An empirical comparison of popular algorithms for learning gene networks

Vera Djordjilović
Department of Statistical Sciences
University of Padova
djordjilovic@stat.unipd.it

Monica Chiogna
Department of Statistical Sciences
University of Padova
monica@stat.unipd.it

Jiří Vomlel
Institute Of Information Theory and Automation
Czech Academy of Sciences
vomlel@utia.cas.cz

Abstract

In this work, we study the performance of different algorithms for learning gene networks from 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. We also discuss merits and disadvantages of categorizing gene expression measurements.

1 Introduction

The interest in modelling gene networks has increased in recent years for two reasons. It is a widely accepted stance that a number of disorders and pathologies are associated with subtle changes in gene functioning. Better understanding of the mechanism that governs gene expression is an essential first step towards the development of efficient and highly specific drugs acting on molecular level. In addition to that, technological advances seen in the last two decades drastically reduced experimental costs, which made measurements of biological activity more readily available. This led to a growing body of experimentally obtained knowledge that is stored, in numerous forms, 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. Although pathway diagrams represent our up-to-date knowledge of the cellular processes, we can not always assume that derived mathematical graphs will be the optimal structure for statistical modelling. There are a number of reasons to consider them tentative models, see [4], and for this reason structure learning is an important task in genomics setting.

In this empirical comparison, we consider representatives of different structure learning approaches, such as the PC algorithm [8], the Gobnilp method [3] and the K2 algorithm [2]. We perform an extensive simulation study in which we study whether the approaches that include prior information, such as K2, perform better than those that rely on data only. We also look at the impact of discretization. In addition to a simulation study, we consider real data from the *Drosophila Melanogaster* experiment performed by the University of Padova [4]. In this experiment that focused on a WNT signalling pathway in a fruit fly, the expression of 12 genes was measured. Figure 5 shows a DAG derived from a WNT pathway diagram, featuring only genes measured in the experiment.

2 Structure learning algorithms

In this empirical study, we consider a number of variants of the PC algorithm [8], the K2 algorithm [2] and the exact Gobnilp method [3]. Of the examined approaches, the K2 algorithm and all modifications of the K2 algorithm considered here, 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.

To summarize, in this empirical study, we consider the following options.

PC The PC algorithm using χ^2 test of independence at the 5% significance level.

PC20 The PC algorithm using χ^2 test of independence at the 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 the modified BIC criterion (the penalty term is multiplied by a factor of 10^{-3}).

G-BICl The Gobnilp algorithm where the 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 maximising the log likelihood function.

CK2 The CK2 algorithm proposed in [4]. The only algorithm in this study that is applied to the continuous measurements.

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, [7] considered both discrete and continuous models. It is clear that the former attenuates the effect of the technical variability, but might lead to information loss, and is sensitive to the choice of the categorization procedure. The former incurs no information loss, but is 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 hypothesis.

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 [7]. When control data are not available, we propose to perform categorization based solely on data at hand. It is assumed that genes can assume 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

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

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

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

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 is called Model Based Clustering and was introduced by [5]. We used its implementation in the R package `mclust` [6]. 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 predictive accuracy

When evaluating the predictive accuracy of different approaches, we restricted our attention to a case with small sample size; a situation most relevant for our field of application. We adopted 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 used to evaluate the predictive accuracy: prediction of the value of every variable is computed given the values of all other variables.

To measure the distance between the observed value and the predicted value for variable Y_i fixing all remaining variables to the values observed on the removed observation j, we use the Brier score, introduced in [1]. If we denote ${}_jy_i = ({}_jy_{i1}, \ldots, {}_jy_{i\hat{K}_i})$ the observed value of variable Y_i in the jth observation, $j = 1, \ldots, n$, the Brier score is defined as

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

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 predictive accuracy with a scalar $B = \sum_{j=1}^{n} \sum_{i=1}^{p} {}_jb_i$. Obviously, algorithms having 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 (1).

3 Simulation study

To attenuate dependence of our conclusions on characteristics of individual graphs, we randomly generated 10 DAGs on 10 nodes. We achieved this by randomly generating

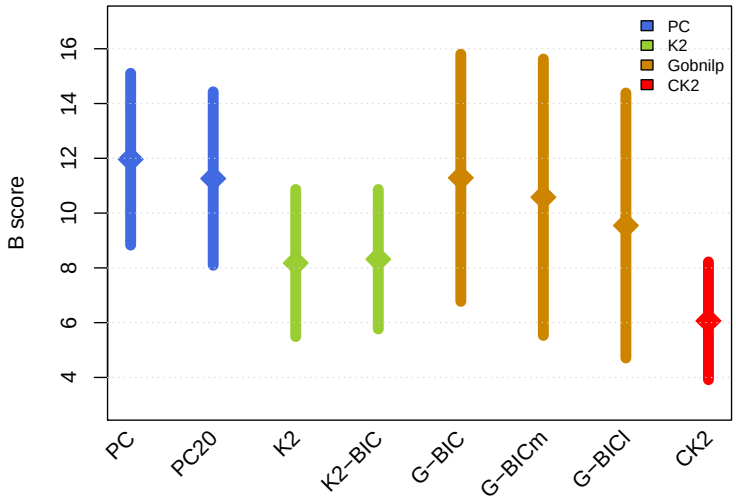

Figure 1: Simulation study: mean value of the B score and its 95% confidence interval.

10 adjacency matrices – for each graph we set a sparsity parameter $\pi \in (0.3, 0.5)$ and fixed the topological ordering. We next sampled an observation from a Bernoulli variable with the parameter π for each plausible edge (corresponding to the upper triangular part of the adjacency matrix) to obtain an adjacency matrix uniquely determining the corresponding DAG. When generating observations from a single DAG, our intention was to mimic the situation in which each gene has two underlying states (low and high expression), that are then affected and, to a certain level, "masked" by some biological and technical variation. We thus generated observations from a mixture of two multivariate normal distributions with a given graphical structure (the so-called Gaussian Bayesian networks, each with weight 0.5), where parameters of each component were randomly sampled from prespecfied intervals. To generate observations for a single component we adopted the structural equations approach, in which each variable is a linear function of its parents and a random error. More precisely, for each of the two components we have

$$X_i = \alpha_i + \boldsymbol{\beta}_i^{\mathsf{T}} \mathrm{pa}(X_i) + \epsilon_i, \quad i = 1, \ldots, p,$$

where $\epsilon_i \sim \mathsf{N}(0, \sigma_i^2)$ is the random disturbance, $\boldsymbol{\beta}_i$ is the vector of regression coefficients giving dependence of X_i on its parents, and α_i is an intercept. Both components were set to have the same matrix of $\boldsymbol{\beta}$ coefficients, so that the dependence structure is shared across components, while the intercept and the random fluctuation were allowed to vary. Before passing these datasets to the algorithms using categorical

variables, we performed categorization as described in 2.1. Namely, we performed model based clustering, where each variable was allowed to have either two or three clusters, depending on the model fit. In this situation, we knew that there were two underlying states–corresponding to two clusters–but we estimated the number of clusters from data so as to approach the conditions of a real study as close as possible. For each graph, we randomly generated 100 datasets.

We first look at the ability of considered algorithms to reconstruct the underlying graphical structure from observations. We rely on two measures: 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. For each considered sample size and for each of the 10 DAGs, we generated 100 datasets and applied structure learning algorithms. The pooled results are shown in Tables 1 and 2 and Figure 2, that shows graphically how PPV and Sensitivity change with sample size for different approaches. Given that the results of the approaches of the same type (such as PC and PC20; and K2 and K2BIC) have nearly identical results, we show one representative per group, namely PC, GBIC and K2. We see that CK2 gives best results in terms of PPV, and even more strikingly in terms of sensitivity. CK2 is followed by the other two (categorical) K2 approaches and Gobnilp methods. On the other hand, PC algorithm performs poorly in this setting. An interesting question is whether these measures of performance depend on the density of the true underlying DAGs. Figure 3 shows how PPV and Sensitivity depend on the number of edges of the DAG used to generate data. For each of the 10 DAGs, we show the value of PPV and Sensitivity for the largest sample size $n = 500$. We see, perhaps not surprisingly, that PPV increases roughly linearly with the number of edges in the underlying DAG, while sensitivity seems largely unaffected. As an illustration of the performance of considered approaches in reconstructing the "true" DAG, we show one example of a reconstructed network in Figure 4. Alongside a "true" DAG used to simulate data there is a DAG inferred by the CK2 algorithm, from one of the 100 simulated datasets ($n = 500$).

Next, we look at predictive accuracy of considered algorithms. Here, we restricted our attention to the smallest sample size ($n = 20$) for two reasons. It is the situation most relevant to our field of application, where the number of observations is usually limited. 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 100 generated datasets of size $n = 20$, we computed the B score 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 95% confidence interval are shown in Figure 1. CK2 reached the lowest B score, followed by K2 and K2-BIC. Of all Gobnilp methods, the likelihood one G-BICl leads to the lowest B score. PC variants perform slightly worse than Gobnilp variants, but the difference is less pronounced than in network reconstruction.

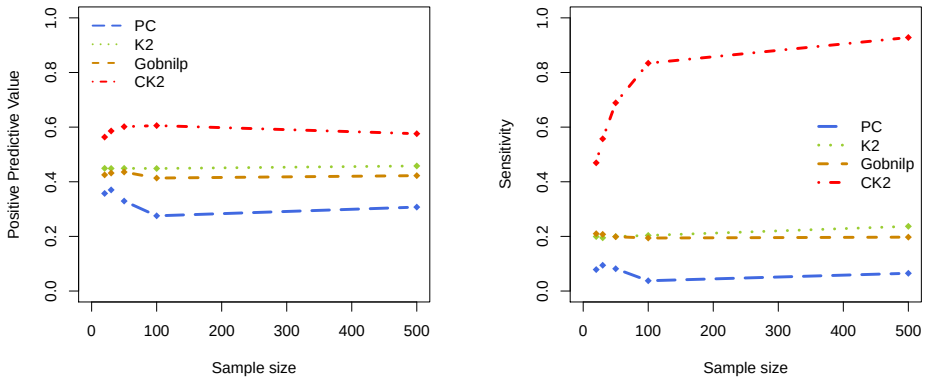

Figure 2: Simulation study: Pooled positive predictive accuracy (left) and sensitivity (right) of considered algorithms for different samples sizes.

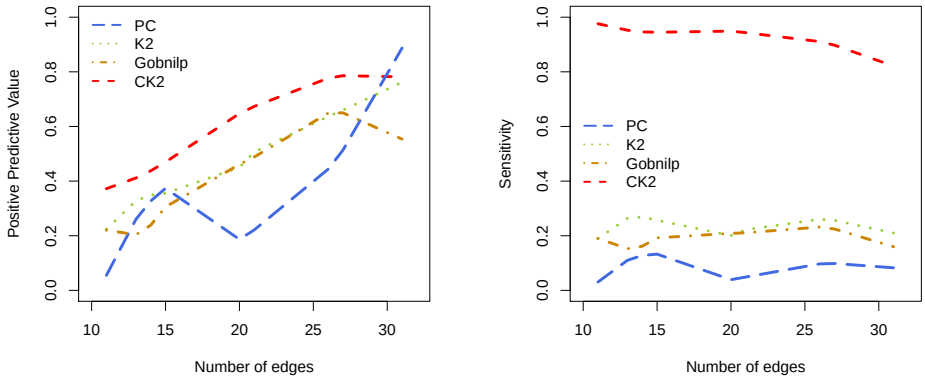

Figure 3: Simulation study: Positive predictive accuracy (left) and sensitivity (right) as a function of the number of edges of the true underlying DAG, for the 10 randomly generated DAGs and the sample size $n = 500$.

Table 1: Pooled positive predictive value.

n	PC	PC20	K2	K2-BIC	GBIC	GBICm	GBICl	CK2
20	0.36	0.35	0.45	0.45	0.43	0.42	0.40	0.56
30	0.37	0.37	0.45	0.45	0.43	0.41	0.40	0.59
50	0.33	0.34	0.45	0.45	0.44	0.41	0.40	0.60
100	0.28	0.25	0.45	0.45	0.41	0.39	0.40	0.61
500	0.31	0.30	0.46	0.46	0.42	0.40	0.41	0.58

Table 2: Pooled sensitivity.

n	PC	PC20	K2	K2-BIC	GBIC	GBICm	GBICl	CK2
20	0.08	0.09	0.20	0.19	0.21	0.21	0.21	0.47
30	0.09	0.10	0.19	0.19	0.21	0.20	0.21	0.56
50	0.08	0.09	0.20	0.20	0.20	0.21	0.20	0.69
100	0.04	0.04	0.20	0.20	0.19	0.21	0.21	0.83
500	0.06	0.07	0.24	0.24	0.20	0.22	0.22	0.93

4 Drosophila Melanogaster experiment

The experimental data from the *Drosophila Melanogaster* experiment performed by the University of Padova [4] consist of 28 observations of 12 genes. All measured genes belong to the WNT signalling pathway involved in embryonic development. DAG derived from this pathway is shown in Figure 5. 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.

The Figure 6 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. This is a little surprising, since Gobnilp finds globally optimal structures, while K2-BIC uses the ordering of variables, and thus might suffer from misspecification. In addition to that, K2-BIC relies on the greedy search, possibly restricting the search space enough to miss the global optima. In fact, structures found by Gobnilp have a lower BIC criterion (and thus a better fit to the data), but are inferior when it comes to prediction. This observation, together with a success of the K2, suggests that possibly the subject matter knowledge employed to specify the ordering of variables is the reason behind their good performance. To test this hypothesis, we generated 20 random orderings and passed them to the K2 algorithm. None of the twenty computed scores is lower than that that determined by pathway, providing support for the practice of using

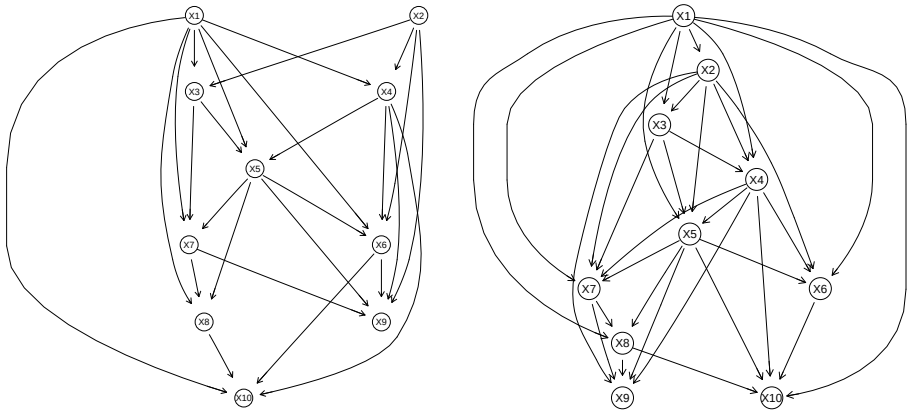

Figure 4: Simulation study: One of the 10 DAGs used to simulate data (left) and the network reconstructed by CK2 from 500 observations.

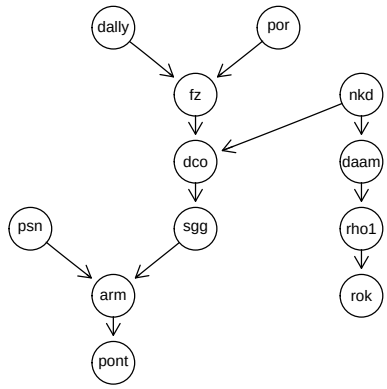

Figure 5: *Drosophila melanogaster* experiment: DAG derived from a diagram representing WNT signaling pathway in fruit flies.

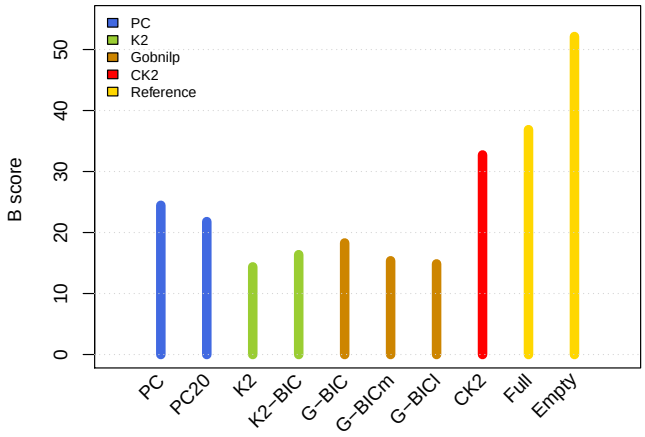

Figure 6: *Drosophila melanogaster* experiment: B score of different algorithms.

the prior information in the form of a topological ordering.

The right plot in Figure 7 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. The results are reported in the Table shown in the left panel of Figure 7. 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 a highly specific setting of gene networks. This area is atypical in that

	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

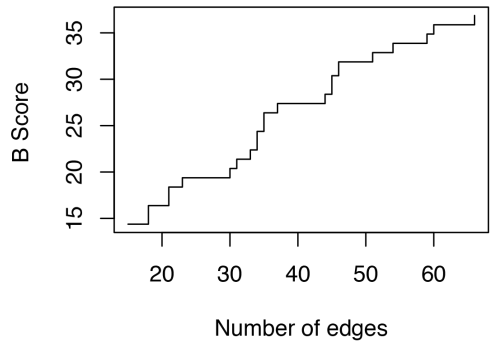

Figure 7: *Drosophila melanogaster* experiment: Average size of the Markov blanket for different algorithms (left) and B score as a function of the number of edges in the inferred DAG.

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 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 categorising gene expression measurements and including vague prior information. To this end, we analysed a real dataset and performed a simulation study specifically designed to mimic limitations of real studies.

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 algorithm, 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.

Results of the simulation study and the real study coincide to a large extent. The most striking difference is the performance of the CK2 algorithm, the only considered algorithm designed for continuous variables. While it performs poorly in the real study, in the simulation study it gives the best results. One possible explanation concerns the simulation mechanism: the data generating mechanism specified in the simulation study might not be a good approximation of the mechanism that gave rise to measurements in the real study. CK2, relying on continuous measurements, would be more sensitive to this difference with respect to its competitors using categorized data. Possible future work would involve investigation of different data generating mechanisms. It would be highly interesting to generate data from a discrete Bayesian network and then introduce random fluctuation for each variable independently.

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.

References

[1] Brier G. W., (1950), Verification of forecasts expressed in terms of probability, *Monthly weather review*, 78(1):1–3.

[2] Cooper, G.F. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine learning*. 9(4):309–347.

[3] Cussens, J. and Bartlett, M. (2013). GOBNILP 1.6.2.

[4] Djordjilović, V. (2105). *Graphical modelling of biological pathways*. PhD thesis, University of Padova.

[5] Fraley, C. and Raftery, A.E., (2002), Model-based clustering, discriminant analysis, and density estimation *Journal of the American Statistical Association*, 97(458):611–631.

[6] Fraley C., Raftery A.E., Murphy T.B. and Scrucca L. (2012), `mclust` Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation, *Technical Report*.

[7] Friedman N., Linial M., Nachman I., and Pe'er D. (2000), Using Bayesian networks to analyze expression data, *Journal of computational biology*, 7(3-4):601–620.

[8] Spirtes, P. and Glymour, C.N. and Scheines, R. (2000). *Causation, prediction, and search*. MIT press.