International Journal of Approximate Reasoning 88 (2017) 602–613

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

An empirical comparison of popular structure learning algorithms with a view to gene network inference ☆

Vera Djordjilović [a,*], Monica Chiogna [a], Jiří Vomlel [b]

[a] Department of Statistical Sciences, University of Padova, Italy
[b] Institute of Information Theory and Automation, Czech Academy of Sciences, Czech Republic

ARTICLE INFO

Article history:
Received 28 March 2016
Received in revised form 10 October 2016
Accepted 20 December 2016
Available online 28 December 2016

Keywords:
Bayesian networks
Structure learning
Reverse engineering
DAGs
Gene networks
Prediction

ABSTRACT

In this work, we study the performance of different structure learning algorithms in the context of inferring gene networks from transcription data. We consider representatives of different structure learning approaches, some of which perform unrestricted searches, such as the PC algorithm and the Gobnilp method, and some of which introduce prior information on the structure, such as the K2 algorithm. Competing methods are evaluated both in terms of their predictive accuracy and their ability to reconstruct the true underlying network. A real data application based on an experiment performed by the University of Padova is also considered.

1. Introduction

Genes and proteins do not act in isolation – it is the interactions between them that underlie all major functions of a living cell, from cell differentiation and signal transduction to metabolic processes and cell division. Better understanding of gene networks is one of the central aims of systems biology. Initial approaches to modeling regulatory mechanisms relied on systems of differential equations [5], which are, although very refined, unfortunately limited to small systems about which we already have a hypothesized theory. However, the invention of the technology for measuring abundance of gene transcripts (gene expression), that serve as a proxy for protein abundance, prompted interest in reconstructing the regulatory network from observational data. The idea is to reason backwards: deduce the structure of a complex system from observations of its behavior. This problem has received much attention in the computational biology literature, resulting in a plethora of different models and methods – we refer the interested reader to [2,12,16] for a comprehensive review of the field. Here, we focus on Bayesian networks, a special class of probabilistic graphical models.

A Bayesian network is a statistical model consisting of a Acyclic Directed Graph (DAG) and a family $\mathcal{F}$ of distributions over a set of variables of interest. The graphical structure consists of a set of nodes V and a set of directed edges E. The nodes represent random variables, while edges imply the absence of conditional independence. If there is a directed edge from u to v, we say that u is a parent of v, and denote by $\mathrm{pa}(v)$ a set of parents of v. If we denote by $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)^\top$ the variables of the model, the structure of a DAG is associated with their joint distribution through a factorization property

☆ This work was supported by the Czech Science Foundation through projects 13-20012S and 16-12010S.
* Corresponding author.
E-mail address: djordjilovic@stat.unipd.it (V. Djordjilović).

http://dx.doi.org/10.1016/j.ijar.2016.12.012
0888-613X/

$$f(\boldsymbol{x}) = f(x_1, x_2, \ldots, x_p) = \prod_{i=1}^{p} f\left[x_i \mid \mathrm{pa}(x_i)\right],$$

where f stands for a generic probability distribution function. When X_i, $i = 1, 2, \ldots, p$ are discrete, we usually assume that all conditional distributions on the right hand side are multinomial, giving rise to a joint multivariate multinomial distribution. When they are continuous, we usually assume that all conditional distributions on the right hand side are normal, giving rise to a joint multivariate normal distribution $\mathrm{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Here $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)^\top$ represents the mean vector and $\boldsymbol{\Sigma} = [\sigma_{rs}]$, $r, s = 1, \ldots, p$, is the covariance matrix encoding the structure of the network. In fact, the model can be equivalently represented in the recursive form

$$\boldsymbol{X} = \boldsymbol{\alpha} + \boldsymbol{B}\boldsymbol{X} + \boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)^\top$ is the base level, $\boldsymbol{B} = [b_{rs}]$, $r, s = 1, \ldots p$ is a matrix of regression coefficients and $\boldsymbol{\epsilon} \sim \mathrm{N}_p(0, \mathrm{diag}\left(\theta_1, \ldots, \theta_p\right))$ is the random disturbance. If variables are topologically ordered with respect to the DAG, $\boldsymbol{B}$ will be strictly upper triangular and there is a simple relation between the two parameterizations being $\boldsymbol{\mu} = (\boldsymbol{I} - \boldsymbol{B})^{-1}\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma} = (\boldsymbol{I} - \boldsymbol{B})^{-1}\mathrm{diag}\left(\theta_1, \ldots, \theta_p\right)[(\boldsymbol{I} - \boldsymbol{B})^{-1}]^{\mathsf{T}}$, for more details see [24].

The problem of learning the structure of a Bayesian network from realizations of the random vector $\boldsymbol{X}$ is conceptually simple but computationally very complex. Many algorithms, with optimal asymptotic properties, have been proposed [see for instance [6,28]]. The problem encountered when applying these algorithms to gene networks is the small sample size – very often the number of considered genes p exceeds the number of statistical units n. This typical property of biological datasets poses serious limitations for structure learning, explored in detail in [18]. To attenuate this problem, several authors exploited the expected sparsity of biological networks [4,26]. Another possible remedy is to use other sources of information and include them in the learning process. This seems reasonable, since technological advances seen in the last two decades drastically reduced experimental costs and made measurements of biological activity more readily available.

Much of the experimentally obtained knowledge is stored in online public databases. One instance is represented by pathway diagrams, which are elaborate diagrams featuring genes, proteins and other small molecules, showing how they work together to achieve a particular biological effect. From a technical point of view, they are networks and can be represented through a graph where genes and their connections are, respectively, nodes and edges. We will study the effect of including some of the information they carry into the learning process. More specifically, for a set of genes of interest we will specify a topological ordering according to the pathway information, and then pass this ordering to the algorithms that use prior information.

In this empirical comparison, we consider representatives of different structure learning approaches, such as the constraint based PC algorithm [28], the exact Gobnilp method [8] and the score based K2 algorithm [6]. The software used is a combination of tools available in Hugin [21], Gobnilp [8] and R [23]. We perform an extensive simulation study, considering two data generating mechanisms, with the aim of verifying whether the approaches that include prior information, such as K2, perform better than those that rely on data only. In addition to simulated data, we also consider real data from the *Drosophila melanogaster* experiment. In this experiment, performed by the University of Padova [11], researchers measured the expression of genes participating in the WNT signaling pathway in a fruit fly.

The outline of the paper is as follows. In Section 2, we introduce algorithms considered in this work, we propose the data driven categorization procedure for continuous gene expression measurements, and describe the method we used to evaluate the performance of algorithms. Details and results of the simulation studies are given in Section 3, while the evaluation on the real dataset is given in Section 4. Conclusions, some limitations of the present work and future perspectives are given in Section 5.

2. Structure learning algorithms

Structure learning algorithms can be roughly divided into two groups: (1) the score based approaches that search the space of possible structures to find the one that maximizes the score (reflecting the goodness of fit and model parsimony), and (2) the constraint based approaches, that test a large number of conditional independencies between variables and then find the structure that best fits the set of inferred relations. Computational complexity of both approaches is prohibitive for all but small problems, and so some strategy for restricting the search space and reducing the number of performed tests is usually adopted. In that case, one does not have a guarantee that the inferred structure will be globally optimal, but must rely on proven asymptotic validity. Recently, a number of approaches to structure learning that do not rely on a search strategy, but explore the entire space of DAGs, have been proposed. Two major directions are dynamic programming and integer linear programming. For the former, see for instance [20]. We consider a representative of the latter: the approach introduced in [7], where the problem of structure learning is translated into an optimization problem with a linear objective function and a set of linear constraints (as well as integrality constraints on the variables).

As already stated, in this empirical study we consider representatives of different structure learning strategies: a number of variants of the PC algorithm [28], the K2 algorithm [6] and the exact Gobnilp method [8]. Of the examined approaches, the K2 algorithm and all modifications of the K2 algorithm here considered, include the prior information. The prior information is in the form of the topological ordering of the studied genes. In the simulation study, we specify the topological ordering according to the true underlying graph. In the real study, we relied on public databases of biological knowledge. In particular,

we used the WNT pathway of the KEGG database to construct a DAG for the set of genes under study, from which we, then, derived a topological ordering. The topological ordering is in general not unique. The consequences of its non-uniqueness will not be discussed here.

Our choice of a specific algorithm as a representative of its class was motivated either by its widespread use and popularity (PC), its implementation and software availability (Gobnilp) and the possibility to include prior information without the need to define the prior distribution over the space of possible structures (K2). To summarize, in this empirical study, we consider the following options.

PC	The PC algorithm using χ^2 test of independence at 5% significance level.
PC20	The PC algorithm using χ^2 test of independence at 20% significance level.
K2	The original K2 algorithm.
K2-BIC	A modified K2 algorithm, where the criterion used to score competing DAGs is BIC, while the search strategy remains the one step greedy search.
G-BIC	The Gobnilp algorithm with the BIC scoring criterion.
G-BICm	The Gobnilp algorithm with a modified BIC criterion (the penalty term is multiplied by a factor of 10^{-3}).
G-BICl	The Gobnilp algorithm with a modified BIC criterion (the penalty term is multiplied by 10^{-9}). This implementation efficiently finds the model with the least number of parameters among all those maximizing the log likelihood function.
CK2	The CK2 algorithm proposed in [11]. The only algorithm in this study that is applied to continuous measurements.

It was brought to our attention, by one of the Reviewers, that Gobnilp, through constraints regarding forbidden arrows, provides a possibility to impose the topological ordering. Clearly, when the ordering is specified correctly, including prior information should improve the performance of the algorithm, in terms of distance to the true structure. For this reason, we report complete results regarding the performance of Gobnilp using prior information in the Appendix, whereas in the following, we restrict our attention to Gobnilp variants with no ordering constraint described above.

2.1. Categorization of expression measurements

Most structure learning algorithms make use of categorical variables, while gene expressions are quantitative measurements, usually continuous. In the work that first introduced the idea of using DAGs for representing gene regulatory networks, [15] considered both discrete and continuous models. It is clear that discrete models attenuate the effect of the technical variability, but might lead to information loss, and results that are sensitive to the choice of the categorization procedure. Continuous models incur no information loss, but are incapable of capturing non-linear relationships between genes. In particular, combinatorial relationships (one gene is over-expressed only if a subset of its parents is over-expressed, but not if at least one of them is under-expressed) can be modeled only with a discrete Bayesian network. The two approaches thus seem complementary and we believe that both can help researchers obtain the biologically relevant results, at least as a means of postulating testable scientific hypotheses.

When the goal of categorization is to obtain categories which are meaningful from the biological perspective, one would ideally have the control group (a previous experiment) which would serve as a reference for comparison [15]. When control data are not available, we propose to perform categorization based solely on data at hand. It is assumed that genes can take only a few functional states, for example “under-expressed,” “normal,” and “over-expressed.” The actual measurements depend on these functional states and the amount of biological variability and technical noise. A plausible model for such data is a mixture of K normal distributions, each centered at one of the K functional states

$$X_i \sim \sum_{k=1}^{K} \tau_{ik} \mathrm{N}(\mu_{ik}, \sigma_{ik}^2), \quad i = 1, \ldots, p,$$

where X_i is an expression of the considered gene, μ_{ik} and σ_{ik}^2 are parameters corresponding to the k-th functional state, τ_{ik} the probability that an observation belongs to the k-th component ($\tau_{ik} \geq 0, \sum_{k=1}^{K} \tau_{ik} = 1$) and p is the number of considered genes. However, it is not always plausible to assume that all K states are present in a single experiment, for example, certain genes remain normally expressed in a wide range of conditions, others can only be downregulated, etc. This led us to propose a data driven approach to categorization: a number of components, that can vary from one (corresponding to a gene with only one observed state) to K (all functional states are present in the data) is estimated from the data for each gene independently. The assumed model for the i-th gene is thus

$$X_i \sim \sum_{k=1}^{\hat{K}_i} \tau_{ik} \mathrm{N}(\mu_{ik}, \sigma_{ik}^2), \quad i = 1, 2, \ldots, p,$$

where $\hat{K}_i$ is the estimated number of components for the i-th gene, τ_{ik} are, as before, the weights of individual components, μ_{ik}, σ_{ik} are component specific parameters. The approach that simultaneously estimates the number of components in the

mixture and parameters pertaining to different components and then classifies each observation according to the estimated model used in model based clustering was introduced by [14]. We used the implementation in the R package `mclust` [13]. In what follows, we will denote $Y_i = (Y_{i1}, \ldots, Y_{i\hat{K}_i})$ the variable obtained from X_i through the proposed categorization, where $Y_{ij} = 1$, if X_i falls to category j, and zero otherwise.

2.2. Evaluation of results

It is well known that several different Bayesian networks can model the same conditional independence model and are thus indistinguishable from the statistical viewpoint. In such a case, it is said that such Bayesian networks belong to the same equivalence class. A typical representative of an equivalence class is the essential graph (also known as PDAG, partially directed acyclic graph). It is a mixed graph with both directed and undirected edges. Undirected edges can be ordered in either direction as long as they do not create a new immorality – for details see, e.g., [30]. In order to perform a fair comparison of the ability of the considered algorithms to reconstruct the underlying graphical structure we rely on structural Hamming distance [1] that compares two essential graphs. Structural Hamming distance between two essential graphs $\mathcal{G}$ and $\mathcal{H}$ on p variables is defined as the number of edges that do not coincide in $\mathcal{G}$ and $\mathcal{H}$:

$$SHD(\mathcal{G}, \mathcal{H}) = \#\left\{\{i, j\} \in V^2 \mid \mathcal{G} \text{ and } \mathcal{H} \text{ do not have the same type of edge between } i \text{ and } j\right\},$$

where $\{i, j\}$ is considered an unordered pair and we allow for four different relations between i and j: no edge, an undirected edge, an edge directed from i to j and an edge directed from j to i.

Methods that use prior information in the form of the topological ordering do not suffer from statistical indistinguishability – they return a unique DAG, since the ordering provides sufficient information to direct all the inferred edges. In that case, we consider two further measures of the ability to reconstruct the underlying graphical structure from observations, the PPV that stands for Positive Predictive Value and is defined as $TP/(TP + FP)$; and Sensitivity, defined as $TP/(TP + FN)$, where TP (true positive), FP (false positive), and FN (false negative) refer to the inferred edges.

To evaluate the predictive accuracy, we adopt a "leave-one-out" approach, where in each step the chosen learning algorithm is applied to the data from which the single observation j has been removed. In the second step, the removed observation is predicted by estimating the value of each variable given the values of all other variables.

To measure the distance between the observed value and the predicted value for variable Y_i, we used the Brier score, introduced in [3]. Let ${}_jy_i = ({}_jy_{i1}, \ldots, {}_jy_{i\hat{K}_i})$ be the observed value of variable Y_i in the jth observation, $j = 1, \ldots, n$. The Brier score, for individual j and variable i, is defined as

$${}_jb_i = \frac{1}{2}\sum_{k=1}^{\hat{K}_i}({}_j\hat{\pi}_{ik} - {}_jy_{ik})^2, \tag{2}$$

where ${}_j\hat{\pi}_{ik}$ is the predicted probability that Y_i falls into the category k. The Brier score measures the squared distance between the forecast probability distribution and the observed value. It can assume values between 0 (the perfect forecast) and 1 (the worst possible forecast).

We measure the overall predictive accuracy with the score

$$B = \sum_{j=1}^{n}\sum_{i=1}^{p}{}_jb_i. \tag{3}$$

Obviously, algorithms having a lower score are preferred.

We compare algorithms designed for categorical and continuous data. The learning algorithms that work with continuous data produce predictions on the continuous scale. In order to make them comparable with categorical predictions, we combine discriminant analysis with the proposed categorization procedure. We classify continuous predictions into one of the gene specific components estimated in the initial categorization. More precisely, we apply the discriminant analysis to the prediction ${}_j\hat{X}_i$; the output is the estimated vector of probabilities $({}_j\hat{\pi}_{i1}, \ldots, {}_j\hat{\pi}_{i\hat{K}_i})$ that ${}_j\hat{X}_i$ falls into associated categories. We can then plug this vector in the expression for the Brier score (2).

3. Simulation studies

To avoid a dependence of our conclusions on characteristics of individual graphs, we randomly generated 10 DAGs on 10 nodes. We achieved this by randomly generating 10 adjacency matrices $\boldsymbol{A} = [a_{rs}]$, $r, s = 1, \ldots, 10$. In the first step, we set a sparsity parameter in the interval $(0.3, 0.5)$ for each graph and fixed a topological ordering. In the second step, we sampled an observation from a Bernoulli variable with the chosen sparsity parameter for each plausible edge (corresponding to the upper triangular part of the adjacency matrix) to obtain an adjacency matrix uniquely determining the corresponding DAG. The density of generated DAGs, computed as the ratio of present and potential links, ranges from 0.12 to 0.34, with the average being around 0.21.

When generating observations, our intention was to mimic the situation in which each gene's underlying latent states (low and high expression) are affected and, to a certain level, "masked" by some biological and technical noise. We considered two different data generating mechanisms: the first one is a Gaussian Bayesian network and the second one is a discrete Bayesian network affected by random fluctuation. In both cases, observations are continuous and before passing them to the algorithms using categorical variables, we performed categorization as described in 2.1. Namely, we performed data driven categorization, where each variable was allowed to have either two or three categories, depending on the model fit. In the second study, we knew that there were two underlying states, but we estimated the number of states from data so as to approach the conditions of a real study as close as possible.

Simulation study 1

We considered a Gaussian Bayesian network whose parameters, given in (1), were generated randomly in the following way

$$\begin{aligned}
&\alpha_r \sim \mathsf{U}(-10, 10);\\
&\theta_r \sim \mathsf{U}(0.5, 1);\\
&b_{rs} \sim \begin{cases} \mathsf{U}[(-1,-0.2)\cup(0.2,1)] & \text{when } a_{rs}=1,\\ 0 & \text{otherwise;}\end{cases}
\end{aligned}$$

where $\boldsymbol{\alpha} = [\alpha_r]$ is the base level, $\mathsf{diag}(\theta_r)$ is the disturbance variance, $\boldsymbol{B} = [b_{rs}]$, $r, s = 1, \ldots, p$ is the graphical structure and U denotes a uniform distribution.

Simulation study 2

In the second study we considered a discrete Bayesian network in which each variable can take two values, leading to the multivariate Bernoulli joint distribution. The basic idea behind this model is that for each gene to be expressed the related genes (parent nodes in the Bayesian network) have to be in certain states (i.e., either expressed or not expressed). Thus, the relation is specified by a logical "and" function of its parents (possibly negated). However, we allow some noise to be present in the relation. Even if a parent is not in the required state, there is a non-zero probability that the modeled gene will be expressed. We will refer to this probability as inhibitory probability, since it inhibits the expected effect of the parent on its child. This model is called "noisy and". It is an example of a canonical model [10] or an ICI model [17].

We assumed that the conditional distribution of each variable is a "noisy and"

$$\mathsf{P}[Z_r = 1 | \mathsf{pa}(Z_r)] = \pi_r \prod_{Z_s \in \mathsf{pa}(Z_s)} p_{rs}^{(1-Z_s)}, \tag{4}$$

where π_r and p_{rs} are leaky and inhibitory probabilities, respectively. In this formulation, the effect of each parent on its child is positive; if the effect of Z_s on Z_r is negative or inhibitive, instead of $p_{rs}^{(1-Z_s)}$ in (4), we have $p_{rs}^{Z_s}$. This model is thus determined by the vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_p)^\top$ of leaky probabilities, a matrix $\boldsymbol{P} = [p_{rs}]$, $r, s = 1, \ldots, p$, of noise probabilities and an influence matrix $\boldsymbol{Q} = [q_{rs}]$, stating the direction of a parent–child effect. We generated these parameters in the following way

$$\begin{aligned}
\pi_i &\sim \begin{cases} \mathsf{U}(0.1, 0.9) & \text{if } Z_i \text{ has no parents,}\\ \mathsf{U}(0.85, 0.98) & \text{otherwise;}\end{cases}\\
q_{rs} &= \begin{cases} 1 & \text{with } \mathsf{P}=0.7 \quad \text{when } a_{rs}=1,\\ -1 & \text{with } \mathsf{P}=0.3 \quad \text{when } a_{rs}=1,\\ 0 & \text{otherwise;}\end{cases}\\
p_{rs} &\sim \begin{cases} \mathsf{U}(0.05, 0.2) & \text{when } a_{rs}=1,\\ 0 & \text{otherwise.}\end{cases}
\end{aligned}$$

Then, a mild jittering is performed on the binary variables to vaguely mimic the technical artifacts related to biological experiments. We therefore add to that each variable some random noise, so that the actual observations are of the form $\boldsymbol{X} = \boldsymbol{Z} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathsf{N}(0, \sigma^2 \boldsymbol{I})$ (we set $\sigma = 0.25$).

Experiments

For each graph of the 10 DAGs, we randomly generated 1000 datasets with the following sample sizes $n = 20, 30, 50, 100, 500$. We then applied structure learning approaches to infer the graphical structure from observations. For each inferred essential graph, we computed its structural Hamming distance from the essential graph of the "true" underlying graph used to generate data. Finally, we pooled results from 10 graphs to obtain an estimate of the average Hamming distance. Given that the results of the approaches of the same type (such as PC and PC20; and K2 and K2-BIC) have nearly

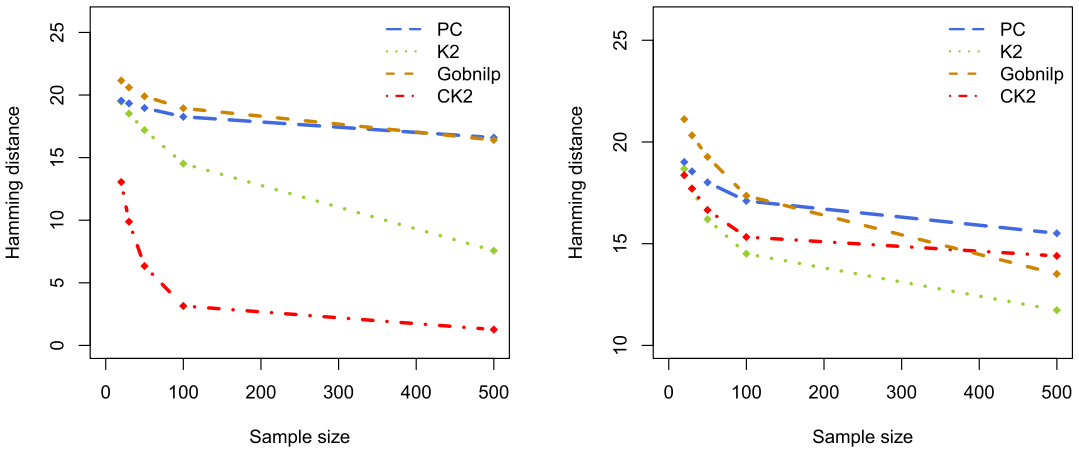


Fig. 1. Simulation study: Pooled Structural Hamming distance in Study 1 (left) and Study 2 (right).

Table 1
Simulation studies: Mean Positive predictive value (and its standard deviation) in Study 1 (left) and Study 2 (right) in percentage.

n	K2		K2-BIC		CK2		K2		K2-BIC		CK2	
20	61%	(5)	75%	(7)	85%	(3)	59%	(4)	71%	(6)	73%	(6)
30	66%	(6)	80%	(5)	88%	(2)	67%	(4)	76%	(4)	74%	(4)
50	72%	(4)	77%	(4)	90%	(2)	76%	(3)	78%	(3)	77%	(3)
100	80%	(3)	82%	(3)	93%	(1)	81%	(3)	80%	(3)	83%	(3)
500	87%	(3)	81%	(3)	96%	(1)	89%	(2)	82%	(2)	80%	(2)

Table 2
Simulation studies: Mean Sensitivity (and its standard deviation) in Study 1 (left) and Study 2 (right) in percentage.

n	K2		K2-BIC		CK2		K2		K2-BIC		CK2	
20	32%	(3)	19%	(2)	51%	(3)	30%	(2)	15%	(2)	14%	(3)
30	35%	(3)	25%	(3)	63%	(3)	36%	(4)	28%	(4)	27%	(4)
50	41%	(3)	35%	(3)	77%	(3)	46%	(5)	41%	(5)	37%	(5)
100	49%	(3)	48%	(3)	89%	(2)	48%	(6)	49%	(6)	41%	(6)
500	76%	(3)	80%	(3)	96%	(2)	68%	(7)	69%	(7)	63%	(8)

identical results, we show one representative per group, namely PC, G-BIC and K2 (complete results, with estimated standard errors, can be found in Appendix A). Results for the two simulation studies are shown in Fig. 1. We see that in the first study the best performing algorithm is CK2 followed by K2. The algorithms that do not use any prior information have significantly higher Hamming distance. In the second study different algorithms give more similar results. For lower sample sizes CK2 still performs very well, while for higher sample sizes K2 reaches a significantly lower mean Hamming distance. Furthermore, of all considered algorithms, the average Hamming distance for K2 seems to decrease at the highest rate with increasing sample size.

For algorithms that use topological ordering and thus return a unique DAG, we considered also mean PPV and Sensitivity to get a better understanding of their ability to learn the structure from data. The results are shown in Tables 1 and 2. We see, in particular, that the good performance of CK2 in the first study is due to the high values of both positive predictive value and sensitivity. Its performance in the second study, where the setting was not as favorable, deteriorated. K2, on the other hand, has a very similar performance in both studies, suggesting higher robustness with respect to a data generating mechanism.

As an illustration of the performance of considered approaches in reconstructing the "true" DAG, we show one example of an original and a reconstructed network in Fig. 2. Alongside the essential graph of the "true" DAG used to simulate data, there is an essential graph with the adjacency matrix computed as the arithmetic average of adjacency matrices of essential graphs reconstructed by the G-BIC algorithm (Gobnilp optimizing the BIC criterion) from $n = 500$ observations in 1000 simulations. The edges in which the graphs differ are dashed. Note that they differ in five edges. The average graphs for other algorithms differ from the original in nine edges (CK2 and K2-BIC). For K2 the average graph was equivalent to the original one.

Next, we look at predictive accuracy of considered algorithms. Here, we restricted our attention to the smallest sample size ($n = 20$) as it is the situation most relevant to our field of application. Furthermore, it gives us the opportunity to compare obtained results to those in the real application described in Section 4, since the ratio p/n is approximately the same. Therefore, for each of the 10 DAGs and 1000 generated datasets of size $n = 20$, we computed the B score, shown

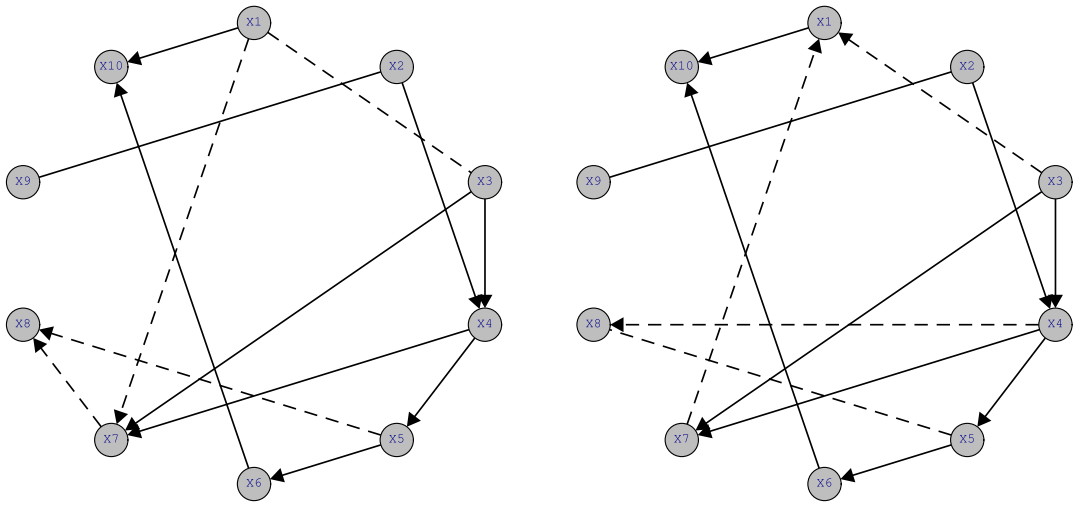


Fig. 2. Study 2: The essential graph of one of the 10 DAGs used to simulate data (left) and the essential graph for the adjacency matrix computed as the arithmetic average of adjacency matrices of essential graphs reconstructed by the Gobnilp algorithm (right).

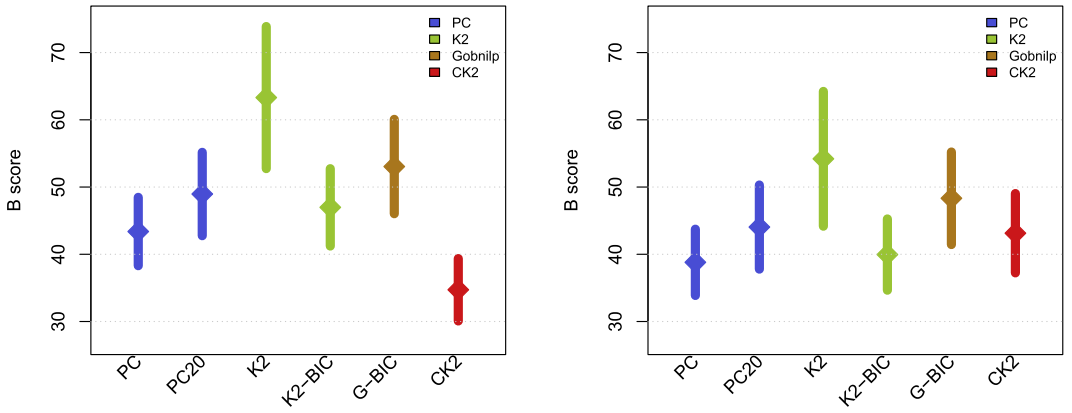


Fig. 3. Simulation studies: mean value of the B score and its approximated 95% confidence interval in Study 1 (left) and Study 2 (right).

in (3), following the "leave-one-out" approach, as described in 2.2. In the end, we performed a random effects meta analysis (assuming that the B score is approximately normally distributed) to combine results for different graphs. The mean B score and its approximated 95% confidence interval in the two studies are shown in Fig. 3. In the first study, CK2 reached the lowest B score, followed by PC and K2-BIC. In the second study, the performance of CK2 is slightly worse, while PC and K2-BIC remain the leading twosome.

4. *Drosophila melanogaster* experiment

The experimental data from the *Drosophila melanogaster* experiment performed by the University of Padova [11] consist of 28 observations of 12 genes. All measured genes belong to the WNT signaling pathway involved in embryonic development of the fruit fly. DAG derived from this pathway is shown in Fig. 4. Two of the genes, *arm* and *rok*, were excluded from the analysis since in the categorized dataset they assumed just one value. The topological ordering of this DAG was passed to the methods that include prior information (K2, K2-BIC and CK2). Other methods rely on data only.

Fig. 5 shows the B score for each of the considered methods. Full (complete) DAG and empty (no arrows) DAG were added for reference. Here, K2 reaches the minimal B score, followed by the Gobnilp's likelihood method G-BICl. The K2 algorithm with the BIC score, K2-BIC, together with the remaining Gobnilp methods, G-BICm and G-BIC, also perform reasonably well with a slightly inferior score with respect to the leading twosome. On the other hand, the PC algorithm gives significantly less accurate predictions. The CK2 algorithm seems to fail in this case. Its B score is almost comparable to the one of the full graph (Full). It is interesting to note that of the two methods on categorized variables using the BIC score, K2-BIC and G-BIC, it is the former that minimizes the B score. Since Gobnilp finds globally optimal structures, while K2-BIC uses the ordering of variables – and thus might suffer from misspecification – this implies that WNT pathway provides

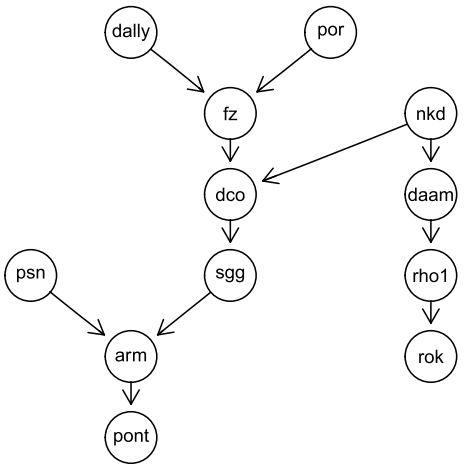


Fig. 4. *Drosophila melanogaster* experiment: The DAG derived from a diagram representing WNT signaling pathway in fruit flies.

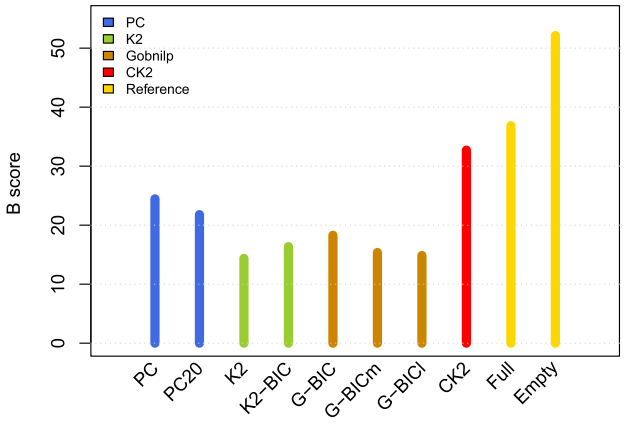


Fig. 5. *Drosophila melanogaster* experiment: B score of different algorithms.

enough accurate information to be used in the learning process and to provide better predictions. To test this hypothesis, we generated 20 random orderings and passed them to the K2 algorithm. None of the twenty computed scores (not reported here) is lower than that determined by pathway, providing support for the practice of using the prior information in the form of a topological ordering.

The right hand side plot in Fig. 6 shows how the B score deteriorates with the addition of arrows to the optimal structure found by K2. Here, the B score is a function of the number of arrows present in the graph. It starts from the K2 structure, containing 15 arrows, and ends with the full graph, containing 66 arrows. Structures in-between are obtained sequentially, by randomly adding a single arrow to the current structure. Obviously, the order of addition of arrows plays a role, and thus this is only one possible way in which the score might evolve between the two extreme points. Nevertheless, the increasing trend of the dependence is informative and independent of the order of arrow inclusion.

One of the reasons behind the success of the K2 algorithm might also be that it identifies DAGs with a relatively high number of edges. To examine this possibility, we computed the average size of the Markov blanket for considered methods. We recall that the Markov blanket of a given node in a DAG consists of the node's parents, children and other parents of its children. It is the set of variables which shields the given variable from the rest of the network. The results are reported in the table shown in the left panel of Fig. 6. We see that K2 indeed has a comparatively large average Markov blanket size, but it is second to the Gobnilp's likelihood method. The ranking of methods with respect to their prediction accuracy suggests therefore that the density of the graphs inferred by K2 is not the only reason for its good performance.

5. Discussion

In this work we performed an extensive empirical study of popular structure learning algorithms in the highly specific setting of inferring gene networks. This area is atypical in that it usually involves a limited number of observations affected by different kinds of substantial "noise", both biological and technical. For this reason, structure learning in genomics faces

	Average size
PC	0.98
PC20	1.32
K2	2.64
K2-BIC	2.30
G-BIC	2.15
G-BICm	2.49
G-BICl	2.71
CK2	4.10

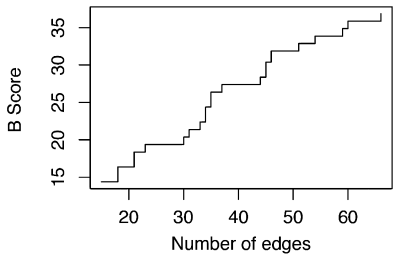


Fig. 6. *Drosophila melanogaster* experiment: Average size of the Markov blanket for different algorithms (left) and B score as a function of the number of edges in the inferred DAG.

a lot of previously unexplored problems and our goal was to better understand the choices made in practice. In particular, we focused on impact of categorizing gene expression measurements and including vague prior information. To this end, we analyzed a real dataset and performed a simulation study specifically designed to mimic limitations of real studies, such as model misspecification and a low signal to noise ratio.

We found that including prior information in the form of a topological ordering can significantly improve the performance, both in terms of network reconstruction and predictive accuracy. This is reflected in the fact that K2 variants, in spite of relying on a heuristic search method, performs either better or equally well as the exact Gobnilp method not including any prior information. This observation is especially important with the limited number of observations and was confirmed by both real and simulated datasets.

While results of the simulation study and the real study coincide to a large extent, there are some noticeable differences. For instance, the CK2 algorithm, which performs poorly in the real study, gives very good results in simulation studies. Such discrepancies are not surprising, since the performance of structure learning algorithms hinges heavily on the data generating mechanism and the real mechanism is, of course, unknown. On one hand, this justifies slight skepticism towards results from artificial experiments. On the other hand, if we would to rely on real datasets only, the absence of gold standard would hinder estimation of specificity, sensitivity, power and accuracy of different methods. For this reason, producing biologically plausible networks and distributions is an open problem and an active area of research, see for instance [25,29].

Another interesting observation regards the performance of the PC algorithm whose predictive accuracy in the simulation studies was good despite its Hamming distance to the true graph being large. The opposite phenomenon was seen in the K2 algorithm. We believe that this is associated with the well known “to predict or to explain” dilemma: simpler models give better predictions than their complex competitors, which in turn might be closer to the true underlying model, see [27]. In our study PC inferred sparser graphs and thus simpler models. Moreover, The PC algorithm at 5% significance level reaches lower Hamming distance than the one at 20% significance level, which infers denser graphs. Even though the difference is not statistically significant, it is consistent across different sample sizes and the two studies. These results confirm the findings presented in [9], where it was shown that when the sample size is small, the PC algorithm should be used with the “remove” rule, which encourages sparsity. On the other hand, the high B score of the K2 algorithm might also be in part attributed to the fact that the predictive accuracy was studied for the case of $n = 20$, and while the performance of K2 improves very quickly with an increasing sample size, see Fig. 1, for the lowest sample sizes the K2 scoring criterion might not be the best option; the BIC criterion employed in K2-BIC, gave better results.

There is a lot of concern regarding the application of structure learning algorithms in genomics setting. When the goal is to elucidate biological mechanisms governing gene expression, reflected in the reconstruction of the gene network, we would agree that this concern is justified. The signal to noise ratio in genomic studies does not seem to allow for an accurate reconstruction, at least for the time being. From the prediction perspective, however, the results reported here are encouraging: learned graphs, that can be considered as rough approximations of the true network, manage to bring considerable improvement over the procedure that does not assume or look for any conditional independence relations between genes. This is an important empirical conclusion that we draw from this study.

In this work we considered learning from purely observational data. Perturbation experiments, in which specific genes are targeted and blocked, are becoming more and more common in the lab practice. They represent the key instrument for both understanding functions of single genes, and elucidating the regulatory network, since other genes affected by the perturbation must be downstream from the targeted gene. The techniques for the joint analysis of the observational and interventional data are already available [24], but their widespread use is limited by the high cost of perturbation experiments.

Gene expression data convey partial information about a biological regulatory network consisting of genes, proteins, microRNAs and other smaller molecules. Inferred edges may indicate transcription regulation (which would correspond to gene–gene interaction), but also a protein–protein interaction. On the basis of gene expression data only, we are not able to distinguish between these competing explanations, which implies that what is inferred is a mixture of a gene regulatory network and a protein interaction network. In order to gain more insight and improve the performance of structure learning

Table A.3
Study 1: Structural Hamming distance (and its standard deviation).

n	PC		PC20		K2		K2-BIC		G-BIC		G-Ord		CK2	
20	19.54	(2.14)	20.48	(1.90)	19.47	(1.41)	18.62	(1.86)	21.16	(1.86)	19.34	(1.81)	13.05	(1.44)
30	19.33	(2.12)	20.22	(1.89)	18.52	(1.44)	17.84	(1.78)	20.60	(1.95)	18.80	(1.87)	9.88	(1.31)
50	18.97	(2.11)	19.74	(1.89)	17.20	(1.47)	16.46	(1.67)	19.90	(2.00)	17.73	(1.83)	6.34	(1.01)
100	18.27	(2.09)	18.90	(1.94)	14.52	(1.47)	13.90	(1.52)	18.94	(2.09)	15.71	(1.89)	3.15	(0.67)
500	16.60	(2.09)	17.62	(2.01)	7.57	(1.14)	8.52	(1.19)	16.41	(2.34)	7.90	(1.27)	1.27	(0.50)

Table A.4
Study 2: Structural Hamming distance (and its standard deviation).

n	PC		PC20		K2		K2-BIC		G-BIC		G-Ord		CK2	
20	19.02	(2.25)	19.90	(2.11)	18.68	(1.73)	17.98	(2.16)	21.12	(1.88)	18.48	(2.09)	18.37	(2.26)
30	18.55	(2.30)	19.33	(2.18)	17.70	(1.88)	17.24	(2.18)	20.32	(2.02)	17.35	(2.22)	17.71	(2.29)
50	18.01	(2.38)	18.65	(2.25)	16.20	(2.07)	16.17	(2.26)	19.27	(2.19)	15.74	(2.46)	16.66	(2.30)
100	17.10	(2.45)	17.72	(2.40)	14.51	(2.24)	14.75	(2.31)	17.36	(2.34)	13.48	(2.67)	15.33	(2.36)
500	15.51	(2.90)	16.58	(2.81)	11.73	(2.20)	11.95	(2.16)	13.51	(3.05)	9.91	(2.61)	14.40	(2.58)

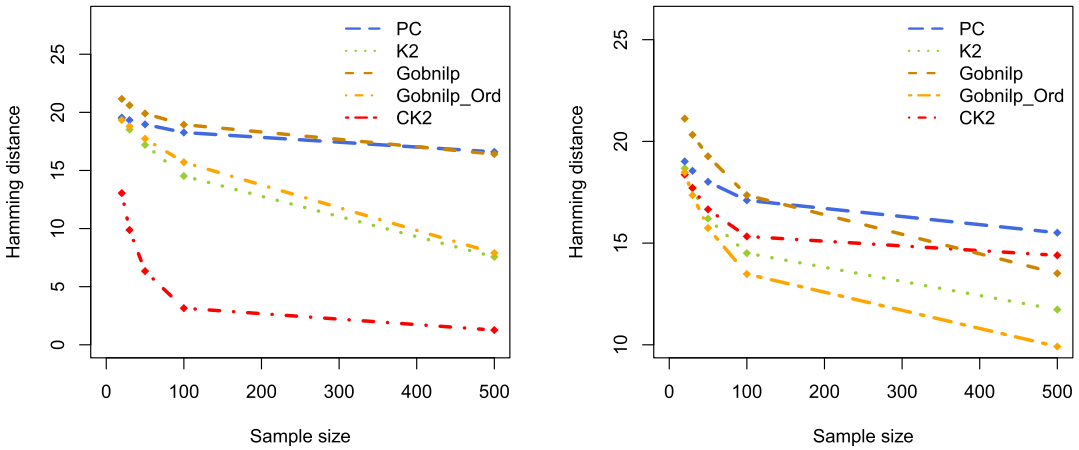


Fig. A.7. Simulation study: Pooled Structural Hamming distance in Study 1 (left) and Study 2 (right).

algorithms, complementary information from other sources, such as proteomic and ChIP Seq data, is needed. How these different sources should be integrated into the learning process remains an open question.

Main criticism of using Bayesian networks for modeling gene networks is the acyclicity constraint – biological networks contain feedback loops. If we assume that the network is a system evolving over time and temporal data are available, a possible solution is offered by dynamic Bayesian networks [22,19], which amounts to unfolding feedback loops. When temporal data are not available, gene expression measurements provide a snapshot of a dynamic system, and therefore a Bayesian network approach should be taken as a preliminary step towards forming new testable hypothesis, which could then ideally be verified in experimental conditions.

Appendix A. Results of experiments including Gobnilp with the topological ordering

Here, we report results that include the evaluation of Gobnilp with the ordering constraint: G-Ord. The criterion used with G-Ord is BIC. Two comparisons are especially interesting

- G-BIC and G-Ord, and
- K2-BIC and G-Ord.

Namely, the first comparison allows us to identify the effect of including prior information. The second comparison shows the effect of using the greedy search strategy (K2-BIC) instead of a complete search (G-Ord).

As expected, imposing a correct topological ordering significantly improves Gobnilp's ability to recover the graphical structure, as reflected by the structural Hamming distance, see Tables A.3, A.4 and Fig. A.7.

The B score obtained by G-Ord is also lower than G-BIC, see Fig. A.8. Note that this improvement in the B score is less pronounced than the improvement in the structural Hamming distance. This is not surprising, since by examining the Fig. A.7, we see that the effect of prior knowledge seems to be small for the lowest sample size ($n = 20$) and becomes more

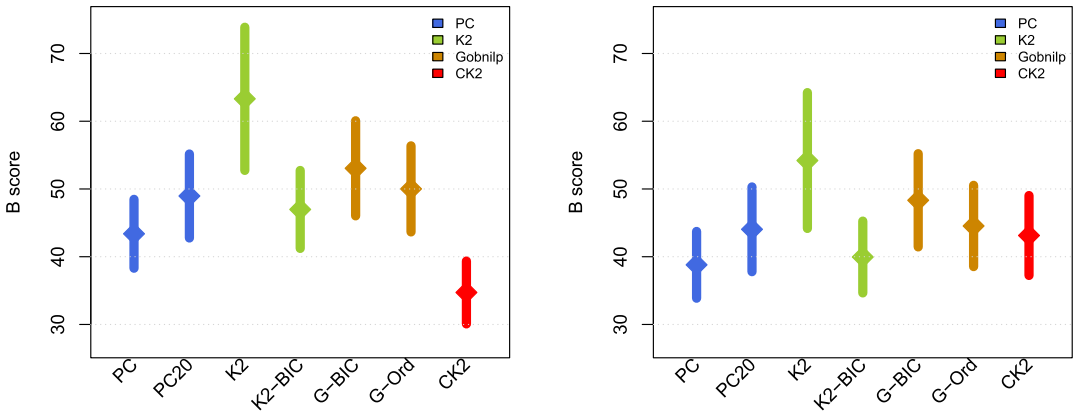
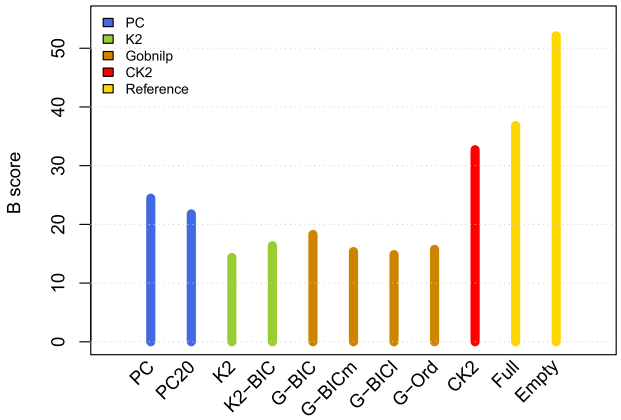


Fig. A.8. Simulation studies: mean value of the B score and its approximated 95% confidence interval in Study 1 (left) and Study 2 (right).



	B score
PC	24.46
PC20	21.79
K2	14.37
K2-BIC	16.37
G-BIC	18.29
G-BICm	15.37
G-BICl	14.86
G-Ord	15.76
CK2	32.72

Fig. A.9. *Drosophila melanogaster* experiment: B score of different algorithms.

visible with the increasing sample size. However, in the prediction evaluation, we have examined only the case with the smallest sample size.

Finally, when applied to real data, G-Ord performs slightly better than G-BIC and is comparable to K2, which achieves the lowest B score, see Fig. A.9.

References

[1] S. Acid, L.M. de Campos, Searching for Bayesian network structures in the space of restricted acyclic partially directed graphs, J. Artif. Intell. Res. (2003) 445–490.
[2] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, D. Di Bernardo, How to infer gene networks from expression profiles, Mol. Syst. Biol. 3 (1) (2007) 78.
[3] G.W. Brier, Verification of forecasts expressed in terms of probability, Mon. Weather Rev. 78 (1) (1950) 1–3.
[4] R. Castelo, A. Roverato, A robust procedure for Gaussian graphical model search from microarray data with p larger than n, J. Mach. Learn. Res. 7 (2006) 2621–2650.
[5] T. Chen, H.L. He, G.M. Church, et al., Modeling gene expression with differential equations, in: Pacific Symposium on Biocomputing, vol. 4, World Scientific, 1999, pp. 29–40.
[6] G.F. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Mach. Learn. 9 (4) (1992) 309–347.
[7] J. Cussens, Bayesian network learning with cutting planes, in: Proceedings of the Twenty-Seventh Annual Conference on Uncertainty in Artificial Intelligence, UAI-11, AUAI Press, Corvallis, OR, 2011, pp. 153–160.
[8] J. Cussens, M. Bartlett, GOBNILP 1.6.2 User/Developer Manual, 2013.
[9] M. De Jongh, M.J. Druzdzel, Evaluation of rules for coping with insufficient data in constraint-based search algorithms, in: European Workshop on Probabilistic Graphical Models, Springer, 2014, pp. 190–205.
[10] F.J. Díez, M.J. Druzdzel, Canonical Probabilistic Models for Knowledge Engineering, Tech. Rep. CISIAD-06-01, UNED, Madrid, Spain, 2006.
[11] V. Djordjilović, Graphical Modelling of Biological Pathways, Ph.D. thesis, University of Padova, 2015.
[12] F. Emmert-Streib, G. Glazko, G. Altay, R. de Matos Simoes, Statistical inference and reverse engineering of gene regulatory networks from observational expression data, Front. Genet. 3 (8) (2012).
[13] C. Fraley, A. Raftery, L. Scrucca, MCLUST: normal mixture modeling for model-based clustering, classification, and density estimation, R Package, http://cran.rproject.org/package=mclust, 1999.
[14] C. Fraley, A.E. Raftery, Model-based clustering, discriminant analysis, and density estimation, J. Am. Stat. Assoc. 97 (458) (2002) 611–631.
[15] N. Friedman, M. Linial, I. Nachman, D. Pe'er, Using Bayesian networks to analyze expression data, J. Comput. Biol. 7 (3–4) (2000) 601–620.
[16] A.J. Hartemink, Reverse engineering gene regulatory networks, Nat. Biotechnol. 23 (5) (2005) 554–555.

[17] D. Heckerman, J. Breese, A new look at causal independence, in: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, Seattle, WA, Morgan Kaufmann, 1994, pp. 286–292.
[18] D. Husmeier, Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks, Bioinformatics 19 (17) (2003) 2271–2282.
[19] S.Y. Kim, S. Imoto, S. Miyano, Inferring gene networks from time series microarray data using dynamic Bayesian networks, Brief. Bioinform. 4 (3) (2003) 228–235.
[20] M. Koivisto, K. Sood, Exact Bayesian structure discovery in Bayesian networks, J. Mach. Learn. Res. 5 (2004) 549–573.
[21] A.L. Madsen, F. Jensen, U.B. Kjaerulff, M. Lang, The Hugin tool for probabilistic graphical models, Int. J. Artif. Intell. Tools 14 (03) (2005) 507–543.
[22] B.-E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, F. d'Alche Buc, Gene networks inference using dynamic Bayesian networks, Bioinformatics 19 (Suppl. 2) (2003) ii138–ii148.
[23] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0, 2013, http://www.R-project.org/.
[24] A. Rau, F. Jaffrézic, G. Nuel, Joint estimation of causal effects from observational and intervention gene expression data, BMC Syst. Biol. 7 (1) (2013) 1.
[25] E. Salviato, V. Djordjilović, M. Chiogna, C. Romualdi, Simpathy: a new method for simulating data from perturbed biological pathways, Bioinformatics (2016), http://dx.doi.org/10.1093/bioinformatics/btw642, in press.
[26] J. Schäfer, K. Strimmer, A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics, Stat. Appl. Genet. Mol. Biol. 4 (1) (2005).
[27] G. Shmueli, To explain or to predict?, Stat. Sci. 25 (3) (2010) 289–310.
[28] P. Spirtes, C.N. Glymour, R. Scheines, Causation, Prediction and Search, Adaptive Computation and Machine Learning, MIT Press, 2000.
[29] T. Van den Bulcke, K. Van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor, K. Marchal, SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms, BMC Bioinform. 7 (1) (2006) 1–12, http://dx.doi.org/10.1186/1471-2105-7-43.
[30] J. Vomlel, M. Studený, Graphical and algebraic representatives of conditional independence models, in: Advances in Probabilistic Graphical Models, in: Studies in Fuzziness and Soft Computing, vol. 213, Springer, 2007, pp. 55–80.