



Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks

Dirk Husmeier

Biomathematics and Statistics Scotland (BioSS), JCMB, The King's Buildings,
Edinburgh, EH9 3JZ, UK

Received on February 27, 2003; revised on May 5, 2003; accepted on May 29, 2003

ABSTRACT

Motivation: Bayesian networks have been applied to infer genetic regulatory interactions from microarray gene expression data. This inference problem is particularly hard in that interactions between hundreds of genes have to be learned from very small data sets, typically containing only a few dozen time points during a cell cycle. Most previous studies have assessed the inference results on real gene expression data by comparing predicted genetic regulatory interactions with those known from the biological literature. This approach is controversial due to the absence of known gold standards, which renders the estimation of the sensitivity and specificity, that is, the true and (complementary) false detection rate, unreliable and difficult. The objective of the present study is to test the viability of the Bayesian network paradigm in a realistic simulation study. First, gene expression data are simulated from a realistic biological network involving DNAs, mRNAs, inactive protein monomers and active protein dimers. Then, interaction networks are inferred from these data in a reverse engineering approach, using Bayesian networks and Bayesian learning with Markov chain Monte Carlo.

Results: The simulation results are presented as receiver operator characteristics curves. This allows estimating the proportion of spurious gene interactions incurred for a specified target proportion of recovered true interactions. The findings demonstrate how the network inference performance varies with the training set size, the degree of inadequacy of prior assumptions, the experimental sampling strategy and the inclusion of further, sequence-based information.

Availability: The programs and data used in the present study are available from <http://www.bioass.sari.ac.uk/~dirk/> Supplements

Contact: dirk@bioass.ac.uk

INTRODUCTION

Molecular pathways consisting of interacting proteins underlie the major functions of living cells. A central goal of molecular biology is therefore to understand the regulatory mechanisms of gene transcription and protein synthesis, and

the invention of DNA microarrays, which measure the abundance of thousands of mRNA targets simultaneously, has been hailed as an important milestone in this endeavour. Several approaches to the reverse engineering of genetic regulatory networks from gene expression data have been explored, reviewed, for instance, by D'haeseleer *et al.* (2000) and De Jong (2002).

At the most refined level of detail is a mathematical description of the biophysical processes in terms of a system of coupled differential equations that describe, for example, the processes of transcription factor binding, diffusion, protein and RNA degradation, etc.; see, for instance, Chen *et al.* (1999). While such low-level dynamics are critical to a complete understanding of regulatory networks, they require detailed specifications of both the relationship between the interacting agents as well as the parameters of the biochemical reaction, like reaction rates, diffusion constants, etc. Obviously, this approach is therefore restricted to very small systems. In a recent study, Zak *et al.* (2002) found that a system of ordinary differential equations describing a regulatory network of three genes with their respective mRNA and protein products is not identifiable when only gene expression data are observed, and that rich data, including detailed information on protein–DNA interactions, are needed to ensure identifiability of the parameters that determine the interaction structure.

At the other extreme of the spectrum is the coarse-scale approach of clustering. Following up on the seminal paper by Eisen *et al.* (1998), several clustering methods have been applied to gene expression data, reviewed, for instance, by D'haeseleer *et al.* (2000). Clustering provides a computationally cheap way to extract useful information out of large-scale expression data sets. The underlying conjecture is that co-expression is indicative of co-regulation, thus clustering may identify genes that have similar functions or are involved in related biological processes. The disadvantage, however, is that clustering only indicates which genes are co-regulated; it does *not* lead to a fine resolution of the interaction processes, indicating, for instance, whether an interaction between two genes is direct or mediated by other genes, whether a gene is

a regulator or regulatee, etc. Clustering, in effect, only groups interacting genes together in a monolithic block, where the detailed form of the regulatory interaction patterns is lost.

A promising compromise between these two extremes is the approach of Bayesian networks (Pearl, 1988; Krause, 1998; Heckerman, 1999), which were first applied to the problem of reverse engineering genetic networks from microarray expression data by Friedman *et al.* (2000), Pe'er *et al.* (2001) and Hartemink *et al.* (2001). Bayesian networks are interpretable and flexible models for representing probabilistic relationships between multiple interacting agents. At a *qualitative* level, the structure of a Bayesian network describes the relationships between these agents in the form of conditional independence relations. At a *quantitative level*, relationships between the interacting agents are described by conditional probability distributions. The probabilistic nature of this approach is capable of handling noise inherent in both the biological processes and the microarray experiments. This makes Bayesian networks superior to Boolean networks (Kauffman, 1969, 1993), which are deterministic in nature.

The probabilistic nature of Bayesian networks makes the inference scheme robust and allows the confidence in the inferred network structures to be estimated objectively. However, the application to gene expression data is particularly hard in that interactions between hundreds of genes have to be learned from very sparse data sets, typically containing only a few dozen time points during a cell cycle. It is therefore not yet clear whether an elicitation of biological network structures from expression data is at all possible, that is, whether the posterior probabilities over network structures can be expected to be sufficiently informative. Most previous studies have assessed the inference results on real gene expression data by comparing predicted genetic regulatory interactions with those known from the biological literature. This approach has two inherent problems, resulting from the absence of known gold standards. First, the estimation of the sensitivity is controversial. Having detected an interaction between two genes from microarray expression data with Bayesian networks, authors tend to delve into the biological literature to substantiate their findings. Sometimes they back up their results with interactions between proteins whose sequences are similar to the ones encoded by genes studied in the performed experiment. Besides the fact that sequence similarity does not necessarily imply similar functions, there is an inherent arbitrariness in deciding on a cut-off value for the *E*-score in a BLAST (Altschul *et al.*, 1990) search. One may therefore suspect that some of the reported *true* interactions are spurious and do not really support the interactions detected in the reverse engineering procedure. The second and more serious drawback is the difficulty in estimating the false detection rate. This is because on predicting a gene interaction that is not supported by the literature, it is impossible to decide, without further expensive interventions in the form of multiple gene knock-out experiments, whether the algorithm has

discovered a new, previously unknown interaction, or whether it has flagged a spurious edge.

The objective of the present study, therefore, is to test the viability of the Bayesian network paradigm in a realistic simulation study. The first section recapitulates the concept of Bayesian networks and discusses the advantages of *dynamic* over *static* Bayesian networks. The second section describes a synthetic benchmark study. The third section describes a realistic simulation: first, artificial gene expression data are generated by simulating a known biological network involving DNAs, mRNAs, inactive protein monomers and active protein dimers. Then, regulatory networks are inferred from these data in a reverse engineering approach, using Bayesian networks and Bayesian learning with Markov chain Monte Carlo (MCMC). Finally, in the discussion section, the results are compared with recent related studies.

METHODS

A Bayesian network is defined by a graphical structure, \mathcal{M} , a family of (conditional) probability distributions, \mathcal{F} , and their parameters, \mathbf{q} , which together specify a joint distribution over a set of random variables of interest. The graphical structure consists of a set of *nodes* or *vertices*, \mathcal{V} , and a set of *directed edges* or *arcs*, \mathcal{E} : $\mathcal{M} = (\mathcal{V}, \mathcal{E})$. The nodes represent random variables, while the edges indicate conditional dependence relations. If we have a directed edge from node *A* to node *B*, then *A* is called the *parent* of *B*, and *B* is called the *child* of *A*. Take, as an example, Figure 1, top left, where we have the set of vertices $\mathcal{V} = \{A, B, C, D, E\}$, and the set of edges $\mathcal{E} = \{(A, B), (A, C), (B, D), (C, D), (D, E)\}$. Node *A* does not have any parents. Nodes *B* and *C* are the children of node *A*, and the parents of node *D*. Node *D* itself has one child: node *E*. The graphical structure has to take the form of a directed acyclic graph or DAG, which is characterized by the absence of directed cycles, that is, cycles where all the arcs point into the same direction. A DAG offers a simple and unique rule for expanding the joint probability in terms of simpler conditional probabilities. Let X_1, X_2, \dots, X_n be a set of random variables represented by the nodes in the graph, and define $pa[X_i]$ to be the set of parents of X_i . Then

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | pa[X_i]) \quad (1)$$

In applying this method to the inference of genetic networks, we associate nodes with genes and their expression levels, while edges indicate interactions between the genes. For instance, the network structure of Figure 1, top left, suggests that gene *A* initiates the depicted transcription cascade, that genes *B* and *C* co-regulate gene *D*, and that gene *D* mediates the interaction between genes (*B*, *C*) and *E*. In what follows, the family of distributions \mathcal{F} is assumed to be known and fixed. We therefore need to distinguish between the *model* or *network structure* \mathcal{M} , which is the set of edges connecting the

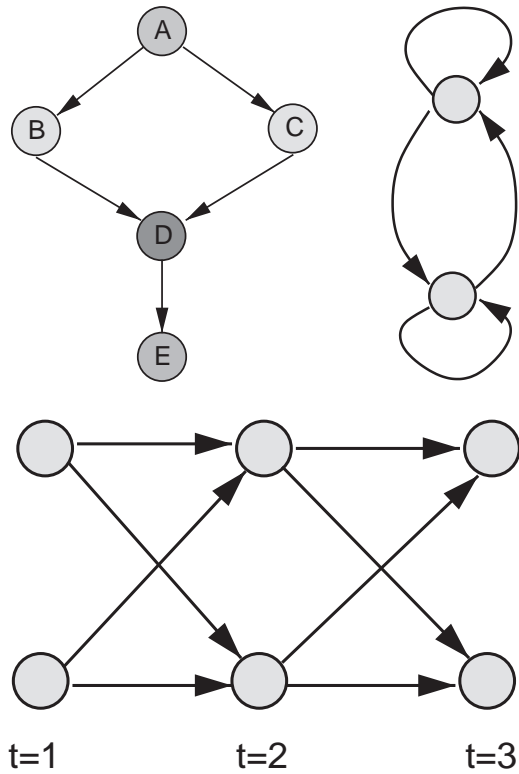


Fig. 1. Bayesian networks. Top left: A simple Bayesian network, for which the joint probability factorizes into $P(A, B, C, D, E) = P(A)P(B|A)P(C|A)P(D|B, C)P(E|D)$. Top right: Recurrent network comprising two genes with feedback that interact with each other. This is *not* a Bayesian network. Bottom: Equivalent dynamic Bayesian network obtained by unfolding the recurrent network in time.

nodes (as in Fig. 1, top left), and the *network parameters* \mathbf{q} . For fixed \mathcal{F} , the latter define the conditional probabilities associated with the edges and determine, for instance, whether the influence of one gene on another is of the form of an excitation or inhibition. Our objective is to learn the network from the data \mathcal{D} produced by a microarray experiment, which, in principle, requires finding the structure \mathcal{M}^* that maximizes $P(\mathcal{M}|\mathcal{D})$,

$$\mathcal{M}^* = \arg \max_{\mathcal{M}} \{P(\mathcal{M}|\mathcal{D})\}, \quad (2)$$

and the parameters \mathbf{q}^* that maximize $P(\mathbf{q}|\mathcal{D}, \mathcal{M}^*)$. From Bayes rule we have

$$P(\mathcal{M}|\mathcal{D}) = \frac{1}{Z} P(\mathcal{D}|\mathcal{M})P(\mathcal{M}), \quad (3)$$

where $Z = \sum_{\mathcal{M}} P(\mathcal{D}|\mathcal{M})P(\mathcal{M})$ is a normalization factor, $P(\mathcal{M})$ is the prior probability on network structures, and

$$P(\mathcal{D}|\mathcal{M}) = \int P(\mathcal{D}|\mathbf{q}, \mathcal{M})P(\mathbf{q}|\mathcal{M}) d\mathbf{q}, \quad (4)$$

is the marginal likelihood for network structures, which requires the parameters \mathbf{q} to be integrated out. If certain regularity conditions, discussed in Heckerman (1999), are satisfied and the data are *complete* (meaning that we do not have any missing values), the integral in (4) is analytically tractable. Two distribution families \mathcal{F} that satisfy these regularity conditions are the linear Gaussian and the multinomial distribution, both with their respective conjugate prior (Friedman *et al.*, 2000). As opposed to the linear Gaussian distribution, the multinomial distribution can capture non-linear dependence relations and will therefore be applied in the present study. Note, however, that this choice requires a discretization of the data, as illustrated in Figure 5, bottom right, and therefore suffers from a certain information loss.

The closed-form solution to (4) does not imply a straightforward solution to (2). The number of network structures increases super-exponentially with the number of nodes, and the optimization problem is known to be NP-hard (Chickering, 1996). One therefore has to resort to heuristic optimization methods, like hill-climbing or simulated annealing. However, there is reason to question the appropriateness of the learning paradigm based on (2) altogether. For microarray experiments, the data \mathcal{D} are usually sparse, which implies that the posterior probability over structures, $P(\mathcal{M}|\mathcal{D})$, is likely to be diffuse. Consequently, $P(\mathcal{M}|\mathcal{D})$ will not be adequately represented by a single structure, \mathcal{M}^* , and it is more appropriate to sample networks from the posterior probability (3), leading to a collection of networks with high posterior probability, that is, networks that offer a good explanation of the data. Since direct sampling from (3) is impossible due to the intractability of the denominator, one has to resort to a MCMC simulation (Chib and Greenberg, 1995). Given a network structure \mathcal{M}_{old} , a new structure \mathcal{M}_{new} is proposed, with proposal probability $Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})$. This new structure is accepted with the Metropolis–Hastings acceptance criterion (Hastings, 1970):

$$P_{MH} = \min \left\{ 1, \frac{P(\mathcal{M}_{\text{new}}|\mathcal{D})}{P(\mathcal{M}_{\text{old}}|\mathcal{D})} \times \frac{Q(\mathcal{M}_{\text{old}}|\mathcal{M}_{\text{new}})}{Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})} \right\}, \quad (5)$$

where $P(\mathcal{M}|\mathcal{D})$ is given by (3), and the intractable denominator Z cancels out in the ratio. Note that (5) is a generalization of the Metropolis algorithm (Metropolis *et al.*, 1953) for asymmetric proposal probabilities Q . The iteration of this procedure produces a Markov chain that under fairly general conditions converges in distribution to the true distribution (3). In practice, a new network is usually proposed by applying one of the elementary operations shown in Figure 2, and discarding those networks that violate the acyclicity condition. The Hastings ratio, in this case, is given by

$$\frac{Q(\mathcal{M}_{\text{old}}|\mathcal{M}_{\text{new}})}{Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})} = \frac{\mathcal{N}(\mathcal{M}_{\text{old}})}{\mathcal{N}(\mathcal{M}_{\text{new}})}, \quad (6)$$

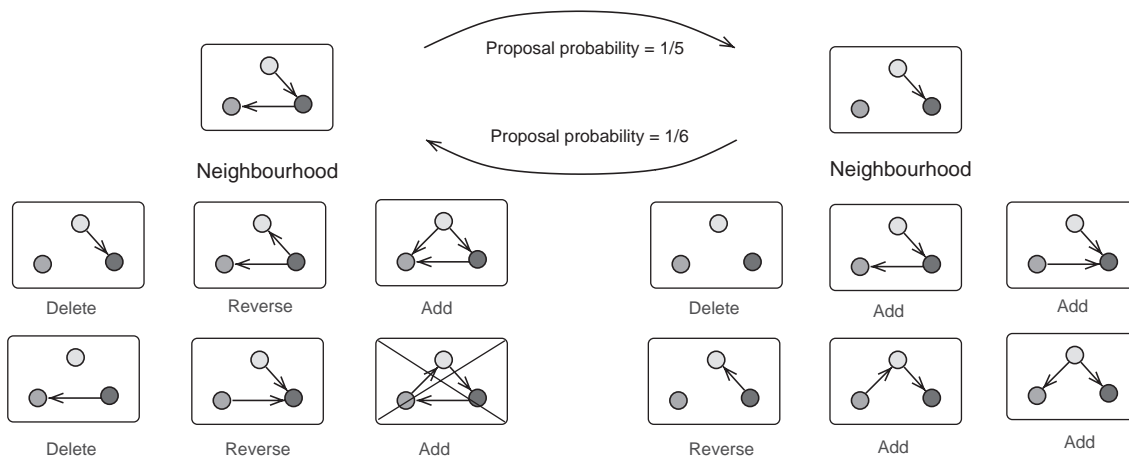


Fig. 2. MCMC proposal moves and Hastings ratio. The figure shows three elementary proposal moves for modifying a network structure by deleting, reversing or adding an edge. Note that the last two operations can cause invalid structures with closed loops, which have to be discarded. The Hastings ratio is given by the ratio of the neighbourhood sizes of the two networks involved in the proposal move, where the neighbourhood is the set of all DAGs that can be reached from the present DAG by applying one of the three elementary proposal moves.

where $\mathcal{N}(\mathcal{M})$ is the size of the neighbourhood of \mathcal{M} , that is, the number of acyclic structures that can be obtained from \mathcal{M} by application of one of the elementary operations of Figure 2.

The approach outlined above has two major limitations. First, several networks with the same skeleton but different edge directions can have the same marginal likelihood $P(\mathcal{D}|\mathcal{M})$, which implies that we cannot distinguish between them on the basis of the data. This *equivalence*, which is intrinsic to *static Bayesian networks* (Chickering, 1995), loses substantial information about the edge directions and thus about possible causal interactions between the genes. The second and more serious restriction is given by the acyclicity constraint, which rules out recurrent structures like those in the top right of Figure 1. Since feedback is an essential feature of biological systems, the usefulness of Bayesian networks for modelling genetic regulatory interactions seems questionable. To proceed, consider Figure 1, top right, which shows a simple network consisting of two genes. Both genes have feedback loops and interact with each other, ruling out the applicability of DAGs. However, interactions between genes are usually such that the first gene is transcribed and translated into protein, which then has some influence on the transcription of the second gene. This implies that the interaction is not instantaneous, but that its effect happens with a time delay after its cause. The same applies to the feedback loops of genes acting back on themselves. We can therefore *unfold* the recurrent network of Figure 1, top right, *in time* to obtain the directed, acyclic network of Figure 1, bottom. The latter is again a proper DAG and corresponds to a *dynamic Bayesian network*. For details, see Friedman *et al.* (1998) and Murphy and Milan (1999, <http://www.ai.mit.edu/~murphyk/Papers/ismb99.ps.gz>). Note that similar unfolding methods have been applied in the study

of recurrent neural networks (Hertz *et al.*, 1991, p. 183). To avoid an explosion of the model complexity, parameters are tied such that the transition probabilities between time slices $t - 1$ and t are the same for all t . The true dynamic process is thus approximated by a homogeneous Markov model. As opposed to Friedman *et al.* (1998), intra-slice connections, that is, edges within a time slice, are not allowed, because this would correspond to instantaneous interactions. Note that a dynamic Bayesian network avoids the ambiguity of the edge directions, discussed above: reversing an edge corresponds to an effect that precedes its cause, which is impossible. The approach has the further advantage of overcoming one of the computational bottlenecks of the MCMC simulations. Recall that the acceptance probabilities (5) of the Metropolis–Hastings sampler depend on the Hastings ratio (6), which implies an acyclicity check for all networks in the neighbourhood of a given candidate network. For networks with many nodes these neighbourhoods and, consequently, the computational costs become prohibitively large, and it is presumably for this reason that a proper MCMC approach was not attempted in Friedman *et al.* (2000) and Pe'er *et al.* (2001) (both papers resort to a frequentist-like approach, where an optimization algorithm—the *sparse candidate algorithm*—is combined with bootstrapping). Since the unfolding process of Figure 1, top right, automatically guarantees acyclicity, the computation of the Hastings ratio becomes trivial for dynamic Bayesian networks, thereby overcoming this major bottleneck.

The main challenge for the inference procedure is that interactions between hundreds of genes have to be learned from short time series of typically only about a dozen measurements. The inevitable consequence is that the posterior distribution over network structures becomes vague, and it is

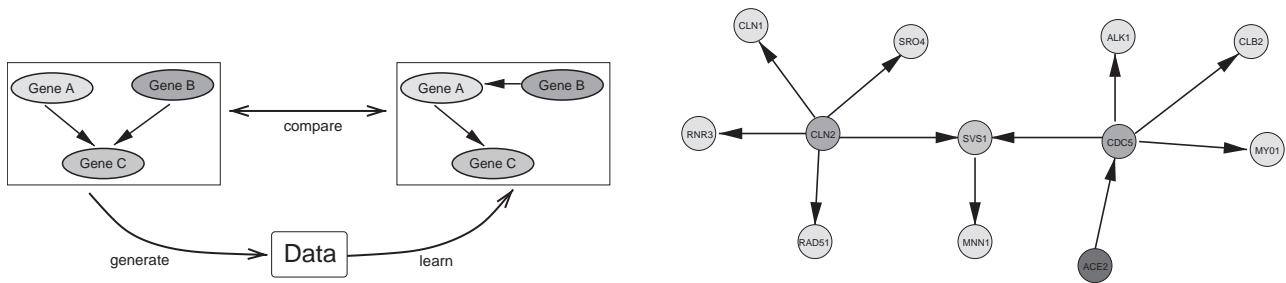


Fig. 3. Synthetic simulation study. Left: Synthetic data are generated from a known Bayesian network. Then, new networks are sampled from the posterior distribution with MCMC and compared with the true network. Redrawn, in slightly modified form, from Smith *et al.* (2002). Right: The structure of the true Bayesian network used in this study is that of a subnetwork of the yeast cell cycle, taken from Friedman *et al.* (2000); 38 unconnected nodes were added, giving a total number of 50 nodes.

the objective of the present study to quantify experimentally how much can be learned from the data in this unfavourable situation. Note that the sparseness of the data implies that the prior $P(\mathcal{M})$ has a non-negligible influence on the posterior $P(\mathcal{M}|\mathcal{D})$ and should therefore be devised so as to capture known features of biological networks. Like most related studies, the present study imposes a limit on the maximum number of edges converging on a node, $FI(\mathcal{M})$, and sets $P(\mathcal{M}) = 0$ if $FI(\mathcal{M}) > \alpha$, for some a priori chosen value of α . This prior incorporates the idea that the expression of a gene is controlled by a small number of active regulators, while, on the other hand, regulator genes themselves are unrestricted in the number of genes they may regulate. The practical advantage of this restriction on the maximum ‘fan-in’ is a considerable reduction of the computational complexity, which improves the convergence and mixing properties of the Markov chain in the MCMC simulation. In the present study, three different threshold values were explored: $\alpha = 2, 3$ and 4.

The simulations reported below were carried out with MATLAB programs that invoke subroutines of the Bayesian network toolbox (Murphy, 2002, <http://www.ai.mit.edu/~murphyk/>). The software is available from the address stated in the abstract.

EVALUATION ON SYNTHETIC DATA

To evaluate the performance of the inference procedure on small data sets, one can proceed as shown in Figure 3, left. Synthetic data, \mathcal{D} , are generated from a known Bayesian network, \mathcal{M}_0 . Then, new networks \mathcal{M}_i are sampled from the posterior distribution $P(\mathcal{M}|\mathcal{D})$. From a comparison between this sample, $\{\mathcal{M}_i\}$, and the true network, \mathcal{M}_0 , we can estimate the accuracy of the inference procedure. Denote by $P(e_{ik}|\mathcal{D})$ the posterior probability for an edge e_{ik} between nodes i and k , which is given by the proportion of networks in the MCMC sample $\{\mathcal{M}_i\}$ that contain this edge. Let $\mathcal{E}(\theta) = \{e_{ik} | P(e_{ik}|\mathcal{D}) > \theta\}$ denote the set of all edges whose posterior probability exceeds a given threshold

$\theta \in [0, 1]$. From this set we can compute (1) the sensitivity, that is, the proportion of recovered true edges, and (2) the complementary specificity, that is, the proportion of erroneously recovered spurious edges. To rephrase this: for a given threshold θ we count the number of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) edges. We then compute the sensitivity = $TP/(TP + FN)$, the specificity = $TN/(TN + FP)$, and the complementary specificity = $1 - \text{specificity} = FP/(TN + FP)$. Rather than selecting an arbitrary value for the threshold θ , we repeat this scoring procedure for several different values of $\theta \in [0, 1]$ and plot the ensuing sensitivity scores against the corresponding complementary specificity scores. This gives the *receiver operator characteristics* (ROC) curves of Figures 4 and 6. The diagonal dashed line indicates the expected ROC curve for a random predictor. The ROC curve of Figure 4, top left, solid line, indicates a perfect retrieval of all true edges without a single spurious edge. In general, ROC curves are between these two extremes, with a larger *area under the ROC curve* (AUROC) indicating a better performance. The true network (or, more precisely, the true inter-slice connectivity of the dynamic Bayesian network) is shown in Figure 3. Two different conditional probability distributions were associated with the edges. *Simulation 1:* noisy regulation according to a binomial distribution with the following parameters: excitation: $P(\text{on}|\text{on}) = 0.9, P(\text{on}|\text{off}) = 0.1$; inhibition: $P(\text{on}|\text{on}) = 0.1, P(\text{on}|\text{off}) = 0.9$; noisy XOR-style co-regulation: $P(\text{on}|\text{on}, \text{on}) = P(\text{on}|\text{off}, \text{off}) = 0.1, P(\text{on}|\text{on}, \text{off}) = P(\text{on}|\text{off}, \text{on}) = 0.9$. *Simulation 2:* Stochastic interaction, where all parameters were chosen at random. These simulations were repeated with both a binomial and a trinomial distribution. Since the results were similar, only the results of the latter will be reported here.

The simulations were repeated for three different time series of length $N = 100, N = 30$, and $N = 7$. In a second series of simulations, 38 redundant unconnected nodes were added to the true network as *confounders*, giving a total of 50 nodes. For each setting, networks were sampled from the

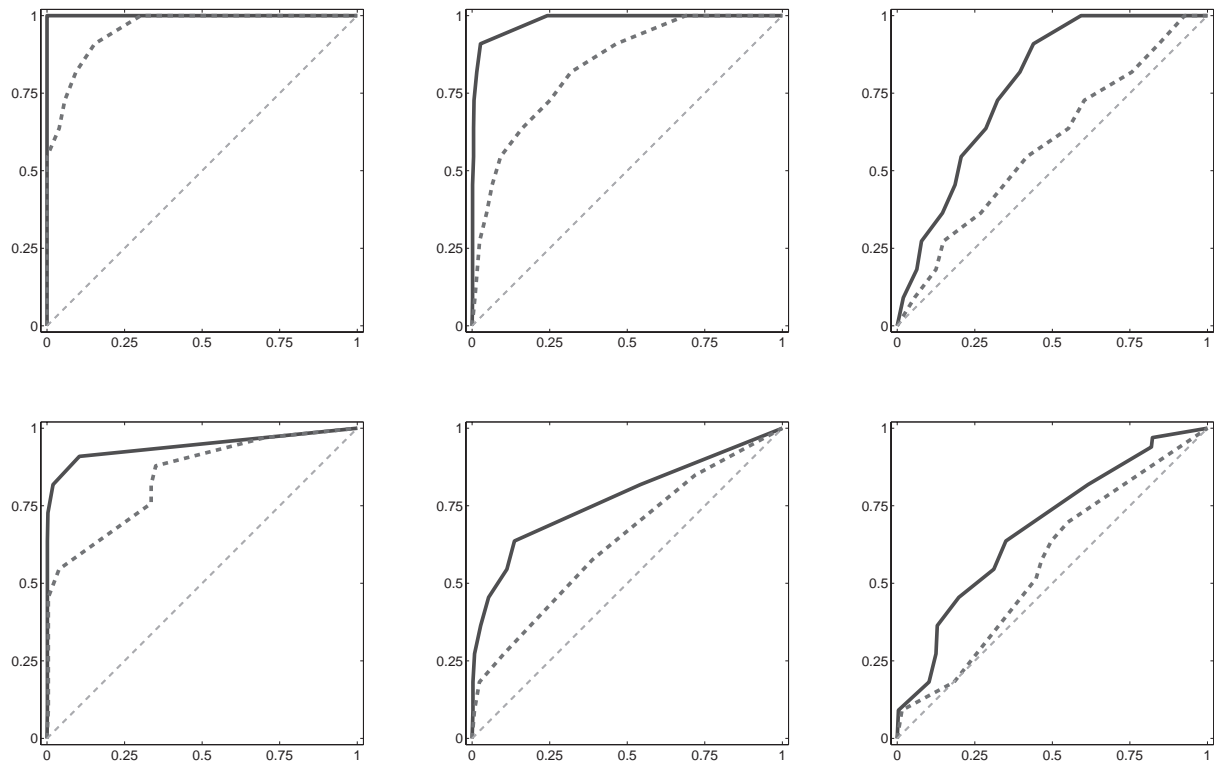


Fig. 4. ROC curves for the synthetic data, averaged over three MCMC simulations, with a maximum fan-in of two. The columns correspond to different training set sizes. Left column: 100; middle column: 30; right column: 7. The rows represent different network sizes. Top row: Networks without redundant nodes. Bottom row: networks with 38 unconnected nodes (50 nodes in total). In each subfigure the sensitivity (proportion of recovered true edges) is plotted against the complementary specificity (proportion of false edges). The thin, diagonal dashed line is the expected ROC curve of a random predictor. The solid thick line shows the ROC curve for simulation 1 (noisy regulation), the dashed thick line that of simulation 2 (stochastic interaction).

posterior distribution $P(\mathcal{M}|\mathcal{D})$ with MCMC, after discarding a sufficiently long equilibration or *burn-in* phase. All simulations were repeated three times for different training data \mathcal{D} , generated from different random number generators.

The results are summarized in Figure 4. The ROC curves show which price in terms of erroneously predicted spurious edges has to be paid for a desired recovery rate of true edges. For instance, for a training set size of $N = 100$ generated from the noisy regulation network without redundant nodes, all true edges can be recovered without incurring any false spurious edges (Fig. 4, top left, solid line). With redundant nodes, we can still recover 75% of the true edges at a zero FP rate, while a price of 25% FPs has to be paid if we want to increase the true prediction rate to 90% (Fig. 4, bottom left, solid line). However, a time series of 100 gene expression measurements is much larger than what is usually available in the laboratory practice, and a realistic experimental situation corresponds much more to Figure 4, bottom right. Here, the inference scheme hardly outperforms a random predictor when the true network has stochastic interactions (thick dashed line). This scenario might be over-pessimistic in that

real gene interactions are not stochastic, but exist for a reason. However, even for a true network with noisy regulation (solid line), one has to pay a price of 25% spurious edges in order to recover 50% of the true edges, or of 50% spurious edges for a true recovery rate of 75%.

EVALUATION ON REALISTIC SIMULATED DATA

Synthetic simulations, as discussed in the previous section, are an important tool to obtain an upper bound on the performance of an inference scheme, that is, they indicate how much can at most be learned about gene interactions for a given training set size. A good performance, however, is no guarantee that we will be able to elicit this amount of information in practice. First, the conditional probabilities resulting from the true, underlying biological process are different from those associated with the edges of a Bayesian network, which means, we have a mismatch between the real data-generating process and the model used for inference. Second, the true continuous signals are typically sampled at discrete time points, which

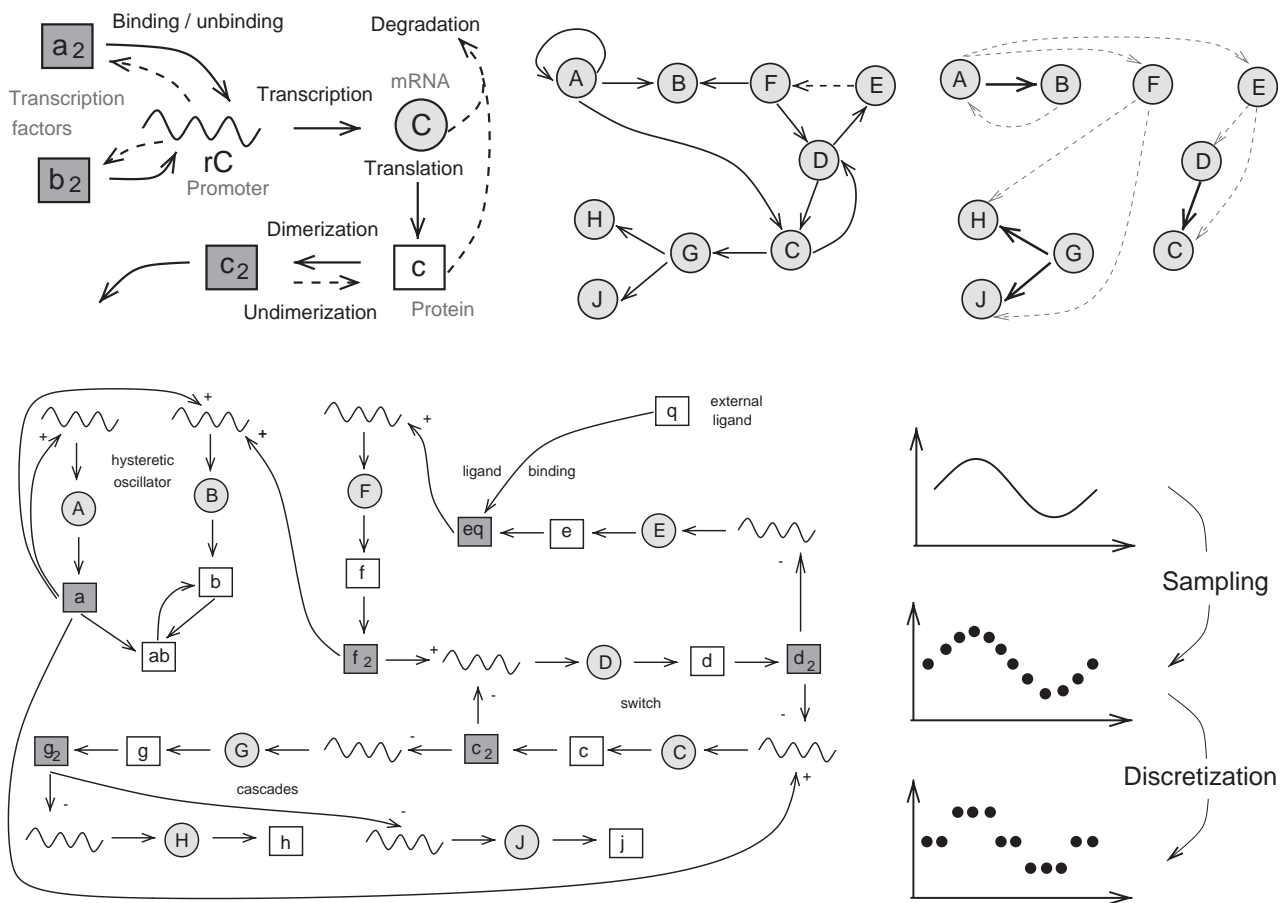


Fig. 5. Realistic simulation. Top, left: Elementary processes. Two transcription factor dimers a_2 and b_2 bind to the *cis*-regulatory site rC in the promoter region upstream of a gene, influencing its rate of transcription. The transcribed mRNA C is translated into protein c , which dimerizes into c_2 to form a new active transcription factor that can bind to other *cis*-regulatory sites. Bottom, left: Realistic biological network composed of these elementary processes, taken from Zak *et al.* (2001) in a slightly modified form. Oscillating lines represent *cis*-regulatory sites in the gene upstream regions, mRNAs are symbolized by upper case letters in circles, proteins are shown by lower case letters in squares. Shaded squares indicate active transcription factors, where in all but two cases this activation is effected by dimerization, and in one case by ligand binding. The symbols + and - indicate whether a transcription factor acts as an activator or inhibitor. The network contains several subnetworks reported in the biological literature. The subnetwork involving mRNAs A and B is a hysteric oscillator. A is translated into protein a , which is an active transcription factor that activates the transcription of B . B is translated into protein b , which forms a dimer ab . This dimerization reduces the amount of free transcription factors a , and oscillations result as a consequence of this negative feedback loop. The subnetwork involving mRNAs C and D is a switch: each mRNA is translated into a transcription factor that inhibits the transcription of the other mRNA, thereby switching the competing path 'off'. Finally, the subnetwork involving mRNA F is triggered by an external ligand, which is needed to form an active transcription factor dimer. Top, middle: Corresponding genetic network, showing only the mRNAs. The dashed line shows an interaction that is triggered by the presence of an external ligand. Bottom, right: Information contained in the true time-dependent mRNA abundance levels is partially lost due to sampling and discretization. Redrawn, in slightly modified form, from Smith *et al.* (2002). Top, right: Genetic network learned from the sampled and discretized data (see text). Solid arrows show true edges and dashed arrows represent spurious edges.

loses information, especially if the sampling intervals are not matched to the relaxation times of the true biological processes. Third, gene expression ratios are typically discretized, which inevitably adds noise and causes further loss of information (see Fig. 5, bottom right).

In an attempt to achieve a more realistic estimation, several authors have tested their inference methods on real microarray

data, testing if a priori known gene interactions (reported in the biological literature) could be recovered with their learning algorithms. While this approach addresses the shortcomings mentioned above, it is prone to the fallacy of judging the performance of a method by the sensitivity score (the TP rate) alone, while the specificity score (the FP rate) is inaccessible, as discussed in the Introduction section.

To proceed, the only satisfactory way is a compromise between the above two extremes and to test the performance of an inference scheme on realistic simulated data, for which the true network is known, but the data-generating processes are similar to those found in real biological systems. The present study applies the model regulatory network proposed by Zak *et al.* (2001), which is shown in Figure 5, and which contains several structures similar to those in the literature, like a hysteretic oscillator (Barkai and Leibler, 2000), a genetic switch (Gardner *et al.*, 2000), as well as a ligand binding mechanism that influences transcription. The elementary processes are shown in Figure 5, top left, and are described by the following system of differential equations, which describe the processes of transcription factor binding, transcription, translation, dimerization, mRNA degradation and protein degradation:

$$\begin{aligned} \frac{d}{dt}[a_2 \cdot rC] &= \lambda_{a_2, rC}^+[a_2][rC] - \lambda_{a_2, rC}^-[a_2 \cdot rC], \\ \frac{d}{dt}[C] &= \lambda_{rC}[rC] + \lambda_{a_2, rC}[a_2 \cdot rC] \\ &\quad + \lambda_{b_2, rC}[b_2 \cdot rC] - \lambda_C[C], \\ \frac{d}{dt}[c] &= \lambda_{Cc}[C] - \lambda_c[c], \quad \frac{d}{dt}[c_2] = \lambda_{cc}^+[c]^2 - \lambda_{cc}^-[c_2]. \end{aligned} \quad (7)$$

Here, the λ_i are kinetic constants, available from the references in Zak *et al.* (2001), t represents time, $[-]$ means concentration, $a_2 \cdot rC$ and $b_2 \cdot rC$ represent transcription factors a_2 and b_2 bound to the *cis*-regulatory site rC , and the remaining symbols are explained in Figure 5. The system of differential equations (7) is taken from chemical kinetics (Atkins, 1986, chapter 28). Consider, for instance, the formation and decay of a protein dimer: $c + c \leftrightarrow c_2$. The forward reaction (formation) is second-order, involving two monomers. Consequently, the time derivative of the dimer concentration, $(d/dt)[c_2]$, is proportional to the square of the concentration of the monomer, $[c]^2$. The reverse reaction (decay) is first order, and $(d/dt)[c_2]$ is proportional to the concentration of the dimer, $[c_2]$. Both processes together are described by the second equation in the last row of (7). The remaining equations can be explained similarly. The system of differential equations for the whole regulatory network of Figure 5, bottom left, is composed of these elementary equations, with three additional but similar equations for ligand binding, ligand degradation, and heterodimerization ($a, b \leftrightarrow ab$). The resulting set of differential equations is stiff and needs to be integrated numerically with a high-order adaptable step-size method (e.g. Runge-Kutta-Fehlberg). Note that except for a , all transcription factors dimerize before they are active, that each gene has more than one rate of transcription, depending on whether promoters are bound or unbound, and that the presence of different time scales makes it representative of a real biological system

and a suitable challenge for the Bayesian network inference algorithm. In contrast to Zak *et al.* (2001), the system was augmented by adding 41 spurious, unconnected genes (giving a total of 50 genes), which were up- and down-regulated at random.

The first experiment followed closely the procedure in Zak *et al.* (2001). Ligand was injected for 10 min at a rate of 10^5 molecules/minute at time 1000 min. Then, 12 data points were collected over 4000 min in equidistant intervals, which, as opposed to Zak *et al.* (2001), also had to be discretized. This discretization was based on the following simple procedure:

$$\begin{aligned} y &\rightarrow 1 && \text{if } y - y_{\min} > \frac{2(y_{\max} - y_{\min})}{3}, \\ y &\rightarrow -1 && \text{if } y - y_{\min} < \frac{y_{\max} - y_{\min}}{3}, \quad \text{and} \\ y &\rightarrow 0 && \text{otherwise.} \end{aligned}$$

The learning algorithm for Bayesian networks was applied as described in the previous sections. Three different structure priors were used, with maximum fan-ins of 2, 3 and 4 edges. The resulting ROC curves are shown in the top of Figure 6. The areas under the ROC curves are small, and the low slope of the ROC curves at the left-hand side of the complementary specificity interval (the x -axis) implies that even the dominant true edges are obscured by a large proportion of spurious edges. This concurs with the findings of Zak *et al.* (2001), who concluded that inferring genetic networks from gene expression data alone was impossible.

The second experiment adopted a sampling strategy different from Zak *et al.* (2001). An analysis of the mRNA abundance levels reveals regular oscillations when the system is in equilibrium. Such signals are known to have a low information content; consequently, it seems to make better sense to focus on the time immediately after external perturbation, when the system is in disequilibrium. The sampler therefore collected 12 data points over a shorter interval of only 500 min immediately after ligand injection, between times 1100 and 1600 min. The resulting ROC curves are shown in the bottom of Figure 6. The areas under the ROC curves have significantly increased, and the larger slope of the curves in the low-sensitivity range implies that the dominant true edges are obscured by far fewer spurious edges.

Recall that in order to obtain a network from the posterior probability on edges, $P(e_{ik}|\mathcal{D})$, one can choose a threshold θ and discard all edges with $P(e_{ik}|\mathcal{D}) < \theta$. Figure 5, top right, shows, for the most restrictive prior (maximum fan-in = 2), the resulting subnetwork of non-spurious genes, where the threshold θ was chosen so as to obtain the same number of edges between non-spurious nodes as in the true network (namely 11). Four true edges, shown as thick arrows, have been recovered (which was consistent in all three MCMC simulations). The probability of finding at least this number

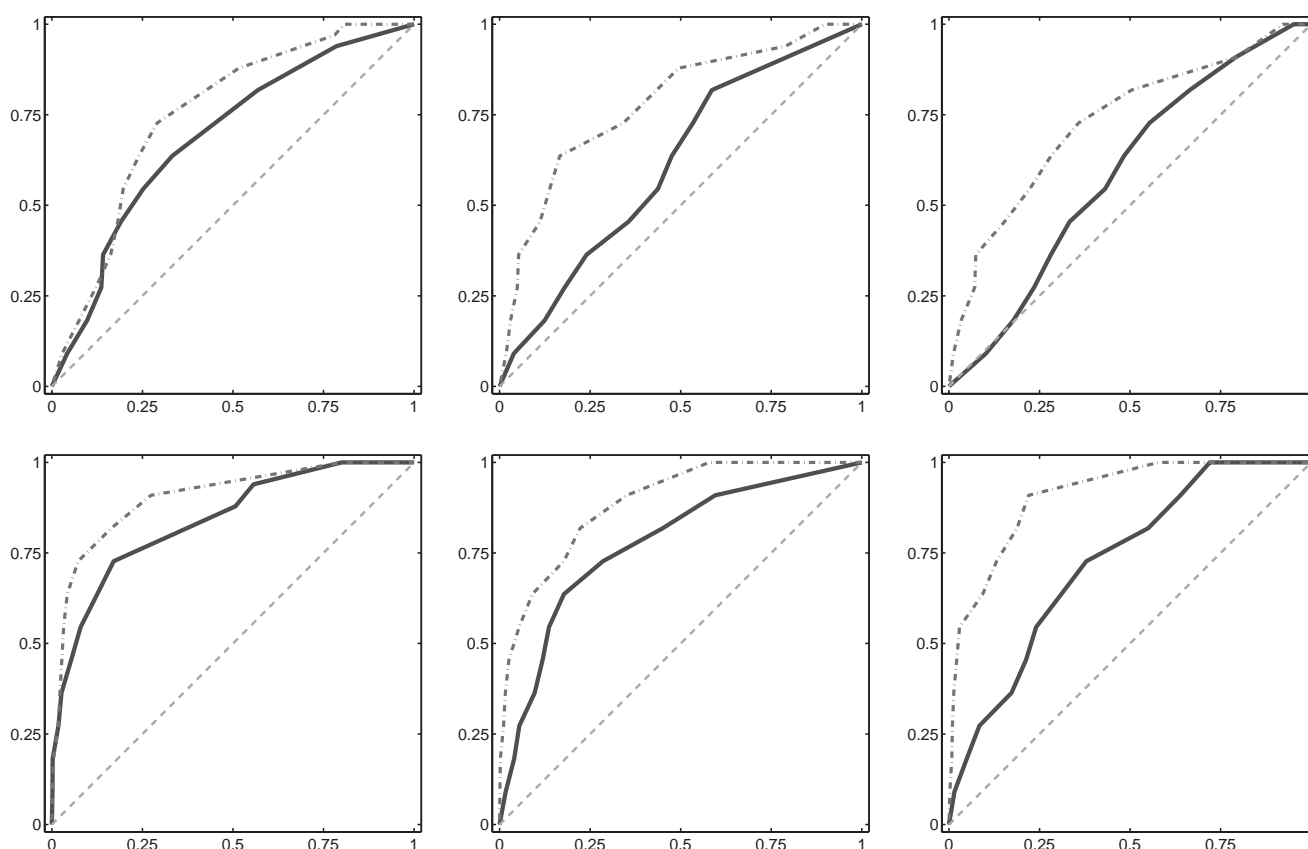


Fig. 6. ROC curves for the realistic simulated data, averaged over three MCMC simulations. The rows represent different sampling periods. Top row: Sampling over a long time interval of 4000 min, which mainly covers the system in equilibrium. Bottom row: Restricting the sampling to a short 500-min time interval immediately after ligand injection, when the system is in a perturbed non-equilibrium state. The columns correspond to different structure priors. Left column: maximum fan-in = 2; middle column: maximum fan-in = 3; right column: maximum fan-in = 4. In each subfigure, the sensitivity (proportion of recovered true edges) is plotted against the complementary specificity (proportion of false edges). The thin, diagonal dashed line is the expected ROC curve of a random predictor. The solid line shows the ROC curve obtained from gene expression data alone, while the dash-dotted line shows the ROC curve obtained when including sequence information.

of true edges by chance is approximately

$$p = \sum_{k=4}^{11} \binom{11}{k} \left(\frac{11}{50^2}\right)^k \left(\frac{50^2 - 11}{50^2}\right)^{(11-k)} \approx 10^{-7},$$

which indicates that the inference procedure has captured real structure in the data. However, this structure is only *local* in nature. The *global* network inferred in this way, shown in Figure 5, top right, shows little resemblance with the true network, depicted in Figure 5, top middle. The ROC curves of Figure 6 reveal that the complete set of true edges can only be recovered at a FP rate of about 75%. Consequently, it is only the most salient local gene interactions that can be inferred from the mRNA abundance data, and the price for this in terms of FP edges can be obtained from the ROC curves of Figure 6. While this number, in absolute terms, is still so large that an experimental verification is indispensable,

Figure 6 also suggests that the search for new genetic interactions preceded and supported by a Bayesian network analysis is significantly more effective than a search from *tabula rasa*. The amount of improvement can, again, be quantified from the ROC curves. Also, note that the most restrictive prior (maximum fan-in = 2) gave consistently the best results, which is in agreement with the true network structure (Fig. 5, top middle). This underlines the obvious fact that, for small data sets, the inclusion of available prior knowledge improves the performance of the inference scheme, and the amount of improvement is quantifiable from the ROC curves.

Given that mRNA abundance levels only convey partial information about a biological regulatory network, it is natural to combine microarray data with genomic and proteomic data. On the genomic side, the identification of certain regulatory sequence elements in the upstream region of a gene (Vilo *et al.*, 2000) can be exploited to predict which transcription factors are most likely to bind to a given promoter, and

this sequence-based estimation can be further assisted experimentally with localization assays (Ren *et al.*, 2000). When location data or known binding motifs in the sequence indicate that a transcription factor, say a , binds to the promoter of another gene, say rB , the respective edge, $A \rightarrow B$ in this case, should be enforced. A straightforward way to incorporate this additional information in the induction was proposed by Hartemink *et al.* (2002): when genomic location data indicate that particular edges corresponding to transcription factor binding reactions should be present, the model prior is modified such that network structures lacking these suggested edges have probability zero. However, location data are usually noisy, and this intrinsic uncertainty is not allowed for by such rigid constraints. An approach that is more involved was proposed by Segal *et al.* (2002), who modelled both the outcome of localization experiments as well as the occurrence of certain binding motifs in the promoter region probabilistically. The method applied in the present study is pitched between these two extremes. It modifies the model prior in a way similar to Hartemink *et al.* (2002), but allows for uncertainty in sequence motif identification and location data. Let $y \rightarrow rX$ denote the event that transcription factor y binds to the promoter r upstream of gene X , and let $B[y]$ represent the set of indicated binding regions for y . Then

$$\frac{P(y \rightarrow rX | r \in B[y])}{P(y \rightarrow rX | r \notin B[y])} = \phi, \quad (8)$$

for some value $\phi > 1$. In words: equation (8) expresses the fact that on identifying a binding region for transcription factor y in the promoter of gene X , this transcription factor is ϕ times as likely to bind to X than in the absence of such an indication. The value of ϕ can be related to the p -value of location data (Ren *et al.*, 2000) and/or the pattern score for binding motifs (Vilo *et al.*, 2000). To quantify the amount of improvement in the induction that can be achieved by combining expression and location data, the previous simulations were repeated for a value of $\phi = 2$ under the assumption that complete location data indicating all regulatory sequence elements are available. The resulting ROC curves are shown as dash-dotted lines in Figure 6, which, as expected, give a consistent improvement on the earlier results obtained from gene expression data alone. Equation (8) is certainly over-simplified. However, the analysis performed here and the comparison of the respective ROC curves allows us to give a rough quantitative estimation of the amount of improvement achievable by merging microarray with sequence and location data.

DISCUSSION

To my knowledge, the first study to test Bayesian networks on gene expression data from a realistic simulation is the paper by Smith *et al.* (2002). The authors simulate a complex biological system at different levels of organization, involving behaviour, neural anatomy and gene expression of

songbirds. They then try to infer the structure of the known true genetic network from the simulated gene expression data with dynamic Bayesian networks. There are two shortcomings of this study, though. First, the genetic network simulator employed by Smith *et al.* (2002) only models genetic activities in a heuristic way; it does not model the molecular biological processes at the different levels of transcription, translation and post-translational modifications. Second, due to constraints imposed on the simulated network structure, the inference procedure was tested for an unrealistically large training set. Smith *et al.* (2002) simulate gene expression time series from the genetic networks in five brain regions of six different birds. Each time series contains 20 samples, which would be an appropriate size representative of real microarray experiments. However, all the $5 \times 6 = 30$ genetic networks are constrained to have the same structure, and they are treated as different samples of the same regulatory network under different conditions. Hence, the authors effectively infer the structure of a single network from a training set of size $6 \times 5 \times 20 = 600$, as stated on page S221 of Smith *et al.* (2002), and this sample is much larger than what is usually available in real microarray experiments. This large size of the training set explains the high inference accuracy reported in Smith *et al.* (2002) (sensitivity: 89%, specificity: 98%), which effectively recovers the true global network (except for a few incorrect edges).

A second study, performed by Zak *et al.* (2001), introduced a more realistic genetic network simulator, where the biological processes at the different levels of transcription, translation and post-translational modifications were modelled with systems of differential equations. However, Zak *et al.* (2001) only applied deterministic linear and log-linear models in their reverse engineering approach, and did not test the more powerful tool of Bayesian networks.

The novel aspect of the present study is to test the reverse engineering of genetic regulatory interactions with dynamic Bayesian networks on simulated expression data that, first, have a sample size representative of real microarray experiments and, second, have been generated from the biologically realistic simulation model of Zak *et al.* (2001).

The present study arrives at the following new findings.

As opposed to the results reported in Zak *et al.* (2001), local structures of the genetic network can, to a certain extent, be recovered. The results depend on the prior used in the Bayesian inference scheme, with a smaller mismatch between reality and prior assumptions obviously leading to better predictions. The amount of this improvement can be quantified from the ROC curves. The results also depend on the sampling scheme. Interestingly, collecting data of the perturbed system in disequilibrium following ligand injection gives better results than when the system is in equilibrium. This finding may suggest that gene expression measurements should, at best, be taken during the relaxation of a biological system after external intervention.

However, the inference results are by far not as positive as reported in Smith *et al.* (2002), which is an immediate consequence of the smaller and more realistic training set size. Note that the global network inferred from the data is meaningless. This shortcoming, illustrated in Figure 5, is a direct consequence of the fact that for sparse data sets \mathcal{D} , the posterior probability on network structures, $P(\mathcal{M}|\mathcal{D})$, is diffuse. Consequently, as discussed earlier by Friedman *et al.* (2000), it is essential to sample networks from $P(\mathcal{M}|\mathcal{D})$, rather than search for a single high-scoring network. From such a sample of structures one can identify edges with high posterior probability, and use them to identify local features and subnetworks with high posterior support. However, a high posterior probability in itself is no guarantee that the respective edge represents a true genetic interaction, partly as a consequence of the fact that the number of spurious interactions, increasing with the square of the number of nodes, substantially outweighs the number of true interactions. Consequently, detected true features of a genetic network are inevitably obscured by a considerable amount of spurious ones. This should be taken as a cautioning note for those trying to back up detected interactions with circumstantial evidence from the biological literature. In practical applications, one has to find a compromise between the number of true edges one wants to detect, and the price in terms of spurious edges one is prepared to pay. The ROC curves shown in the present study do not offer a universal law from which a practical decision support system for the biologist could easily be derived. They do, however, demonstrate empirically how the sensitivity–specificity score ratios vary with the training set size, the degree of inadequacy of prior assumptions, the experimental sampling strategy, and the inclusion of further, sequence-based information. These results can therefore be assumed to shed more light on the important issue of reverse engineering genetic networks from expression data than many previous studies that have tried to evaluate such inference procedures in the absence of known gold standards.

As a final remark, note that gene expression data do not contain information on post-transcriptional and post-translational processes. It would therefore be more appropriate to allow for incomplete observations by including hidden nodes in the Bayesian network architecture. The practical complication of this approach is that (4) has no longer a closed-form solution. In principle one could sample from the joint distribution of model structures *and* their associated parameters with transdimensional reversible jump MCMC (Green, 1995), but convergence and mixing of the Markov chain might become prohibitively slow. Rangel *et al.* (2001) applied a linear state-space model (Kalman smoother) to model the influence of unobserved regulators on gene expression levels. Ong *et al.* (2002) described a dynamic Bayesian network with hidden nodes, whose structure incorporated substantial prior knowledge about operons. I believe that simulation studies, as

presented in the present paper, will be essential to assess the validity of inference procedures for these extended and more complex modelling schemes.

ACKNOWLEDGEMENTS

I would like to thank Chris Glasbey and Peter Ghazal for their interest in this project, and for many stimulating discussions. This work was funded by the Scottish Executive Environmental and Rural Affairs Department (SEERAD). The simulation study made use of the Bayesian network toolbox (Murphy, 2002) and MATLAB programs written by Daniel E. Zak.

REFERENCES

- Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
- Atkins,P.W. (1986) *Physical Chemistry*, 3rd edn. Oxford University Press, Oxford.
- Barkai,N. and Leibler,S. (2000) Circadian clocks limited by noise. *Nature*, **403**, 267–268.
- Chen,T., He,H.L. and Church,G.M. (1999) Modeling gene expression with differential equations. *Pac. Symp. Biocomput.*, **4**, 29–40.
- Chib,S. and Greenberg,E. (1995) Understanding the Metropolis-Hastings Algorithm. *Amer. Statist.*, **49**, 327–335.
- Chickering,D.M. (1995) A transformational characterization of equivalent Bayesian network structures. *Int. Conf. Uncertainty in Artif. Intell. (UAI)*, **11**, 87–98.
- Chickering,D.M. (1996) Learning Bayesian networks is NP-complete. In Fisher,D. and Lenz,H.J. (eds), *Learning from Data: Artificial Intelligence and Statistics*, Vol. 5, Springer, New York, pp. 121–130.
- De Jong,H. (2002) Modeling and simulation of genetic regulatory systems: A literature review. *J. Comput. Biol.*, **9**, 67–103.
- D’haeseleer,P., Liang,S. and Somogyi,R. (2000) Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, **16**, 707–726.
- Eisen,M.B., Spellman,P.T., Brown,P.O. and Botstein,D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, **95**, 14863–14868.
- Friedman,N., Linial,M., Nachman,I. and Pe’er,D. (2000) Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, **7**, 601–620.
- Friedman,N., Murphy,K. and Russell,S. (1998) Learning the structure of dynamic probabilistic networks. In Cooper,G.F. and Moral,S. (eds), *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI)*. Morgan Kaufmann Publishers, San Francisco, CA, pp. 139–147.
- Gardner,T.S., Cantor,C.R. and Collins,J.J. (2000) Construction of a genetic toggle switch in *Escherichia coli*. *Nature*, **403**, 339–342.
- Green,P. (1995) Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination. *Biometrika*, **82**, 711–732.
- Hartemink,A.J., Gifford,D.K., Jaakkola,T.S. and Young,R.A. (2001) Using graphical models and genomic expression data to

- statistically validate models of genetic regulatory networks. *Pac. Symp. Biocomput.*, **6**, 422–433.
- Hartemink,A.J., Gifford,D.K., Jaakkola,T.S. and Young,R.A. (2002) Combining location and expression data for principled discovery of genetic network models. *Pac. Symp. Biocomput.*, **7**, 437–449.
- Hastings,W.K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.
- Heckerman,D. (1999) A tutorial on learning with Bayesian networks. In Jordan,M.I., (ed.), *Learning in Graphical Models, Adaptive Computation and Machine Learning*. MIT Press, Cambridge, Massachusetts, pp. 301–354.
- Hertz,J., Krogh,A. and Palmer,R.G. (1991) *Introduction to the Theory of Neural Computation*. Addison Wesley, Redwood City, CA.
- Kauffman,S.A. (1969) Metabolic stability and epigenesis in randomly connected nets. *J. Theoret. Biol.*, **22**, 437–467.
- Kauffman,S.A. (1993) *The Origins of Order, Self-Organization and Selection in Evolution*. Oxford University Press.
- Krause,P.J. (1998) Learning probabilistic networks. *Knowl. Eng. Rev.*, **13**, 321–351.
- Metropolis,N., Rosenbluth,A.W., Rosenbluth,M.N., Teller,A.H. and Teller,E. (1953) Equation of state calculations by fast computing machines. *J. Chem. Phys.*, **21**, 1087–1092.
- Murphy,K.P. (2002) Bayes Net Toolbox. *Technical Report*, MIT Artificial Intelligence Laboratory.
- Murphy,K.P. and Milan,S. (1999) Modelling Gene Expression Data Using Dynamic Bayesian Networks. *Technical Report*, MIT Artificial Intelligence Laboratory.
- Ong,I., Glasner,J. and Page,D. (2002) Modelling regulatory pathways in *E.coli* from time series expression profiles. *Bioinformatics*, **18**, S241–S248.
- Pearl,J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, CA, USA.
- Pe'er,D., Regev,A., Elidan,G., and Friedman,N. (2001) Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, **17**, S215–S224.
- Rangel,C., Wild,D.L., Falciani,F., Ghahramani,Z. and Gaiba,A. (2001) Modeling biological responses using gene expression profiling and linear dynamical systems. *Proceedings of the 2nd International Conference on Systems Biology*, pp. 248–256.
- Ren,B., Robert,F., Wyrick,J.J., Aparicio,O., Jennings,E.G., Simon,I., Zeitlinger,J., Schreiber,J., Hannett,N., Kanin,E. et al. (2000) Genome-wide location and function of DNA binding proteins. *Science*, **290**, 2306–2309.
- Segal,E., Barash,Y., Simon,I., Friedman,N. and Koller,D. (2002) From promoter sequence to expression: a probabilistic framework. *Res. Comput. Mol. Biol. (RECOMB)*, **6**, 263–272.
- Smith,V.A., Jarvis,E.D. and Hartemink,A.J. (2002) Evaluating functional network inference using simulations of complex biological systems. *Bioinformatics*, **18**, S216–S224. (ISMB02 special issue).
- Vilo,J., Brazma,A., Jonassen,I., Robinson,A. and Ukkonen,E. (2000) Mining for putative regulatory elements in the yeast genome using gene expression data. In Bourne,P.E., Gribskov,M., Altman,R.B., Jensen,N., Hope,D., Lengauer,T., Mitchell,J.C., Scheeff,E., Smith,C., Strande,S. and Weissig,H. (eds), *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*. AAAI, pp. 384–394.
- Zak,D.E., Doyle,F.J., Gonye,G.E. and Schwaber,J.S. (2001) Simulation Studies for the Identification of Genetic Networks from cDNA Array and Regulatory Activity Data. *Proceedings of the Second International Conference on Systems Biology*, pp. 231–238.
- Zak,D.E., Doyle,F.J. and Schwaber,J.S. (2002) Local identifiability: when can genetic networks be identified from microarray data? *Proceedings of the Third International Conference on Systems Biology*, pp. 236–237.