Applying Dynamic Bayesian Networks to Perturbed Gene Expression Data

Norbert Dojer, Anna Gambin, Bartek Wilczyński, Jerzy Tiuryn

April 19, 2005

Abstract

Motivation: A central goal of molecular biology is to understand the regulatory mechanisms of gene transcription and protein synthesis. Because of their solid basis in statistics, allowing to deal with the stochastic aspects of gene expressions and noisy measurements in a natural way, Bayesian networks appear attractive in the field of inferring gene interactions structure from microarray experiments data. However, the basic formalism has some disadvantages, e.g. it is sometimes hard to distinguish between the origin and the object of an interaction. Two kinds of microarray experiments yield data particularly rich in information regarding the direction of interactions: time series and perturbation experiments. In order to correctly handle them, the basic formalism must be modified. For example, dynamic Bayesian networks apply to time series microarray data.

Results: We extend the framework of dynamic Bayesian networks in order to handle perturbations. A new discretization method, specialized for datasets from time series perturbations experiments, is also introduced. We compare networks inferred from realistic simulations data by our method and by dynamic Bayesian networks learning techniques. We conclude that application of our method substantially improves inferring.

1 Introduction

As most genetic regulatory systems involve many components connected through complex networks of interactions, formal methods and computer tools for modeling and simulating are needed. Therefore, various formalisms were proposed to describe genetic regulatory systems, including Boolean networks and their generalizations, ordinary and partial differential equations, stochastic equations and Bayesian networks (see [4] for a review).

While differential and stochastic equations describe the biophysical processes at a very refined level of detail and prove useful in simulations of well studied systems, Bayesian networks appear attractive in the field of inferring the regulatory network structure from gene expression data. The reason is that their learning techniques have solid basis in statistics, allowing to deal with the stochastic aspects of gene expressions and noisy measurements in a natural way.

A *Bayesian network* (BN) is a representation of a joint probability distribution over a set of random variables. It consists of two components:

- a directed acyclic graph whose vertices correspond to random variables and edges indicate conditional (in-)dependence relations
- a family of conditional distributions for each variable, given its parents in the graph.

Together, these two components determine a unique joint distribution.

When applying Bayesian networks to genetic regulatory systems, vertices are identified with genes and their expression levels, edges indicate interactions between genes and conditional distributions describe these interactions. Given a set of gene expression data, the learning techniques for Bayesian networks allow one to infer networks that match this set well. However, as it was shown in [3], the problem of finding an optimal network is NP-hard. Consequently, one has to choose between restricting to small gene networks (a relatively quick algorithm was given in [13]) and inferring suboptimal networks by heuristic search methods (see for instance [5]).

It should be also pointed out that the basic BN formalism has some major limitations. First, several networks with the same undirected graph structure but different directions of some edges may represent the same distribution. In the case, given expression levels only, the origin and the object of an interaction become undistinguishable. Second, the acyclicity constraint rules out feedback loops, essential in genetic networks. Third, the dynamics of a gene regulatory system is lost.

The above limitations may be overcome by *Dynamic Bayesian networks* (DBNs), which model the stochastic evolution of a set of random variables over time. In comparison with BNs, the discrete time is introduced and conditional distributions are related to the values of parent variables in the previous time point. Moreover, in DBNs the acyclicity constraint is relaxed.

Given a set of time series of expression data, the learning techniques adapted from BNs allow one to infer dynamic networks that well match the temporal evolution recorded in the series (see [6], [11], [14], [12], [15], [8], [9], [18]).

A special treatment is required for experiments, in which expression of some genes were perturbed (e.g. knockout experiments). Since perturbations change the structure of interactions (regulation of affected genes is excluded), the learning techniques have to use data selectively. Inferring networks from perturbed expression profiles by means of BNs was investigated in [14] and [16].

In this report we extend the framework of DBN in order to handle perturbations and introduce a discretization method specialized for datasets from time series perturbations experiments. We compare networks inferred by our method and by DBN learning techniques. We conclude that application of our method substantially improves inferring.

2 Data and preprocessing

When analyzing learning procedure's efficiency, the procedure should be applied to the data generated by a known network, which then might be compared with the inferred one. To this aim, most studies apply procedures to gene expression data and compare inferred interactions with those found in biological literature. The disadvantage of this approach is that our knowledge of the structures of real biological networks is far from being complete even in the most deeply investigated organisms. Although many interactions between genes are known, there are very few results stating the absence of some interactions. Thus no conclusion can be drawn from the fact that the procedure inferred unknown interaction.

The above disadvantage is no longer present when data are generated from a mathematical model simulating real networks. However, the danger of this approach is that the employed model simplifies the real process, losing important biological features. In that case, analysis of a learning procedure is aimed at its ability to infer an artificial model rather than real biological networks.

Husmeier in [8] suggests that a satisfactory compromise between the above two extremes is to apply the learning procedure to data generated by a system of ordinary differentional equations.

In the present study we generate data using the model proposed in [17]. The model consists of 54 species of molecules, representing 10 genes with their transcription factors, promoters, mRNAs, proteins and protein dimers, connected through 97 elementary reactions, including transcription factor binding, transcription, translation, dimerization, mRNA degradation and protein degradation. The dynamics of the model is described by the system of ordinary differential equations of the following form:

$$\frac{d[G_2]}{dt} = k_{diG}[G]^2 - k_{udG}[G_2] - k_{biHG}[G_2][pH] + k_{ubHG}[G_2 \cdot pH] - k_{deG_2}[G_2]$$

$$\frac{d[pH]}{dt} = -k_{biHG}[G_2][pH] + k_{ubHG}[G_2 \cdot pH]$$

$$\frac{d[mH]}{dt} = -k_{dmH}[mH] + k_{trH}[pH] + k_{trHG}[G_2 \cdot pH]$$

$$\frac{d[H]}{dt} = -k_{deH}[H] + k_{tlH}[mH]$$

where t represents time, k_x are kinetic constants of related reactions, $[X]$ means concentration, pX, mX, X and X_2 are promoter, mRNA, protein and dimer X, respectively, finally $X \cdot Y$ stands for a transcription factor bound to a promoter.

The system is composed of structures reported in the biological literature (see [1], [2], [7]), i.e. a hysteretic oscillator, a genetic switch, cascades and a ligand binding mechanism that influences transcription (during the simulation, the ligand is injected for a short time). The whole network is shown in Fig. 1. All the equations and parameters of the model, as well as the MATLAB code to integrate it, are available in the supplementary materials to [17].

This model is chosen for two reasons. First, differential equations formalism and biological origin of the structure guarantee realistic simulations. Second, small size of the system (note that, according to microarray experiments data, the learning procedure will be provided with mRNA concentrations only) allows to avoid a noise arising from heuristic search methods, which are necessary when dealing with large networks. Such noise might influence comparison of methods.

Since genes G and K from the model are regulated by the same gene C and have the same kinetic constants, trajectories of their concentrations are identical. Consequently, their contributions to the regulatory interactions are undistinguishable, given the expression data. For this reason, Husmeier in [8] tests efficiency of DBN based learning techniques using the model slightly modified by identifying both genes. His result, obtained by Markov chain Monte

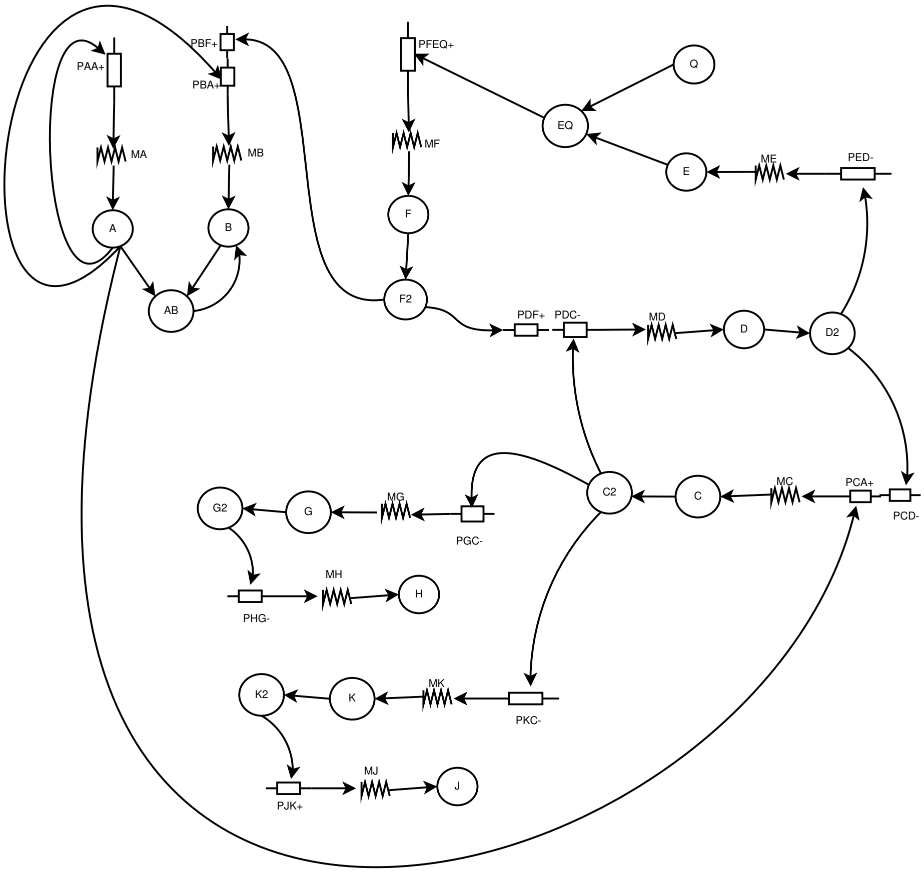

Figure 1: The gene regulatory network from [17]. Rectangles denote promoters, zigzags stand for mRNAs and circles indicate proteins, dimers and a ligand (Q)

Carlo method provided in Bayes Net Toolbox by Murphy [10], is shown on Fig. 2(b).

In the present study we introduce perturbations to the model. It is done by replacing the differential equation regarding the mRNA of a knocked out gene by the following equation:

$$\frac{d[mX]}{dt} = -k_{kX}[mX]$$

which makes the concentration exponentially converging to 0. There are executed altogether 11 simulations: one simulation with the basic system and 10 simulations each with one gene knocked out.

The total time of each simulation is set to 5000 minutes. At time 1000 minutes the ligand is injected for 10 minutes, completely changing the expression levels of all genes. The influence of the injection to expression decays throughout the interval 1500-3200 minutes (depending on the gene), but at time 2400 minutes system dynamics becomes similar to the initial one.

Husmeier [8] chooses in his test 12 time points in equal length intervals between 1100 and 1600 minutes, motivating that more information contains

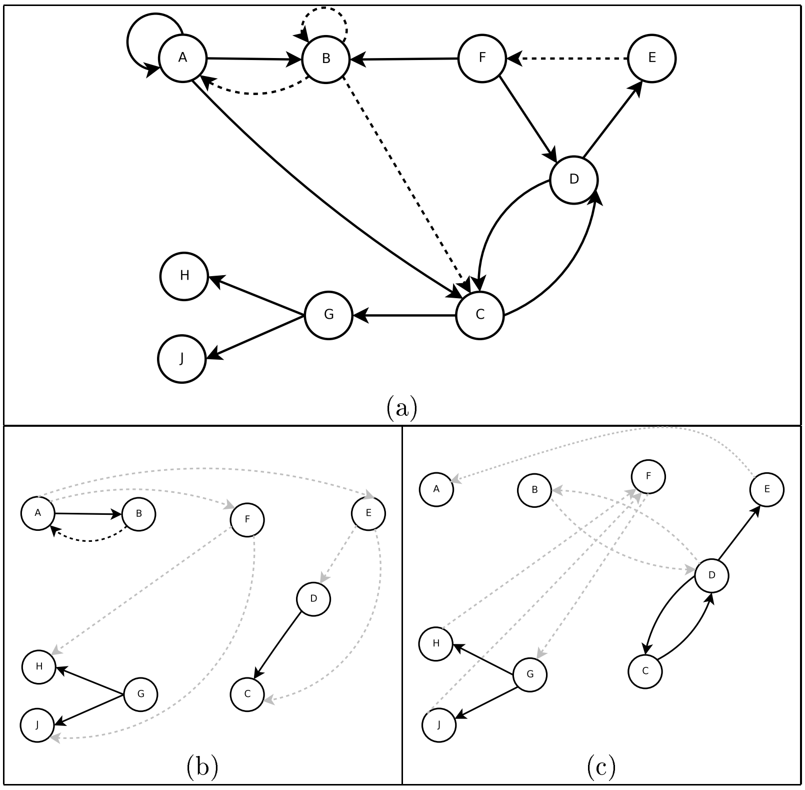

Figure 2: (a) The mRNA interactions in the system restricted to 9 genes; solid arrows denote transcriptional regulation, dashed ones represent interactions triggered by the ligand and posttransciptional regulation. (bc) The networks inferred from 12 slices from the interval 1100-1600: in [8] by Markov chain Monte Carlo method (b) and by our exact algorithm with BDe score and forbidden self-loops (c); black arrows show true inferred edges, grey dashed arrows represent spurious edges.

system got out of equilibrium by the ligand injection. In our tests Husmeier's choice is repeated and other intervals are tried, as reported below.

Before applying our learning procedure (as well as Husmeier's), the expression levels need to be discretized. One of the simplest methods of discretizing is partition of the interval of real numbers covering mRNAs concentrations of each gene into subintervals of equal length, relevant to particular disretized values. Another strategy is to base the discretization procedure on the meanings of introduced discrete expression levels (e.g. 'on'–'off' or 'under-expressed'–'baseline'–'over-expressed').

Husmeier in [8] applies the first approach with 3 discretized expression levels, and we follow him in the case of unperturbed data. For perturbed data we choose the second approach. Each concentration value from a knockout simulation is compared with the value of the same gene concentration from the same time

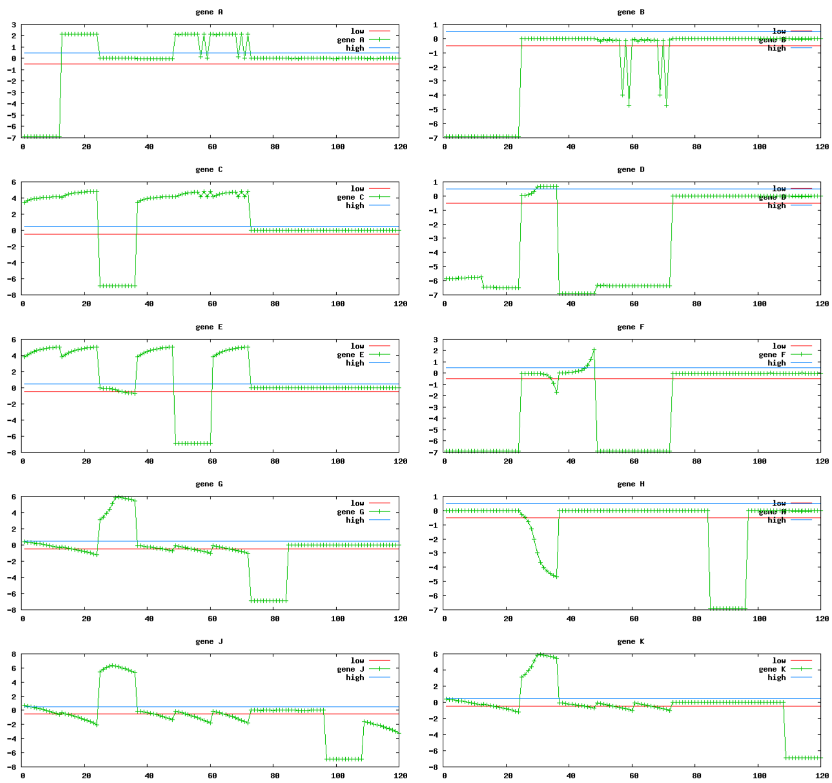

Figure 3: Log-ratios of mRNA concentration values for all knockout experiments. 12 time points in equal length intervals between 1100 and 1600 minutes are taken from each experiment (120 slices together). Ratios lower than 0.001 were set to this value. The horizontal lines indicate the top and bottom limits of the baseline expession level.

point of the basic system simulation. When the values are close each other, i.e.

$$\left| ln \frac{\text{perturbed value}}{\text{base value}} \right| < 0.5$$

the expression level is set to 'baseline', otherwise it is set to 'over-' or 'under-expressed'. The log-ratios of concentration values for 12 time points in equal length intervals between 1100 and 1600 minutes are shown on Fig. 3.

3 Methods

A *dynamic Bayesian network* $\mathcal{N}$ is a representation of stochastic evolution of a set of random variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ over discretized time. Represented temporal process is assumed to be *Markovian*, i.e. $P(\mathbf{X}(t)|\mathbf{X}(0), \mathbf{X}(1), \ldots, \mathbf{X}(t-1)) = P(\mathbf{X}(t)|\mathbf{X}(t-1))$, and *time homogenous*, i.e. $P(\mathbf{X}(t)|\mathbf{X}(t-1))$ are independent of t. The representation consists of two components:

- a directed graph $G = (\mathbf{X}, \mathbf{E})$ encoding conditional (in-)dependencies

- a family of conditional distributions $P(X_i(t)|\mathbf{Pa}_i(t-1))$, where $\mathbf{Pa}_i = \{X_j \in \mathbf{X} | (X_j, X_i) \in \mathbf{E}\}$

By asssumption, the joint distribution over all the possible trajectories of the process decomposes into the following product form

$$P(\mathbf{X}(0), \mathbf{X}(1), \ldots, \mathbf{X}(T)) = P(\mathbf{X}(0)) \prod_{t=1}^{T} P(\mathbf{X}(t)|\mathbf{X}(t-1))$$

Consequently, the evolution of the random variables is given by

$$\begin{aligned} P(\mathbf{X}(1), \ldots, \mathbf{X}(T)|\mathbf{X}(0)) &= \prod_{t=1}^{T} P(\mathbf{X}(t)|\mathbf{X}(t-1)) = \\ &= \prod_{t=1}^{T} \prod_{i=1}^{n} P(X_i(t)|\mathbf{Pa}_i(t-1)) = \prod_{i=1}^{n} \prod_{t=1}^{T} P(X_i(t)|\mathbf{Pa}_i(t-1)) \end{aligned} \quad (1)$$

The problem of learning a DBN is understood as follows: find a network graph that best matches a given dataset of time series of $\mathbf{X}$-instances. The notion of a good match is formalized by means of a *scoring function*, usually one of the following:

- MDL score [6], based on the *parsimony* principle, is proportional to the length of the network description together with the data description, the last being made according to the network parameters,

- BDe score [6] is derived from the posterior probability of the network, given the data (the prior distributions over the network parameters have to be assumed).

Owing to the product decomposition (1), both scores decompose into the sums over the set of random variables. This property is extremely useful in learning procedures, since the parents of each variable may be computed independently. When the set of variables is small enough (the boundary is about 20), one may score every subset as a possible parents set for each variable and choose the best match. Otherwise, heuristic search methods have to be applied and the decomposition property is helpful in reducing the computation cost when scoring locally changed networks.

Since our training datasets consist of mRNA concentrations of 10 genes only, we may apply an algorithm founding the optimal network. This choice allows us to avoid the noise caused by using heuristic search methods.

Edges in the inferred network graph witness conditional dependences between variables in neighbouring time points, which is interpreted as interaction between corresponding genes. However, a special care is required when inferring self-regulation. The reason is that an inferred self-depependence of a variable may be caused by the inertness of the system (especially when time intervals of data were small as compared with the lag of regulation processes), by the coarse-grained discretization or by too little impact of time processes into the variable. Therefore self-connecting edges often are either forced or forbidden in learned networks (see [6], [12]), which leaves inferring self-regulation out of account.

When expression of a gene is perturbed in an experiment (e.g. by knocking out), its natural regulation is blocked and replaced by the perturbation scheme. Consequently, regarding that gene's regulation mechanisms, the experiment contributes a noise instead of information. On the other hand, the rest of interactions might be significantly reflected, in particular those in which the gene acts as a regulator. Therefore our learning procedure has to make use of data from perturbation experiments selectively.

Recall that the parents sets of each gene may be inferred independently. Thus, when inferring parents of a particular gene, we apply the standard learning procedure to the dataset restricted to those experiments, in which this gene's expression was not perturbed. When parents of all genes are computed, the network graph is composed. Related method in the framework of static BNs was successfully used by Pe'er et al. in [14] and Yoo et al. in [16].

Our algorithm can be expressed as follows:

```
for each gene G
  choose all experiments with unperturbed expression of G
  for each potential parents set Pa of G
    compute the local score in G for Pa and chosen experiments
  choose the parents set of G yielding optimal score
compose the network from chosen parents sets
```

4 Results and discussion

In the first experiment we followed the procedure of Husmeier [8] (the system restricted to 9 genes, 12 time points chosen in equal length intervals between 1100 and 1600 minutes, simple discretization into 3 levels). The sensitivity of our exact algorithm was similar to Husmeier's heuristics – see Fig. 2(b) and (c).

Next experiments concerned the perturbed data. Recall that the entire system with all 10 genes was considered and the distretization was made according to the comparison of expression levels from perturbed and unperturbed profiles.

The first set of time points followed the above experiment, resulting in 6 true edges recovered (see Fig. 4(b)).

In the next experiment we decided to lengthen the time interval to 1500 minutes (from 1100 till 2600 minutes) in order to cover all the period of disequilibrium caused by the ligand injection. The resulting network includes 8 edges corresponding to direct transcriptional regulation, 1 edge due to interaction triggered by the ligand, 1 reflecting posttrascriptional regulation and no spurious edge (see Fig. 4(c)).

Another time intervals were tried, resulting in networks less accurate than the two above (data not shown), which confirms Husmeier's assertion of low information content of signals from a system being in equilibrium.

Since the datasets used in the above experiments were quite large (10 series 12 time points each gives 120 slices), we decided to choose for the next experiment 3 time points in equal length intervals between 1100 and 1600 minutes. The accuracy of the received network appeared to be higher than the one of the result of the experiment with 12 time points (see Fig. 4(d)). It shows that increasing the size of a dataset does not guarantee a success in inferring the network structure, and choosing suitable time intervals may be crucial.

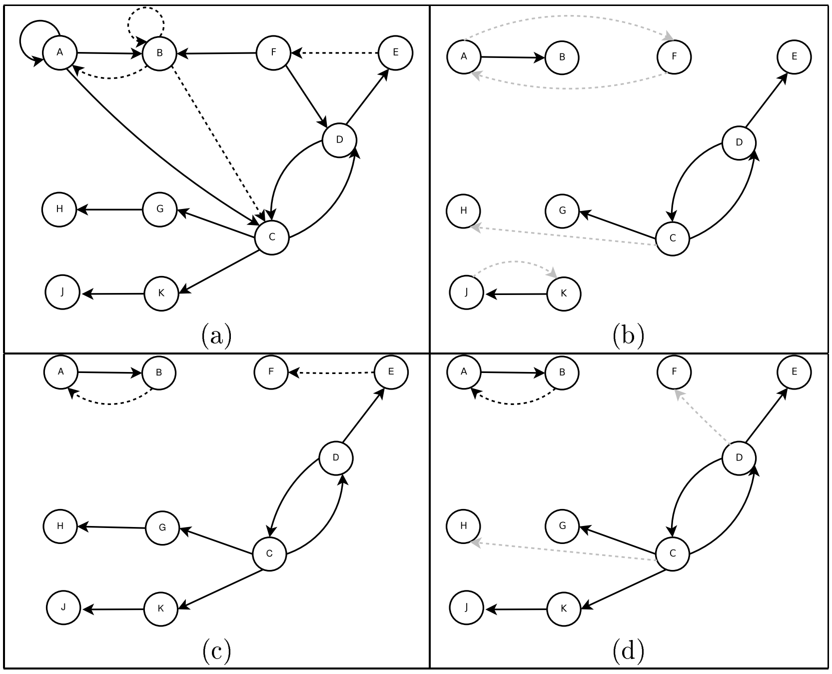

Figure 4: (a) The mRNA interactions in the considered system; solid arrows denote transcriptional regulation, dashed ones represent interactions triggered by the ligand and posttransciptional regulation. (bcd) The networks inferred from perturbed data: 12 slices from the interval 1100-1600 (b), 12 slices from the interval 1100-2600 (c) and 3 slices from the interval 1100-1600 (d); black arrows show true inferred edges, grey dashed arrows represent spurious edges. All the networks are obtained with MDL score and forbidden self-loops.

The above considerations refer to inferring *local* interactions between genes, represented by particular edges. In order to analyse the ability to infer a *global* interaction system, one has to compare the score of the original network with the scores of other networks. Since it is impossible to compute the scores of all graphs (there are 2^{100} directed graph structures with 10 nodes), we generated at random 100000 graphs. For each graph, edges were generated independently, each with the same probability, resulting in the expected number of 12 edges in the graph. This number was chosen to be close to the number of edges in the original graph, as well as in the best scored ones. Note that setting this probability to 1/2 (which results in the uniform distribution on the space of all graphs) would cause scores of most of graphs to be dominated by high penalties due to excessive structures.

Table 1 shows how many generated networks received (in various experiments) scores better then the original one restricted to direct transcriptional regulations. Table 2 shows corresponding values for the original network enriched with edges referring to posttransciptional regulation and an interaction triggered by the ligand. We compare the results obtained with combinations of MDL or BDe score with one of 3 options of the algorithm: with forbidden, permitted or forced self-loops, i.e. edges leading from a vertex to itself.

	dataset		s-l forbidden		s-l permitted		s-l forced	
pert.	interval	slices	MDL	BDe	MDL	BDe	MDL	BDe
−	1100-1600	12	24.409	10.572	18.236	9.114	30.72	35.786
+	1100-1600	12	0.008	0.182	0.044	0.875	30.92	2.543
+	1100-2600	12	0.001	0.001	0.002	0.145	28.999	0.274
+	1100-1600	3	4.191	0.003	4.406	0.021	30.472	2.59

Table 1: The percentages of generated networks with scores better than one of the original network, restricted to edges referring to direct transcriptional regulations. In the options of the algorithm with forbidden or forced self-loops, the original network was modified by removing or adding appropriate edges, respectively.

	dataset		s-l forbidden		s-l permitted		s-l forced	
pert.	interval	slices	MDL	BDe	MDL	BDe	MDL	BDe
−	1100-1600	12	56.048	22.559	68.423	43.407	62.273	65.437
+	1100-1600	12	0.049	0.001	3.204	0.114	61.073	1.763
+	1100-2600	12	0.004	0	1.061	0.029	59.382	0.124
+	1100-1600	3	33.636	0	58.808	0.02	61.453	5.762

Table 2: The percentages of generated networks with scores better than one of the original network, enriched with edges referring to interactions triggered by the ligand and posttransciptional. In the options of the algorithm with forbidden or forced self-loops, the original network was modified by removing or adding appropriate edges, respectively.

The tables show that using perturbed data significantly improves the possibility of inferring the original network (except the option of forced self-loops with MDL score). The best results are obtained in the experiment with 12 time points in the interval from 1100 till 2600 minutes, but the differences between results of all perturbation experiments are small in most cases. The exception is the experiment with 3 time points and MDL score.

Both exceptions show that MDL score is much more sensitive than BDe one to the number of expected edges in relation to the size of the dataset (note also that, for MDL score, the values in Table 2 are always greater than those in Table 1, which is no longer true for BDe score).

In the other cases the results are similar for both scores, a little bit better for MDL score in Table 1, and slightly better for BDe score in Table 2.

Comparison of the values for particular options of the algorithm shows that the best results are received when self-loops are forbidden, slightly worse when self-loops are permitted and significantly worse when they are forced. The analysis of the best scored networks with permitted self-loops leads to the statement that self-regulation of genes can not be handled within our framework correctly and requires more refined methods (data not shown). Therefore, as regards our algorithm's options, the best choice is the one with forbidden self-loops.

5 Acknowledgments

We thank Andrzej Mizera for implementing software applied in the present study. The research was supported by the KBN grant nr 3T11F02128.

References

[1] N Barkai and S Leibler. Circadian clocks limited by noise. *Nature*, 403(6767):267–268, Jan 2000.

[2] J L Cherry and F R Adler. How to make a biological switch. *J Theor Biol*, 203(2):117–133, Mar 2000.

[3] D. M. Chickering. Learning Bayesian networks is NP-complete. In D. Fisher and H.-J. Lenz, editors, *Learning from Data: Artificial Inteligence and Statistics V*. Springer-Verlag, 1996.

[4] Hidde de Jong. Modeling and simulation of genetic regulatory systems: a literature review. *J Comput Biol*, 9(1):67–103, 2002.

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

[6] N Friedman, K Murphy, and S Russell. Learning the structure of dynamic probabilistic networks. In G F Cooper and S Moral, editors, *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Inteligence*, pages 139–147, 1998.

[7] T S Gardner, C R Cantor, and J J Collins. Construction of a genetic toggle switch in Escherichia coli. *Nature*, 403(6767):339–342, Jan 2000.

[8] Dirk Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, 19(17):2271–2282, Nov 2003.

[9] Sunyong Kim, Seiya Imoto, and Satoru Miyano. Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data. *Biosystems*, 75(1-3):57–65, Jul 2004.

[10] K P Murphy. Bayes Net Toolbox. Technical report, MIT Artificial Intelligence Laboratory, 2002.

[11] K P Murphy and Mian S. Modelling Gene Expression Data Using Dynamic Bayesian Networks. Technical report, MIT Artificial Intelligence Laboratory, 1999.

[12] I Ong, J Glasner, and D Page. Modelling regulatory pathways in *E. coli* from time series expression profiles. *Bioinformatics*, 18 Suppl 1:S241–S248, 2002.

[13] S Ott, S Imoto, and S Miyano. Finding optimal models for small gene networks. *Pac Symp Biocomput*, pages 557–567, 2004.

[14] D Pe'er, A Regev, G Elidan, and N Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17 Suppl 1:215–224, 2001.

[15] B-E Perrin, L Ralaivola, A Mazurie, S Bottani, J Mallet, and F D'Alche-Buc. Gene networks inference using dynamic Bayesian networks. *Bioinformatics*, 19 Suppl 2:II138–II148, 2003.

[16] C. Yoo, V. Thorsson, and G.F. Cooper. Discovery of causal relationships in a generegulation pathway from a mixture of experimental and oberservational DNA microarray data. In *Proceedings of Pacific Symposium on Biocomputing 7:498-509*, 2002.

[17] Daniel E Zak, Francis J 3rd Doyle, Gregory E Gonye, and James S Schwaber. Simulation Studies for the Identification of Genetic Networks from cDNA Array and Regulatory Activity Data. In *Proceedings of the Second International Conference on Systems Biology*, pages 231–238, 2001.

[18] Min Zou and Suzanne D Conzen. A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics*, 21(1):71–79, Jan 2005.