correlation method can also be applied to large networks. This is particularly interesting for biological applications, as in the context of microarray and deep-sequencing technologies this type of large-scale data becomes more available.

Theoretically, dependencies in the examples violate the assumption of Theorem 1. In an experimental evaluation (Supplementary Figs S2–S4), we found that these dependencies only marginally influence the reconstruction process as long as these dependencies are not too strong (visually discernible only for bias probabilities greater than 0.8).

While our new method is to some extent similar to using the Fourier transform, it does not need the Fourier transform's mathematical overhead (Bshouty and Tamon, 1996; Mossel *et al.*, 2003). We do not see any specific advantage of moving from the time domain into the spectral domain, as the values are (in the case of the Fourier transform) hard to interpret. We think that the notion of a correlation value is much more intuitive.

Our experiments on the randomly created artificial networks have shown that for noisy data and a low number of samples, Pearson correlation still finds a high percentage of the dependencies and, for a range of thresholds, even surpasses the best fit algorithm with regard to recall, precision and F-score (see Fig. 1, Supplementary Figs S5 and S6). A further advantage of our approach, besides the short running times, is the possibility to vary the threshold according to the desired outcome, i.e. whether a high recall or rather a high precision are requested. In case of little or no previous knowledge, for example, preferably a high threshold should be applied so that dependencies can be assumed with a high reliability, while a lower threshold can be applied, if more previous knowledge exists that can help to (pre-)select meaningful dependencies.

For the simulated *E.coli* network, we also could reconstruct more than 50% of the present and non-present dependencies, already from a relatively low number of datasets (Fig. 2). The comparison of the areas under the ROCs indicate that especially for functions that depend only on one or two variables, dependencies can be found reliably. For functions depending on more variables, the identification of dependencies is more complicated, but the AUCs show that the correlation method still is able to provide information about these dependencies (Fig. 3). One reason for the increased difficulty when reconstructing functions with higher in-degree is the fact that for higher in-degree, variables can have a lower influence and dependencies are therefore harder to detect. In addition, the set of Boolean functions can be partitioned into sets of varying difficulty for a learning algorithm [cf. Gordon and Peretto (1990)]. This partition, however, also depends to a large extent on the in-degree of the functions.

Furthermore, we could show that Pearson correlation also performed reasonably well when reconstructing dependencies from real biological data, reaching identification rates (true positive, true negative) similar to those for the simulated *E.coli* datasets. Compared to the best fit algorithm, interactions were not only identified faster using correlation, but the identification was also more reliable. As seen also in Figure 4, almost 75% of the assumed dependencies could be correctly identified at the chosen threshold, in spite of the regulatory complexity of the yeast cell cycle network, even for components regulated by more than three factors (like Swi4, Swi6 or Yhp1). Further examining which of the expected dependencies were not identified by our method, we find that in these cases the regulatory mechanisms are based on a quite complex interplay of several factors involved. So, for example, Cln3 can activate transcription of Swi6 through inactivation of the transcriptional repressor Whi5 as well as independent of Whi5 (Wittenberg and Reed, 2005), and the Whi5-independent mechanism might hamper identification of the inhibitory influence of Whi5 on Swi6. A second example is the transcriptional activator Mcm1, which can act in concert with an activating transcription factor complex consisting of Fkh2/Ndd1 or the transcriptional repressors Yhp1 and Yox1 (Wittenberg and Reed, 2005). Accordingly, the influence of Mcm1, which was mostly not identified by our method, is strongly dependent on the presence of the respective co-factors, for which our approach correctly identified the respective dependencies. With regard to the reconstruction from real data, it has to be kept in mind that we cannot be completely sure that the published network whose interactions we are rebuilding exactly matches the real *in vivo* situation. Thus, for example, some identified dependencies, that were assumed to be false positives, might actually be real dependencies. In line with this reasoning, the precision of network reconstruction was indeed lower for the reconstruction from real data, for both the best-fit and the correlation algorithm. Furthermore, it has to be kept in mind, that correlations representing indirect effects within the cell cycle also were counted as false positives, while they do actually confer some biological meaning.

In addition, Pearson correlation is not only suitable for measuring the dependencies between sequences of binarized values. Especially when real-valued examples are difficult to binarize, applying Pearson correlation directly on the raw, non-binarized data might already give some valuable information about a network.

5 CONCLUSION

For a Boolean network consisting only of monotone Boolean functions, we showed that Pearson correlation is a fast method to find dependencies in the network. This method makes it possible to analyze large networks that contain nodes with large input degree. We could show for both simulated and real microarray data that our approach could reconstruct large parts of published regulatory networks.

ACKNOWLEDGEMENTS

The authors would especially like to thank Guido Adler for continuing support. The authors would also like to thank the anonymous reviewers for their valuable comments.

Funding: Graduate School of Mathematical Analysis of Evolution, Information and Complexity (to H.A.K.). International Graduate School in Molecular Medicine at Ulm University which is funded by the Excellence Initiative of German governments (GSC 270 to B.K.). Federal ministry of education and research (BMBF) within the framework of the program of medical genome research (PaCa-Net; project ID PKB-01GS08, in part). The responsibility for the content lies exclusively with the authors.

Conflict of Interest: none declared.

REFERENCES

Alon,U. (2006) *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman and Hall/CRC.

Babu,M. *et al.* (2004) Structure and evolution of transcriptional regulatory networks. *Curr. Opin. Struct. Biol.*, **14**, 283–291.

Bornholdt,S. (2005) Systems biology: less is more in modeling large genetic networks. *Science*, **21**, 449–451.

Bshouty,N. and Tamon,C. (1996) On the fourier spectrum of monotone functions. *J. ACM (JACM)*, **43**, 747–770.

Cho,R. *et al.* (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, **2**, 65–73.

Covert,M. *et al.* (2004) Integrating high-throughput and computational data elucidates bacterial networks. *Nature*, **429**, 92–96.

Faith,J. *et al.* (2007) Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biol.*, **5**, e8.

Fawcett,T. (2006) An introduction to ROC analysis. *Pattern Recognit. Lett.*, **27**, 861–874.

Gordon,M. and Peretto,P. (1990) The statistical distribution of Boolean gates in two-inputs, one-output multilayered neural networks. *J. Phys. A Math. Gen.*, **23**, 3061.

Hoeffding,W. (1963) Probability inequalities for sums of bounded random variables. *J. Am. Stat. Assoc.*, **58**, 13–30.

Kauffman,S. (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, **22**, 437–467.

Kauffman,S. (1993) *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, USA.

Kauffman,S. *et al.* (2003) Random Boolean network models and the yeast transcriptional network. *Proc. Natl Acad. Sci. USA*, **100**, 14796–14799.

Lähdesmäki,H. *et al.* (2003) On learning gene regulatory networks under the boolean network model. *Mach. Learn.*, **52**, 147–167.

Liang,S. *et al.* (1998) Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pac. Symp. Biocomput.*, **3**, 18–29.

Margolin,A. *et al.* (2006) Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, **7**, S7.

Mossel,E. *et al.* (2003) Learning juntas. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of Computing*, ACM, New York, NY, USA, pp. 206–212.

Müssel,C. *et al.* (2010) BoolNet - an R package for generation, reconstruction, and analysis of Boolean networks. *Bioinformatics*, **26**, 1378–1380.

Opgen-Rhein,R. and Strimmer,K. (2007) Learning causal networks from systems biology time course data: an effective model selection procedure for the vector autoregressive process. *BMC Bioinformatics*, **8** (Suppl. 2), S3.

Orlando,D. *et al.* (2008) Global control of cell-cycle transcription by coupled CDK and network oscillators. *Nature*, **453**, 944–947.

Orphanides,G. and Reinberg,D. (2002) A unified theory of gene expression. *Cell*, **108**, 439–451.

Pearl,J. (2000) *Causality*. Cambridge University Press, Cambridge.

Pramila,T. *et al.* (2006) The forkhead transcription factor hcm1 regulates chromosome segregation genes and fills the s-phase gap in the transcriptional circuitry of the cell cycle. *Genes Dev.*, **20**, 2266–2278.

Rani,T.S. *et al.* (2007) Analysis of *E.coli* promoter recognition problem in dinucleotide feature space. *Bioinformatics*, **23**, 582–588.

Shipley,B. (2000) *Cause and Correlation in Biology: A User's Guide to Path Analysis, Structural Equations and Causal Inference*. Cambridge University Press, Cambridge.

Spellman,P. *et al.* (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, **9**, 3273.

Spirtes,P. *et al.* (2000) *Causation, Prediction, and Search.*, 2nd edn. MIT Press, Cambridge, MA.

Tsukimoto,H. and Hatano,H. (2003) The functional localization of neural networks using genetic algorithms. *Neural Networks*, **16**, 55–67.

Venters,B.J. and Pugh,B.F. (2009) How eukaryotic genes are transcribed. *Crit. Rev. Biochem. Mol. Biol.*, **44**, 117–141.

Wittenberg,C. and Reed,S. (2005) Cell cycle-dependent transcription in yeast: promoters, transcription factors, and transcriptomes. *Oncogene*, **24**, 2746–2755.

Zoppoli,P. *et al.* (2010) Timedelay-aracne: Reverse engineering of gene networks from time-course data by an information theoretic approach. *BMC Bioinformatics*, **11**, 154.