BIOINFORMATICS ORIGINAL PAPER

Vol. 21 no. 11 2005, pages 2706–2713
doi:10.1093/bioinformatics/bti388

Systems biology

Inferring genetic regulatory logic from expression data

Svetlana Bulashevska[1,*] and Roland Eils[1,2]

[1]Division 'Theoretical Bioinformatics', German Cancer Research Center, Im Neuenheimer Feld 280, 69120 Heidelberg, Germany and [2]Department 'Bioinformatics and Functional Genomics', Institute of Pharmacy and Molecular Biotechnology (IPMB), University of Heidelberg, Germany

Received on May 18, 2004; revised on March 8, 2005; accepted on March 9, 2005
Advance Access publication March 22, 2005

ABSTRACT

Motivation: High-throughput molecular genetics methods allow the collection of data about the expression of genes at different time points and under different conditions. The challenge is to infer gene regulatory interactions from these data and to get an insight into the mechanisms of genetic regulation.

Results: We propose a model for genetic regulatory interactions, which has a biologically motivated Boolean logic semantics, but is of a probabilistic nature, and is hence able to confront noisy biological processes and data. We propose a method for learning the model from data based on the Bayesian approach and utilizing Gibbs sampling. We tested our method with previously published data of the *Saccharomyces cerevisiae* cell cycle and found relations between genes consistent with biological knowledge.

Availability: The code for the software BUGS is available upon request.

Contact: s.bulashevska@dkfz.de

Supplementary information: http://oslo.inet.dkfz-heidelberg.de/ibios_old/people/bulashev/Supplement/

INTRODUCTION

One of the goals of functional genomics is to understand the mechanisms of genetic regulation. The advent of microarray technology facilitated the large-scale monitoring of gene expression. Typically, the expression data are processed with clustering algorithms for the identification of groups of co-expressed genes. Then, the regulatory regions of the co-expressed genes are analyzed to detect common overrepresented motifs, based on the assumption that co-expressed genes might be co-regulated by a common regulator. However, the expression level of a gene can depend on multiple transcription factors, and, therefore, on multiple genes. The regulatory control is provided by the cooperative binding of transcription factors to the binding sites of genes (*cis*-regulatory elements). Genes assigned to one cluster by clustering analysis might belong to different regulatory or signalling pathways. We propose a method for the analysis of gene expression data which is based on the explicit modelling and inference of gene regulatory interactions.

The working principles of the *cis*-regulatory elements can be described by means of logic (Kauffman, 1996). Some genes can be activated by one of a few different possible transcription factors ('OR' logic). Other genes require that two or more transcription factors must all be bound for activation ('AND' logic). The transcriptional activation of some genes may be inhibited by one of a few possible repressor proteins ('NOT OR' logic; in our notation, 'NOR'). In case of 'OR–NOR' logic, a gene is regulated by a set of possible activators and a set of possible inhibitors. The gene is transcribed if and only if one of its possible activators is active and it is not repressed by one of its possible repressors. The gene's regulatory interactions can be presented as logic gates (Fig. 1).

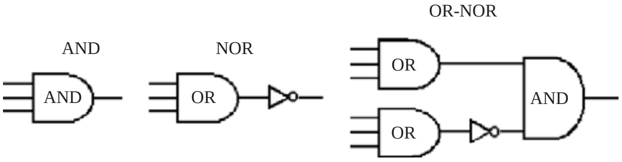

Fig. 1. Examples of regulatory functions presented as logic gates.

A pioneering attempt to model genetic regulation was based on the Boolean network model (Kauffman, 1996; Somogyi and Sniegosky, 1996; Liang *et al*., 1998). In the Boolean network the expression state of each gene is functionally related to the expression states of some other genes using logical rules. The major limitation of the Boolean network model is its inherent determinism, in contradiction with the stochastic nature of the underlying process of gene transcription and with the noisy character of the experimental measurements of messenger RNA (mRNA). Friedman *et al*. (2000) proposed to employ the Bayesian network for modelling the genetic regulatory network. The Bayesian network (Pearl, 1998; Jensen, 1996; Heckerman, 1998) is a probabilistic model; i.e. it uses probability as a means to express uncertainty about modelling variables and their dependencies. The Bayesian network is a directed acyclic graph (DAG) G, whose vertices correspond to the random variables $X_1, \dots, X_n$. The graph G encodes conditional independencies between the variables: given the value of its parents in G, the variable is conditionally independent of other variables in the network except its descendants. Due to the notion of conditional independence, probabilistic dependencies among the variables in the network can be represented only by the specification of conditional probability distributions (CPD). The CPD for a variable defines its conditional probability given every possible combination of the values of its parents. Hence, the global relations of genes in the genetic network can be described as being composed of local interactions between each gene and its regulatory genes.

The Bayesian network formalism allows modelling arbitrary interactions between parents $X_1, \dots, X_n$ of a variable Y. The complete CPD for a binary variable with n parents requires the specification

[*]To whom correspondence should be addressed.

of $2^n - 1$ independent parameters (one parameter for each parent's state configuration). This combinatorial semantics of the parents' interaction in the Bayesian network makes it difficult to interpret the results of Bayesian network learning and to uncover the 'true' *cis*-regulatory logical relationships covered in this presentation. The exponential explosion of the parameter space makes model learning computationally expensive. Besides the computational complications, in small datasets, there might be not sufficient cases available for learning conditional probabilities. Learning distributions with fewer parameters is more reliable. We propose a model for genetic regulatory interactions that combines the simple and biologically motivated Boolean logic semantics of Boolean networks and the possibility of dealing with uncertainty offered by Bayesian networks. In contrast to Bayesian networks, the parents' interactions of variables in our model are defined with logical functions. We present a general framework that allows for a particular gene to find a set of its regulators (activators and inhibitors), given a particular Boolean logic function governing this regulation.

In the following we first introduce our model of gene regulation which originates from the field of probabilistic graphical models. Then we present our approach for learning the structure and parameters of the model from gene expression data, which is based on the Bayesian methodology. Since there is no closed form solution for the problem of Bayesian model selection, we applied the Markov Chain Monte Carlo (MCMC) simulation technique, namely Gibbs sampling. We introduced an additional parameter into the model so that the problem of model selection transformed into a variable selection task. We tested our approach on the gene expression dataset of the *Saccharomyces cerevisiae* cell cycle.

SYSTEMS AND METHODS

The model of gene regulatory interactions

The Bayesian network formalism exploits independencies among variables in the network and achieves more compact representations of the joint probability distribution of the variables by expressing them with conditional probability distributions. One can further exploit the independencies between parents of a variable in a Bayesian network to get more compact representations of CPDs. In the past, several models were proposed with special types of causal interaction (Heckerman and Breese, 1994; Meek and Heckerman, 1997; Srinivas, 1993). One type of such models is the causal independence model which uses the notion of independence of parents of each variable in the model. The variables $X_1, \ldots, X_n$, which are parents of the variable Y, can affect Y through independent 'mechanisms'. The results of these effects are combined by a rule represented with a Boolean-logic function. Such models were introduced by Pearl (1998) and were called 'noisy OR-Gate' and 'noisy AND-Gate'.

We employ these kinds of models for modelling the genetic regulatory interactions. We assume that the variable X_i (regulator) can execute its influence on the variable Y (regulatee) independently of other possible regulators $X_1, \ldots, X_n$ of Y. The biological mechanism underlying this modelling assumption is the binding of protein transcribed by the regulator to the DNA of the regulatee. This process is not deterministic; rather each gene X_i can regulate the gene Y with probability θ_i and can fail to do this with probability $1-\theta_i$. The general structure of the gene interaction in our models is represented by a directed graph (Fig. 2). In this graphical representation, intermediate variables $I_1, \ldots, I_n$ are introduced, through which the variables $X_1, \ldots, X_n$ execute their influence on a given common effect variable Y.

Each intermediate variable I_i has only one parent, the variable X_i. Its probability distribution is defined as follows: given that $X_i = 1$, I_i takes the value 1 with probability θ_i and the value 0 with probability $1-\theta_i$, respectively. Given that $X_i = 0$, I_i takes the value 0 with probability 1. The combined regulatory influence on the variable Y is calculated as the Boolean function F on the input variables $I_1, \ldots, I_n$. If $X_1, \ldots, X_n$ are activators, then the state of the variable Y is $F(I_1, \ldots, I_n)$; if $X_1, \ldots, X_n$ are inhibitors, the state of Y is $1 - F(I_1, \ldots, I_n)$. The Boolean 'interaction function' F defines in which way the intermediate affects I_i, and indirectly, in which way the variables X_i interact. We consider two interaction functions: AND and OR. The semantics of the OR-function implies that the variables X_i are each assumed to be sufficient to influence Y. In the case of AND-function, all variables X_i need to execute their own influence on the variable Y so that Y will be active.

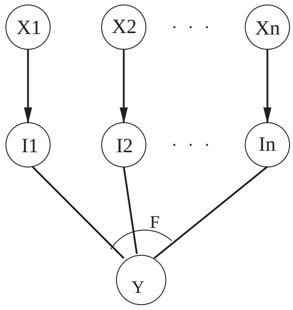

Fig. 2. Model of gene regulatory interactions; F—Boolean function ('AND', 'OR').

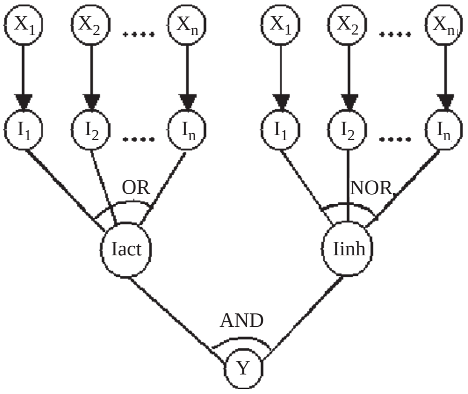

Fig. 3. Complex model of gene regulatory interactions with activators and inhibitors ('OR–NOR' regulation).

Introduction of the hidden state variables I_i allows the insertion of 'noise' into the Boolean-logic based models. It allows modelling such that the biological mechanism of the regulation of one gene by another could be inhibited for unknown reasons. Thus, the input variables can be considered as observables from which we make our noisy measurements, while the hidden variables have the 'true' latent biological values.

In the present work we consider simple models with activatory regulation ('OR', 'AND') and inhibitory regulation ('NOR', 'NAND'), as well as complex models: 'AND–NAND', 'AND–NOR', 'OR–NAND' and 'OR–NOR'. In the complex models the regulatory influences of multiple activators and multiple inhibitors are combined with AND-function as exemplified in Figure 3.

The conditional probability distribution for the regulatee Y that can be activated by two possible activators ('OR'-activation) is presented in Table 1. Note that the model with the Boolean logic-based interaction of parent variables allows the specification of the entire conditional probability distribution for a variable with only n parameters $\theta_1, \ldots, \theta_n$; i.e. polynomial on number of parents.

Bayesian model selection

We employ the Bayesian methodology for learning the structure and parameters of the model from data. The Bayesian approach addresses the problem

Table 1. Conditional probability table of regulatee Y that is regulated by two possible activators $X1$ and $X2$ (‘OR’-activation)

		Y	
$X1$	$X2$	0	1
0	0	1	0
1	0	$1-\theta_1$	θ_1
0	1	$1-\theta_2$	θ_2
1	1	$(1-\theta_1)(1-\theta_2)$	$1-(1-\theta_1)(1-\theta_2)$

as calculating the posterior probability of a model given data for a collection of candidate models and selecting the most probable model. Suppose that the data D has been generated by a model m, one of a set M of candidate models, $m \in M$. If $p(m)$ is the prior probability of model m, then the posterior model probability by Bayes rule is $p(m|D) \propto p(D|m)p(m)$. The marginal likelihood $p(D|m)$ is calculated as $p(D|m) = \int p(D|m,\theta_m)p(\theta_m|m)\mathrm{d}\theta_m$, where $p(\theta_m|m)$ is the prior distribution of model parameters θ_m for model m. The calculation of the marginal likelihood is the general computational bottleneck of the Bayesian methodology, since the integral is analytically tractable only in certain restricted examples, when a prior distribution for the parameters of the model exists, so that the integral will have a closed form solution (*conjugate prior*).

Consider the model with ‘OR’-activation. Assume the variable Y is commonly influenced by the variables $X_1, \ldots, X_n$. The probability distribution of Y given the values of its parents can be written as:

$$P(Y=0|\theta) = \prod_{i=1}^{n}(1-\theta_i)^{X_i}$$

and

$$P(Y=1|\theta) = 1-\prod_{i=1}^{n}(1-\theta_i)^{X_i},$$

where $\theta = (\theta_1, \ldots, \theta_n)$ is the vector of parameters. Assume we have a sample of N cases corresponding to the states of the variables $X_1, \ldots, X_n$ and the variable Y. Denote by Y_j the state of the variable Y in case j, and by X_{ij} the state of the variable X_i in case j. The likelihood function is then

$$L(\theta) = \prod_{j=1}^{N}\left(\prod_{i=1}^{n}(1-\theta_{ij})^{X_{ij}}\right)^{1-Y_j}\left(1-\prod_{i=1}^{n}(1-\theta_{ij})^{X_{ij}}\right)^{Y_j}$$

If we substitute ψ_{ij} by $-\log(1-\theta_{ij})$, the likelihood function transforms into

$$L(\psi) = \prod_{j=1}^{N}(e^{-\eta_j})^{1-Y_j}(1-e^{-\eta_j})^{Y_j},$$

where $\eta_j = \sum_{i=1}^{n}\psi_{ij}X_{ij}$ is a linear predictor. This is the generalized linear model (McCullagh and Nelder, 1983). There is no conjugate prior for the model, since it cannot be expressed in the form of the general exponential family parametric models. (For introduction to conjugate analysis, see Bernardo and Smith, 1994.) The ‘AND’ model is intractable analogously.

The ‘OR’ model can be written as:

$$Y \sim \mathit{Bernoulli}\left(1-\prod_{i=1}^{n}(1-\theta_i)^{X_i}\right)$$

(The operator $\sim$ stands for ‘is distributed as’.) Now consider the complex model ‘OR–NOR’. Assume the variable Y is influenced by a set of activators $X_1^{\mathrm{act}}, \ldots, X_n^{\mathrm{act}}$ and a set of inhibitors $X_1^{\mathrm{inh}}, \ldots, X_k^{\mathrm{inh}}$. The variable Y takes the value 1, if the activators executed their influence *and* the inhibitors failed, otherwise Y is 0. The ‘OR–NOR’ model then can be defined as:

$$Y \sim \mathit{Bernoulli}\left(\left(1-\prod_{i=1}^{n}(1-\theta_{ij}^{\mathrm{act}})^{X_{ij}^{\mathrm{act}}}\right)\prod_{i=1}^{k}(1-\theta_{ij}^{\mathrm{inh}})^{X_{ij}^{\mathrm{inh}}}\right)$$

This model is also intractable.

Jaakkola and Jordan (1996) apply variational methods and propose the lower and upper bound approximations of the posterior distributions of the ‘OR’ and ‘AND’ models. However, approximation techniques require a high number of training data that are usually not available within gene expression studies.

In recent years, the development of MCMC techniques facilitated the estimation of posterior probabilities involved in the Bayesian learning (Gilks, 1993). The MCMC technique is a stochastic simulation technique, which generates samples from the joint posterior distribution of the unknown quantities in a model allowing to make estimates on them. Sampling from the joint posterior distribution $p(m, \theta_m|D)$ allows one to estimate the posterior model probability $p(m|D)$ and the posterior parameter probability $p(\theta_m|D)$.

One of the MCMC approaches is Gibbs sampling (Geman and Geman, 1984). Gibbs sampling reduces the problem of dealing simultaneously with a large number of unknown parameters in a joint distribution into a much simpler problem of dealing with one variable at a time, iteratively sampling each from its full conditional distribution given the current values of all other variables in the model. As stated by Pearl (1987), performing Gibbs sampling is particularly appropriate for a graphical model. Due to the factorization of the joint probability distribution, the full conditional for a given node in the DAG involves only a subset of nodes participating in its Markov blanket (i.e. the set of parents, children and parents of the children for a node).

Gibbs variable selection

Our problem of model selection is formulated as follows: given the data on the gene Y and its potential regulators $X_1, \ldots, X_p$, for a given Boolean logic function F, identify the subset $X_1, \ldots, X_n$ of actual regulators of Y. Standard MCMC techniques such as Gibbs sampler cannot be directly applied for the model selection because of the variable size of the problem space (candidate models have different number of parameters). Gibbs sampling approaches applicable for model selection problems were developed by George and McCulloch (1996), Kuo and Mallick (1998) and by Dellaportas *et al*. (2000, 2002). It was proposed to substitute the model indicator $m \in M$ with the *variable indicator* $\gamma = (\gamma_1, \ldots, \gamma_p)$, a binary vector, representing which of the X_j, $j = 1, \ldots, p$, should be included in the desirable ‘true’ model. This allows the consideration of one joint space of the model parameters and the variable indicator, keeping the dimensionality constant across all possible models. By introducing the variable indicator the ‘OR’ model may be written as

$$Y \sim \mathit{Bernoulli}\left(1-\prod_{i=1}^{n}(1-\theta_i)^{\gamma_i X_i}\right)$$

The model selection problem is then referred to as the variable selection problem.

The Bayesian approach requires setting up a joint probability distribution over all parameters, in our case $p(\theta, \gamma)$. Let D denote the observed data for the variables X_j, $j = 1, \ldots, p$ and Y. The joint posterior distribution given the observed data is $p(\theta, \gamma|D)$. The Gibbs sampling procedure samples successively from univariate conditional distributions, simulating a Markov chain

$$\theta^{(0)}, \gamma^{(0)}, \theta^{(1)}, \gamma^{(1)}, \ldots, \theta^{(t)}, \gamma^{(t)}, \ldots$$

which converges in distribution to $p(\theta, \gamma|D)$. The subsequence

$$\gamma^{(0)}, \gamma^{(1)}, \ldots, \gamma^{(t)}, \ldots$$

converges to $p(\gamma|D)$. This sequence can be used to identify the high probability values of γ_j. These are the values that appear most frequently in the sequence.

Consider a partition of θ into $(\theta_\gamma, \theta_{-\gamma})$ corresponding to those components of θ which are included and not included, respectively, in the model. Then the posterior distribution of the parameters $p(\theta|\gamma, D)$ may be partitioned into $p(\theta_\gamma|\theta_{-\gamma}, \gamma, D)$ and $p(\theta_{-\gamma}|\theta_\gamma, \gamma, D)$. From the model definition it is obvious that the components of the vector $\theta_{-\gamma}$ do not affect the model likelihood. The full conditional posterior distributions required for the Gibbs sampling procedure are given by:

$$p(\theta_\gamma|\theta_{-\gamma},\gamma,D) \propto p(D|\theta,\gamma)p(\theta_\gamma|\gamma)p(\theta_{-\gamma}|\theta_\gamma,\gamma),$$
$$p(\theta_{-\gamma}|\theta_\gamma,\gamma,D) \propto p(\theta_{-\gamma}|\theta_\gamma,\gamma),$$

where $p(D|\theta,\gamma)$ is the model likelihood, $p(\theta_\gamma|\gamma)$ is the model prior and $p(\theta_{-\gamma}|\theta_\gamma,\gamma)$ is the *pseudoprior*.

In our model the terms γ_j of the variable indicator γ are independent. Each γ_j can be sampled from a Bernoulli distribution with success probability $O_j/(1+O_j)$, where

$$\begin{aligned} O_j &= \frac{p(\gamma_j=1|\gamma_{-j},\theta,D)}{p(\gamma_j=0|\gamma_{-j},\theta,D)} \\ &= \frac{p(D|\theta,\gamma_j=1,\gamma_{-j})}{p(D|\theta,\gamma_j=0,\gamma_{-j})}\,\frac{p(\theta|\gamma_j=1,\gamma_{-j})}{p(\theta|\gamma_j=0,\gamma_{-j})}\,\frac{p(\gamma_j=1,\gamma_{-j})}{p(\gamma_j=0,\gamma_{-j})}\end{aligned}$$

The methods for Gibbs variable selection differ in their approaches on specifying prior distributions for the model parameters. The most simple is the 'unconditional prior' approach of Kuo and Mallick where the prior distribution of model parameters θ is defined independent of variable indicator γ. In the Stochastic Search Variable Selection (SSVS) method of George and McCulloch, the priors for θ_j depend on γ_j and are defined as mixtures of two Normal distributions for $\gamma_j = 0$ and $\gamma_j = 1$. If $\gamma_j = 0$, the parameters (pseudopriors) are kept close to 0 by defining the mean of the normal distribution equal to 0. The method of Dellaportas *et al*. (2000, 2002) differs from SSVS in that the pseudopriors may not be distributed around 0; rather they may be chosen in a way to help increase the efficiency of the sampling procedure. Efficient performance can be achieved when the moves of the MCMC chain between different models γ could be 'local'. In variable selection problems, where the new sampled value of γ differs from the current value in a single component, it is reasonable to retain the parameter values for those terms γ_j which are present in both the current and new models. Dellaportas *et al*. use *proposal* densities for the pseudopriors. These proposal densities can be estimated using a *pilot run* of the MCMC for the *saturated* model; i.e. the model where all terms $\gamma_j = 1$ for all j. In the present work we adopt the method of Dellaportas *et al*. (2000, 2002).

Bayesian modelling allows for the hierarchical formulation of the model: the distributions for the parameters can be formulated, in turn, with the help of hyperparameters. We defined the parameter priors with a Beta distribution with hyperparameters a_j and b_j:

$$\theta_j \sim Beta(a_j, b_j)$$

Beta distribution constrains the parameters to the $[0, 1]$-interval. The hyperparameters a_j and b_j were defined equal to 1, if $\gamma_j = 1$, therefore making the prior non-informative ($Beta(1, 1)$). If $\gamma_j = 0$, we calculated the proposal distributions for the pseudopriors, following Dellaportas *et al.*. That is, we calculated the hyperparameters a_j and b_j by the formulas (*method of moments*):

$$\begin{aligned} a_j + b_j &= \frac{mean_j(1-mean_j)}{var_j} - 1, \\ a_j &= (a_j+b_j)mean_j, \\ b_j &= (a_j+b_j)(1-mean_j), \end{aligned}$$

where $mean_j$ and var_j, the mean and the variance of the parameters θ_j, were estimated from the pilot run of the saturated model.

Next, one must define the prior distribution for the variable indicator γ. Since the terms γ_j are independent, the prior can be decomposed into independent Bernoulli distributions for each term: $\gamma_j \sim Bernoulli(\pi_j)$, where π_j is the prior probability to include term j into the model. A simple and popular choice in variable selection problems is the uniform prior on γ, assuming that models are a priori equally probable, i.e. $\pi_j = \pi = 0.5$. This prior is noninformative in the sense of favoring all models equally, but is not non-informative with respect to the model size. If p is the number of potential regulators, and n is the number of actual regulators, then $E(n) = 0.5p$ and $var(n) = 0.25p$. For example, if $p = 19$ (as in our test study described below), then n lies in the range 5–14 with prior probability close to 1, and thus it is possible that the sampling procedure will not sample models with <5 regulators. This may be crucial for 'AND' models, since there might be a sparse number of regulators of a gene combined with AND-function. To favor more parsimonious models, one can set the probability π so as to restrict n a priori to lie in a short range by setting $E(n)$ and $var(n)$ to the desired values, and using

$$E(n) = \pi * p, \qquad var(n) = \pi(1-\pi)p$$

A more flexible approach is to place a hyperprior on π,

$$\pi \sim Beta(\alpha, \beta)$$

then the prior for the number of actual regulators n is Beta-binomial:

$$n \sim Betabin(p, \alpha, \beta)$$

The values for α and β can be chosen by setting $E(n)$ and $var(n)$ to the desired values and solving the following equations (Kohn *et al*., 2001):

$$p\frac{\alpha}{\alpha+\beta} = E(n)$$
$$\frac{\alpha+1}{\alpha+\beta+1} = \frac{var(n) - E(n)(1-E(n))}{(p-1)E(n)}$$

While performing Gibbs variable selection with the complex models like 'OR–NOR', we considered the same set of variables (genes) as potential activators and inhibitors. We used two variable indicators, γ^{act} and γ^{inh}, representing that a particular variable is included in the model as activator or inhibitor, respectively. To ensure that terms γ_j^{act} and γ_j^{inh} cannot be 1 at the same time, we specified γ_j^{inh} as:

$$\gamma_j^{\text{inh}} \sim Bernoulli((1-\gamma_j^{\text{act}})\pi_j^{\text{inh}})$$

where π_j^{inh} is the prior probability to include the term j into the set of 'true' inhibitors.

We have implemented Gibbs variable selection by utilizing BUGS (Bayesian updating with Gibbs sampling), the general purpose software for Gibbs sampling on graphical (DAG) models (Spiegelhalter *et al*., 1996; Gilks, 1993; Ntzoufras, 1999 http://www.ba.aegean.gr/ntzoufras/tr.htm). BUGS provides a declarative language for specifying a graphical model. The BUGS code for our models is available upon request. The runs of the MCMC can be monitored using the package CODA implemented in R-language (http://cran.r-project.org).

The output of Markov chain simulation can be used to summarize the posterior distribution of the variables of interest. After the burn-in time of 2000 iterations, we used 10 000 Markov chain simulations to count the number of times γ_j had the value 1 in the chain. If the frequency of 1s in the chain exceeded 0.7, we assumed that $\gamma_j = 1$ and the respective regulator should be included in the 'true' model. Otherwise, the regulator j should be excluded. For the complex models, like 'OR–NOR', we used 5000 iterations for the burn-in, and 10 000 iterations for the frequency estimations. The examples of the traces of MCMC simulations for parameters γ_j and θ_j are available in the supplementary material.

The Markov chain must be monitored for diagnosing slow convergence or lack of convergence. As proposed by Gelman and Rubin (1992), a number of parallel runs of Markov chains should be carried out from different starting points. Convergence is diagnosed when the output from different Markov chains is indistinguishable. For parallel runs of Markov chains we used different initial values of the parameter indicator γ (when $\gamma_j = 0$ for all j and when $\gamma_j = 1$ for all j). Procedures for monitoring convergence of MCMC are available in the package CODA.

Model checking

After the execution of the Gibbs variable selection and the estimation of the variable indicator γ, the check of goodness-of-fit of the model to data is required, to check whether the model assumptions were appropriate.

Bayesian model checking uses the posterior predictive distributions (Gelman *et al.*, 2000). The goal is to perform posterior predictions under the model and to assess the discrepancy between predicted and observed data. If the model is reasonably accurate, the predicted data should be similar to the observed data.

Here we wish to check the ability of the inferred regulatory model to predict the state of the gene Y from the states of its regulators. Let y be the observed data on Y and θ be the vector of parameters. Denote y^{rep} the *replicated* data generated under the model with parameters θ. The posterior predictive distribution is

$$p(y^{\text{rep}}|y) = \int p(y^{\text{rep}}|\theta)p(\theta|y)\mathrm{d}\theta$$

The posterior predictive distribution can be computed by simulation: simulate parameters θ from their posterior distribution, and simulate y^{rep} from the sampling distribution $p(y^{\text{rep}}|\theta)$ conditioning on values of the simulated parameters. An advantage of using BUGS is that the generation of the replicate data can be easily incorporated into the model inference procedure. Based on the current simulated values of the parameters θ obtained at each iteration of the MCMC, we generate replicate dataset $\{y^{\text{rep}}\}$ from the sampling distribution of Y.

Our model-checking strategy is based on the examination of individual observations of Y, $y_i, i = 1, \dots, N$ (N is the number of data samples) and the comparison of them to the posterior predictive distributions. For the comparison we use the residual function $r_i = y_i - E(y_i)$, where the expectation $E(y_i)$ is estimated based on the replicate dataset. Observations for which the residual is not close to 0 indicate some lack-of-fit of the model and should be regarded as outliers. We regarded the residual as not close to 0 if its absolute value exceeded one estimated standard deviation. We calculate the model prediction accuracy as the percentage of non-outliers.

RESULTS

To test our approach for inferring genetic regulatory interactions, we used the microarray data from Spellman *et al.* (1998) [including the data from Cho *et al.* (1998)] obtained for *S.cerevisiae* cell cultures that were synchronized by three different methods. Accordingly, the study contains three different datasets: the *cdc15*, *cdc28* and alpha-factor datasets. We used the *cdc15* experiment (arrest of *cdcd15* temperature-sensitive mutant) containing the largest number of data samples (25) as the training dataset. The remaining experimental datasets, *cdc28* (18 samples) and alpha-factor (19 samples), were used as test sets.

For the discretization of the continuous gene expression values into two states (0—not expressed; 1—expressed), we used a vector quantization technique, namely the clustering algorithm k-means (Gersho and Gray, 1992). For each gene we clustered its expression values into two groups by the k-means algorithm with two initial values: 0 and the maximum expression value of the gene.

Since our data are a time-series data, two different regulatory situations can be considered. First, the state of the gene i in the sample j depends on the states of its regulators in the same sample. Second, the state of the gene i in the sample j depends on the states of its regulators in the previous sample $j - 1$. We refer to the first type of regulation as 'simultaneous', and to the second type of regulation as 'time delay'.

The gene transcription in the mitotic division of the yeast is coordinated in a periodic manner according to the consecutive phases of the cell cycle G1, S, G2, M and M/G1 (for a review, see Mendenhall and Hodge, 1998). Events such as DNA replication and chromosome segregation are promoted with the actions of specific cyclin-dependent kinases (CDKs), which are dependent on the activity of cyclins. The cycle periodicity requires also degradative, proteolytic processes that eliminate cyclically acting proteins at stages when they are no longer required. Some cell cycle transitions are negatively regulated by specific inhibitors that must be eliminated in a timely fashion to initiate cell cycle transition.

In our pilot study we considered the group of 20 genes known to be involved in cell-cycle regulation of *S.cerevisiae*. The same set of genes was used by Chen *et al.* (2000), who presented a mathematical model of the cell-cycle events. We applied our approach for learning the models 'AND', 'OR', 'NOR', 'NAND', 'AND–NAND', 'AND–NOR', 'OR–NAND' and 'OR–NOR' from the data, for each gene in the dataset, considering all other genes in the dataset as candidate regulators. We considered both 'simultaneous' and 'time delay' problems. After the vector of variable indicators was obtained by Gibbs variable selection procedure, we performed model checking. The results are displayed in Supplementary Tables 1–5.

We have experimented with different settings of the prior for the variable indicator γ. We tried the Bernoulli distribution with parameters $\pi = 0.5$ and $\pi = 0.1$, and also the setting with the Beta distribution described previously. We tried $\mathit{Beta}(16, 133)$ that keeps expectation and variance of the number of actual regulators $E(n) = 2$, $\mathit{var}(n) = 2$, and also $\mathit{Beta}(0.8, 14.4)$ with $E(n) = 1$, $\mathit{var}(n) = 2$. The results of the 'OR' and 'OR–NOR' models with these different prior settings appeared to be the same, but for the 'AND' model, which is apparently more restrictive, we found few regulatory relations for some genes with $\mathit{Bernoulli}(0.1)$ and Beta distribution settings (Supplementary Tables 3–5).

For some genes the 'NOR'-model suggested more inhibitors than the 'OR–NOR'-model (Supplementary Tables). Obviously, the two models have different semantics. Learning the 'NOR'-model identifies only the inhibitors of a gene; i.e. the model 'explains' the non-activity of the gene with the activity of its inhibitors. By the 'OR–NOR'-model, the non-activity of the regulatee can also be 'explained' with the failure of its activators. Finally, we checked the 'OR–NOR'-model with the activators and inhibitors suggested by learning all possible models, and selected the results giving the highest accuracy. The results for the case of 'simultaneous' regulation are summarized in Table 2, and for the case of 'time delay' regulation are presented in Table 3.

We validated the regulatory interactions learned from the *cdc15* dataset on the alpha-factor and *cdc28* datasets. The results of the model checking for these datasets are presented in the last two columns of Table 2. Highly accurate regulatory interactions were found for the genes CLN1, CLN2, CLB1, CLB2, CLB5, SWI5 and SWI4. Some of the regulatory models induced from the *cdc15* dataset had poor confirmation in the alpha-factor and *cdc28* datasets. The reason for this might be that some genes have much stronger signals during the *cdc15* experiment than during the other two.

The inferred genetic interactions for the 'simultaneous' regulation are presented graphically in Figure 4; the relationship between genes regulating one common gene is described by 'OR'-function. (The graph was generated with the program GraphViz, www.graphviz.org.) Our results are consistent with previous biological knowledge: the interrelationships between the genes reflect the coincidence with different phases of the cell cycle. The genes CLN1 and CLN2 transcribing the G1 cyclins and the genes CLB5 and CLB6 transcribing the B-cyclins Clb5 and Clb6 are expressed in the G1-phase. Note the activatory connections amongst the genes CLN1, CLN2, CLB5 and CLB6. The 'time delay' learning revealed the activatory influences CLN1→CLN2, CLB6→CLB5, CLN1→CLB6 and CLN3→CLB6 (CLN3 is also the G1-specific

Table 2. 'OR–NOR' regulators of the genes inferred from the *cdc15* dataset[a]

Genes	Activators	Inhibitors	Accuracy (%)		
			cdc15	*α*	*cdc28*
CLN1	CLB6	CLB2	84	63	78
	CDC28	CLB2	80	63	67
CLN2	CLB5	CDC20	80	74	94
CLN3	No	CLB4, MCM1	88	—	61
CLB1	CLB2, SWI5	CLB6, SIC1, SWI4	92	79	83
CLB2	CLB1, SWI5	CLB6, SIC1, SWI4	96	84	67
CLB4	CDC34	CLN3, CLB6	80	—	—
	CDC34	CLN3	88	—	—
CLB5	CLN2	CDC20	80	—	72
	CLN2	No	88	73	89
CLB6	CLN1, CLN2	CLB2	84	68	77
MCM1	CLN2	CDC20	72	—	61
SIC1	SWI4	No	76	—	72
SWI6	CLB5	CDC20	72	—	61
CDC28	No	No	No	No	No
CDC53	No	No	No	No	No
MBP1	CLN3, SWI5, HCT1	CDC34	92	74	61
CDC34	CLB2, CLB4, SIC1	MBP1	92	58	61
SWI5	CLB1, CLB2	SWI4	92	79	72
SKP1	MBP1	CLB6, CDC34	68	63	—
SWI4	SWI6	CLB2	88	68	78
CDC20	SIC1, CDC34	CLN2	80	63	67
HCT1	No	No	No	No	No

[a] 'Simultaneous' gene activities considered. The last two columns present the accuracy of models as validated on the alpha-factor and *cdc28* datasets.

Table 3. 'OR–NOR' regulators of the genes inferred from the *cdc15* dataset ('time delay' gene activities considered)

Genes	Activators	Inhibitors	Accuracy (%)
CLN1	No	No	No
CLN2	CLN1	CLB2	92
CLN3	CDC20	MCM1, SWI6	63
CLB1	No	CLN3, CLB6, SIC1	63
CLB2	CLB1	CLN3, CLB6, SIC1	92
CLB4	SKP1	CDC20	67
CLB5	CLB6	CLB1, CLB2	63
CLB6	CLN1, CLN3	No	84
MCM1	SWI4	MBP1	75
SIC1	CLN3	SWI5	71
SWI6	No	No	No
CDC28	No	No	No
CDC53	No	CLN2	71
MBP1	No	No	No
CDC34	SKP1	No	88
SWI5	CLB1	CLN3, SIC1	83
SKP1	CDC53	MCM1	63
SWI4	SKP1	SWI5	88
CDC20	No	No	No
HCT1	SWI5	CLN1, SKP1	63

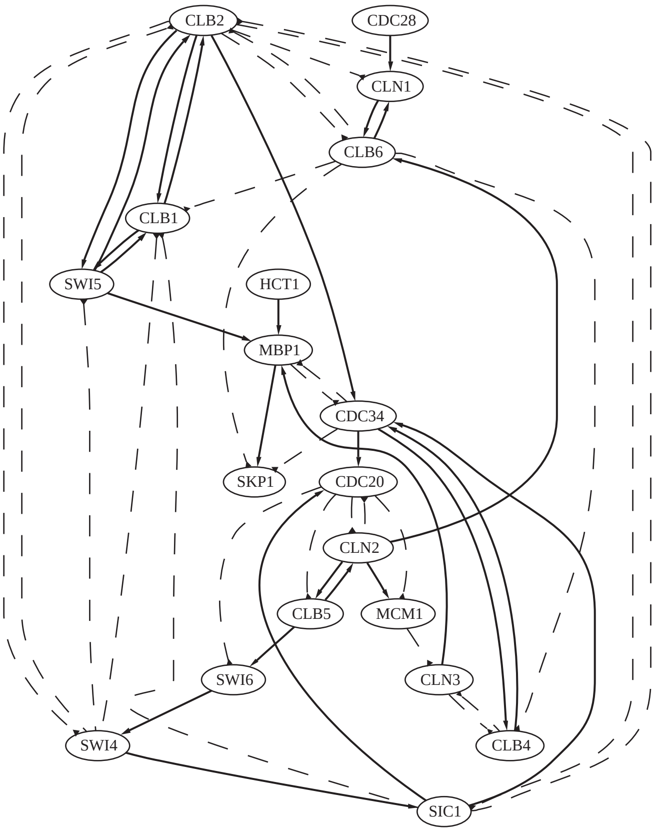

Fig. 4. Regulatory interactions of 20 genes of *S.cerevisiae*. The full arcs represent activatory regulation, the dashed arcs represent inhibitory regulation. The relationship between genes regulating one common gene is described by 'OR'-function.

cyclin). The genes CLB1 and CLB2 are G2-specific cyclins, the gene SWI5 is the transcription factor also known to be expressed in G2-phase. Note the activatory connections between the genes CLB1, CLB2 and SWI5. The 'time delay' problem inferred the activatory regulation CLB1→SWI5 and CLB1→CLB2 (the 'time delay' 'AND' model suggested SWI5→CLB1, SWI5→CLB2). The inhibitory influences were inferred between the G1- and G2-specific genes confirming that the expression of these genes is separated in phases. The 'time delay' learning also revealed the inhibitory connections: CLB1, CLB2⊣CLB5, CLB2⊣CLN2, CLB6⊣CLB1, CLB6⊣CLB2, CLN3⊣ SWI5, CLN3⊣ CLB1 and CLN3⊣ CLB2. The gene regulatory interactions described above find support in the literature (Althoefer *et al*., 1995; Loy *et al*., 1999; Hwang *et al*., 1998; Schneider *et al*., 1998; Toyn *et al*., 1997).

SWI6 encodes Swi6, the regulatory component of SBF and MBF transcription factor complexes controlling the expression of genes in G1-phase and important for start-specific gene expression. Protein Swi4 is a component of the SBF complex and forms a complex with Swi6. The 'simultaneous' learning found the activatory connection from SWI6 to SWI4 and negative connections from SWI4 to the genes expressed in G2-phase. Both 'simultaneous' and 'time delay' learning revealed that when SWI5 is active, SWI4 is inactive.

SIC1 is known to be an inhibitor of the Clb complexes and is active in the G1-phase maintaining CLB1 and CLB2 in an inactive state. Note the inhibitory connections of SIC1 to the G2 cyclins CLB1 and CLB2 in Figure 4. The 'time delay' learning also inferred the inhibitory influence of SIC1 on the genes CLB1, CLB2 and SWI5 (Toyn *et al*., 1997).

The ‘simultaneous’ learning found a positive association between SIC1 and CDC20, CDC34. It can be explained with this that the CDK complexes CDC20 and CDC34 are needed for proteolytic degradation of Sic1 at the G1–S boundary to trigger the initiation of the DNA synthesis.

The gene CDC20 is required for proteolytic degradation of G1 regulators. This explains the negative connections of CDC20 to SWI6 and to MCM1, both of them encoding transcription factors. CDC20 is transcribed in the late S/G2 phase, whereas CLN2 and CLB5 are expressed in G1, explaining the negative connection between CDC20 and these genes. The gene CDC34 encodes Cdc34 which is the E2 ubiquitin-conjugating enzyme required for proteolytic degradation. In the results for ‘simultaneous’ regulation, the genes CDC34 and MBP1 negatively influence each other, likely because the activity of CDC34 and the activity of MBP1, as part of the MBF transcription factor complex, are completely separated in time (Goebl *et al*., 1994). In Figure 4 there is a negative connection from the gene CDC34 to the gene SKP1, whereas Skp1 is the E3 ubiquitin ligase which is needed for Cdc34 essential function. However, the ‘time delay’ learning revealed the positive influence of SKP1 on CDC34. Apparently, a time interval is needed between the transcription of these genes to achieve their function (Willems *et al*., 1999). CDC34 is required for the proteolysis of Clb proteins Clb2 and Clb4 at the border of G2–M (positive connections from CLB2 and CLB4 to CDC34 in Fig. 4).

Both ‘simultaneous’ and ‘time delay’ results display the negative connection from MCM1 to CLN3, which is explained by time separation in the activities of these genes. The gene CLN3 is expressed at the M/G1 border. The MCM1 gene encodes the transcription factor and is active during the G2/M transition.

The graph displayed in Figure 4 is not a directed acyclic graph; rather it contains cycles. For some genes (for instance, CLB1 and CLB2) the symmetric interactions were found. Note that we learned local models; i.e. for each gene we considered all other genes as possible regulators, without testing any global criteria for gene interactions. The symmetric activatory and inhibitory relations between pairs of genes could be potentially found by clustering. However, we have found many more relations representing complex regulatory dependencies between multiple genes which would not be unravelled by clustering.

Obviously, most of the regulatory interactions coordinating the cell division cycle of the yeast occur at the protein level (phosphorylation, proteolytic degradation, etc.). At the present moment we do not have measurements about concentrations of proteins during the yeast cell cycle. The interactions reconstructed from the gene expression data can give only hypotheses on the true biological relationships.

DISCUSSION

In this paper we proposed a model for the genetic regulatory interactions and presented a method for learning the structure and parameters of the model from gene expression data. Our model represents the Boolean logic semantics of *cis*-regulatory logic. In contrast to the standard Boolean networks applied earlier for modelling gene interactions, our model has a probabilistic nature, more suitable for dealing with the stochastic biological process of genetic regulation and noisy experimental data. The model is a probabilistic graphical model explicitly representing the dependencies between a gene and its regulators. It can be seen as an intermediate model between the models of local interactions defined in Boolean networks and Bayesian networks. The model is not fully observable; it rather contains hidden variables representing factors that could not be measured. Due to the statistical context of the model, unlike Boolean networks, we could employ the methodology of Bayesian statistics for learning the model from data.

The Bayesian modelling allows for flexibility in defining complex models with many parameters. By inserting into the model a new parameter, namely the variables indicator, it was possible to convert the model selection problem into the variable selection task. The learning of the resulting model was facilitated with Gibbs sampling. In contrast to the classical model estimation methods such as maximum likelihood, the Bayesian learning is free from assumptions of asymptotic normality, and therefore is more appropriate for learning from sparse datasets.

Previously, models of genetic regulation were suggested with the same idea of extending Boolean networks to make them robust against noise. In the noisy Boolean networks of Akutsu *et al*. (2000), the authors defined a probability with which a certain number of input/output patterns of gene expression will not be discarded by an inference algorithm, even if a certain Boolean function is not satisfied. In contrast, our approach inserts ‘noise’ directly into the model as model parameters enabling the application of statistical learning for model inference. Shmulevich *et al*. (2002) presented probabilistic Boolean networks. They inserted ‘noise’ into the model by accomodating more than one possible Boolean functions for each node in the network. They introduced a probability with which a certain Boolean function is selected from the set of possible functions for calculating the output of the target gene. We see the source of uncertainty in genetic regulation not in the realizations of different Boolean functions, but rather in the fact that independent basic elements of the genetic regulatory mechanism could fail to execute their regulatory influence.

Another class of regulation functions, called chain functions, was suggested by Gat-Viks *et al*. (2003). The authors study the computational problem of reconstructing the chain functions using a minimum number of perturbation experiments. They also consider combinations of several chains with a Boolean function. Unlike our approach, the chain function model assumes that the functional relations are deterministic.

Segal *et al*. (2003) also employ probabilistic modelling to infer classes of genes (possibly ‘molecular pathways’) which exhibit similar expression profiles. The genes are likely to fall into the same class if their protein products interact. Pe’er *et al*. (2002) infer small sets of active regulators and their regulatees by using a scoring function based on mutual information.

We developed a general computational framework enabling us to define a model of gene interactions with a particular regulatory function and to perform learning of this model from data. Given expression data on a gene and its potential regulators, our methodology is able to detect the most likely regulators of the gene. The main advantage of our approach is that the relationships found with our method do not require laborious manual analysis for their interpretation, as the arbitrary combinatorial interactions are learned with standard Bayesian networks algorithms. Rather, the results of model inference can be directly utilized in an automatic system for analyzing transcription factor binding sites in the regulatory regions of the genes.

Our method allows for elucidating more complex multi-gene relations which go beyond pairwise relations retrieved by clustering algorithms that are widely used for the analysis of gene expression data. We tested our method with the data of the *S.cerevisiae* cell cycle and found relations between genes consistent with previously published biological knowledge. Although we have exemplified our approach on a relatively small subset of genes, it can be readily applied to larger datasets.

One of the advantages of the Bayesian approach is that it enables including 'subjective' prior information into the model. In this study we used the subjective prior specification to enforce the number of gene regulators to lie in the desired range. Potentially, one could define priors aiming to incorporate previous biological knowledge into the model learning.

Regulatory pathways of the cell rely not only on the transcriptional regulation but to a great extent on the post-transcriptional and external signalling events. The reconstruction of the genetic regulatory interactions from expression data can only reveal an incomplete picture of the genetic regulatory pathways. Unobserved events on the protein level can be represented in a probabilistic model by introducing hidden variables. When more detailed proteomics data will be available, it can still be handled by our approach. Our future goal is to extend the framework described here by integrating information on genes' regulatory sequences and genes' functional annotations. An example of such an integrated approach considering both gene expression and promoter sequence data in a unified probabilistic model is the work of Segal *et al*. (2003b).

ACKNOWLEDGEMENTS

We are grateful to Ioannis Ntzoufras, Department of Business Administration, University of the Aegean, Greece, for his useful suggestions. We further wish to thank Peter Lichter, DKFZ Heidelberg, for his continuous support. This work was funded by the German Federal Ministery for Research and Education (BMBF) through the National Genome Research Network (NGFN, 01GR0101 and 01GR0450).

REFERENCES

Akutsu,T. *et al*. (2000) Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, **16**, 727–734.

Althoefer,H. *et al*. (1995) Mcm1 is required to coordinate G2-specific transcription in *Saccharomyces cerevisiae. Mol. Cell Biol.*, **19**, 5917–5928.

Bernardo,J.M. and Smith,A.F.M. (1994) *Bayesian Theory*, *Wiley Series in Probability and Mathematical Statistics*. John Wiley and Sons, Chichester.

Chen,K.C. *et al*. (2000) Kinetic analysis of a molecular model of the budding yeast cell cycle. *Mol. Biol. Cell*, **11**, 369–391.

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

Dellaportas,P., Forster,J.J. and Ntzoufras,I. (2000) Bayesian variable selection using the Gibbs sampler. In Dey,D.K., Ghosh,S. and Mallick,B. (eds), *Generalized Linear Models: A Bayesian Perspective*. Marcel Dekker, New York, pp. 271–286.

Dellaportas,P. *et al*. (2002) On Bayesian model and variable selection using MCMC. *Statist. Comput.*, **12**, 27–36.

Friedman,N. *et al*. (2000) Using Bayesian network to analyze expression data. *J. Comput. Biol.*, **7**, 601–620.

Gat-Viks,I. *et al*. (2004) Reconstructing chain functions in genetic networks. *Pacific Symp. Biocomput.*, **9**, 498–509.

Gelman,A. and Rubin,D.B. (1992) Inference from iterative simulation using multiple sequences. *Statist. Sci.*, **7**, 457–472.

Gelman,A., Carlin,J.B., Stern,H.S. and Rubin,D.B. (2000) *Bayesian Data Analysis*. Chapman and Hall/CRC Press.

Geman,S. and Geman,D. (1984) Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, **6**, 721–741.

George,E.I and McCulloch,R.E. (1993) Stochastic search variable selection. In Gilks,W.R., Richardson,S. and Spiegelhalter (eds), *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London, UK, pp. 203–214.

Gersho,A. and Gray,R.M. (1992) *Vector Quantization and Signal Compression*. The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, Dordrecht.

Gilks,W.R., Richardson,S. and Spiegelhalter,D.J. (eds) (1993) *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London.

Goebl,M.G. *et al*. (1994) The Ubc3 (Cdc34) ubiquitin-conjugating enzyme is ubiquitinated and phosphorylated *in vivo*. *Mol. Cell Biol.*, **14**, 3022–3029.

Heckerman,D. (1998) A tutorial on learning with Bayesian networks. In Jordan,M.I. (ed.), *Learning in Graphical Models*. Kluwer Academic Publishers, Dordrecht.

Heckerman,D. and Breese,J.S. (1994) Causal independence for probability assessment and inference using bayesian networks. *IEEE Trans. Syst., Man Cybernet.*, **26**, 826–831.

Hwang,L.H. *et al*. (1998) Budding yeast Cdc20: a target of the spindle checkpoint. *Science*, **13**, 1041–1044.

Jaakkola,T.S. and Jordan,M.I. (1996) Computing upper and lower bounds on likelihoods in intractable networks. In *Proceeding of 12th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, Portland, OR, pp. 340–348.

Jensen,F.V. (1996) *Introduction to Bayesian Networks*. Springer, New York.

Kauffman,S.A. (1996) *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, New York.

Kohn,R. *et al*. (2001) Nonparametric regression using linear combinations of basis functions. *Statist. Comput.*, **11**, 313–322.

Kuo,L. and Mallick,B. (1998) Variable selection for regression models. *Sankhya B*, **60**, 65–81.

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

Loy,C.J. *et al*. (1999) NDD1, a high-dosage suppressor of cdc28-1N, is essential for expression of a subset of late-S-Phase-specific genes in *Saccharomyces cerevisiae*. *Mol. Cell Biol.*, **19**, 3312–3327.

McCullagh,P. and Nelder,J.A. (1983) *Generalized Linear Models*. Chapman and Hall, London.

Meek,C. and Heckerman,D. (1997) Structure and parameter learning for causal independence and causal interaction models. In *Proceedings of the thirteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Francisco, CA, pp. 366–375.

Mendenhall,M.D. and Hodge,A.E. (1998) Regulation of Cdc28 cyclin-dependent protein kinase activity during the cell cycle of the yeast *Saccharomyces cerevisiae. Microbiol. Mol. Biol. Rev.*, **62**, 1191–1243.

Ntzoufras,I. (1999) Gibbs variable selection using BUGS, *Technical report*.

Pearl,J. (1987) Evidential reasoning using stochastic simulation of causal models. *Artif. Intell.*, **32**, 245–257.

Pearl,J. (1998) *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Francisco, CA.

Pe'er,D. *et al*. (2002) Minreg: inferring an active regulator set. *Bioinformatics*, **18** (Suppl. 1), s258–s267.

Schneider,B.L. *et al*. (1998) Yeast G1 cyclins are unstable in G1 phase. *Nature*, **395**, 86–89.

Segal,E. *et al*. (2003a) Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics*, **19** (Suppl. 1), i264–i272.

Segal,E. *et al*. (2003b) Genome-wide discovery of transcriptional modules from DNA sequence and gene expression. *Bioinformatics*, **19** (Suppl. 1), i273–i282.

Shmulevich,I. *et al*. (2002) Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, **18**, 261–274.

Somogyi,R. and Sniegosky,C.A. (1996) Modeling the complexity of genetic networks: understanding multigenic and pleiotropic regulation. *Complexity*, 1.

Spellman,P.T. *et al*. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast *Saccaharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, **9**, 3273–3297.

Spiegelhalter,D.J., Thomas,A. and Best,N.G. (1996) Computation on Bayesian graphical models. In Bernardo,J.M., Berger,J.O., Dawid,A.P. and Smith,A.F.M. (eds), *Bayesian Statistics*, Vol. 5, pp. 407–425.

Srinivas,S. (1993) A generalization of the noisy-or model. In *Proceedings of the ninth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Francisco, CA.

Toyn,J.H. *et al*. (1997) The Swi5 transcription factor of *Saccharomyces cerevisiae* has a role in exit from mitosis through induction of the Cdk-inhibitor Sic1 in telophase. *Genetics*, **145**, 85–96.

Willems,A.R. *et al*. (1999) SCF ubiquitin protein ligases and phosphorylation-dependent proteolysis. *Philos. Trans. Soc. Land Biol. Sci.*, **354**, 1533–1550.