



Reverse Engineering of Gene Regulatory Networks Combining Dynamic Bayesian Networks and Prior Biological Knowledge

Mariana C. de Souza  and Carlos H. A. Higa  

Federal University of Mato Grosso do Sul, Campo Grande, MS 79070-900, Brazil
mariana.caravanti@aluno.ufms.br, higa@facom.ufms.br
<https://www.facom.ufms.br>

Abstract. An important problem in the Systems Biology field is the reverse engineering of gene regulatory networks from gene expression data. In this work, we addressed this problem using a probabilistic graphical model known as Dynamic Bayesian Network to model the regulatory relations among the genes. We also used a Boolean formalism, assuming that each gene can take on two possible values: 0 (not expressed) and 1 (expressed). To learn the Dynamic Bayesian Network from time-series gene expression data we search for the network structure that best matches the data using the Bayesian Information Criterion score and the BDe score and compared them. Besides that, we used a source of prior biological knowledge from a database named STRING, unlike most of the reverse engineering algorithms that does not take into account any source of additional information. The results show that this approach can improve the quality of the inferred networks, and we also showed that the Dynamic Bayesian Network performs better than its standard version, Bayesian Network.

Keywords: Gene regulatory network · Dynamic Bayesian Network
Reverse engineering

1 Introduction

The reverse engineering of Gene Regulatory Networks (GRN) is an important problem studied in the Systems Biology and Bioinformatics field. The importance of dealing with this problem is that it can help us to understand the regulatory mechanisms of a biological phenomenon under study. For instance, researchers have studied GRN in the context of cell cycle [15], circadian cycle of plants [18], and more important, diseases such as cancer [9, 19]. Given the importance of this problem, several models of GRN along with algorithms for the inference of GRN have been proposed over the years. Generally, these algorithms receive a dataset of gene expression as input and outputs a regulatory

Supported by CNPq and UFMS.

network represented by a specific model. The input data are those provided by technologies such as DNA microarrays [22] and, more recently, RNA-Seq [27].

Among the proposed mathematical models of GRN are the Boolean networks [12], Bayesian networks [4], models based on ordinary differential equations [6], to name a few of them. To review a set of models of GRN we suggest the papers [7, 11, 17]. Algorithms for the reverse engineering of GRN are based on mathematical models and each one has its own characteristics besides advantages/advantages. We can not say that there is an algorithm that outperforms the others in every situation. For instance, some algorithms are specific to work with time-series data while some others work with steady-state data. It is worth mentioning that the reverse engineering of GRN is an ill-posed problem, *i.e.*, given a set of gene expression data, there could be several consistent solutions (networks) with these data. This fact makes the problem a very difficult one.

In this work, we modeled the gene expression in a Boolean fashion where a gene can take on two possible values: 0 (not expressed) and 1 (expressed). We also used a probabilistic graphical model known as Dynamic Bayesian Network, which is commonly used to represent complex stochastic processes. It differs from the standard Bayesian Network in the sense that it models the stochastic evolution of a set of variables over time. Both models are defined by a graphical structure and a set of parameters, which together specify a joint distribution over the random variables. Algorithms for learning the parameters and the structure of such networks have been developed [14, 21]. In this paper, to learn the structure of Dynamic Bayesian Networks from time-series gene expression data we search for the network structure that best matches the data using the Bayesian Information Criterion score [23] and the BDe score [8].

Most of the reverse engineering algorithms of gene regulatory networks do not take into account any sources of prior biological knowledge. The approach of using additional biological information in this problem has shown interesting results, as can be seen in [5, 26, 29]. The source of prior knowledge used in this work is a database called STRING [20, 25], which covers more than 2000 organisms in its latest version (10.0). In [13] a similar approach was performed using the Boolean network model and a feature selection algorithm. Here, we use a different model and approach do infer the model, but using the same source of prior biological knowledge. Because we are facing an ill-posed problem, this approach aims to improve the inference of gene interactions, once we take into account some interactions already supported by the literature.

To validate the methodology we used data from the DREAM challenge [2]. We also applied the methodology in a dataset of yeast (*Saccharomyces cerevisiae*) cell-cycle gene expression [24] and compared the results against the literature through the KEGG (Kyoto Encyclopedia of Genes and Genomes) database [10].

In the next section, we present the methodology, beginning with the introduction of the Dynamic Bayesian Networks, followed by the definitions of the score functions used in the learning process. We also explain how the prior biological knowledge database was used in the process. In Sect. 3 we discuss the results, followed by our conclusions, in Sect. 4.

2 Methodology

In this section we present the mathematical model of GRN used in our methodology along with the methodology to learn the model from the data. We also present the source of prior biological knowledge used and how it is used in the process of reverse engineering a GRN. In the final part of the section we discuss how the methodology is validated.

2.1 Dynamic Bayesian Networks

A *Dynamic Bayesian Network* (DBN) is a probabilistic graphical model defined by a network structure and a set of parameters to specify the joint probability of random variables. Let $\mathcal{X} = \{X_1, X_2, \dots, X_n\}$ be a set of random variables where each variable X_i can assume values in the finite set $\text{Val}(X_i)$. In DBN, unlike standard Bayesian networks, the time is taken into account such that $X_i[t]$ is the value of X_i at time t . Thus, $\mathcal{X}[t]$ is the set of all random variables at time t .

To represent the beliefs about the process, a probability distribution over the variables $\mathcal{X}[0] \cup \mathcal{X}[1] \cup \mathcal{X}[2] \cup \dots$ is necessary. We assume that the process is Markovian in the sense that $P(\mathcal{X}[t+1] | \mathcal{X}[0], \dots, \mathcal{X}[t]) = P(\mathcal{X}[t+1] | \mathcal{X}[t])$. The probability $P(\mathcal{X}[t+1] | \mathcal{X}[t])$ is time independent.

A DBN representing the joint distribution over all possible trajectories of a process is composed of: (i) a prior network B_0 specifying a distribution over $\mathcal{X}[0]$; (ii) a transition network B_{\rightarrow} over $\mathcal{X}[0] \cup \mathcal{X}[1]$ that is taken to specify $P(\mathcal{X}[t+1] | \mathcal{X}[t])$ for all t . In Fig. 1(a) and (b) we can see a simple example.

The transition probability of this network is given by

$$P_{B_{\rightarrow}}(\mathcal{X}[1] | \mathcal{X}[0]) = \prod_{i=1}^n P_{B_{\rightarrow}}(X_i[1] | \mathbf{Pa}(X_i[1])), \quad (1)$$

where $\mathbf{Pa}(X_i)$ is the parents of X_i . Thus, the structure of a DBN is defined by a pair (B_0, B_{\rightarrow}) corresponding to a network over the variables $\mathcal{X}[0], \dots, \mathcal{X}[\infty]$. In practice, we reason only about the finite interval $0, \dots, T$. To this end, we “unroll” the DBN structure into a Bayesian Network over $\mathcal{X}[0], \dots, \mathcal{X}[T]$. In slice 0, the parents of $X_i[0]$ are specified in B_0 . In slice $t+1$, the parents of $X_i[t+1]$ are nodes in slices t and $t+1$ corresponding to the parents of $X_i[1]$ in B_{\rightarrow} . In Fig. 1(c) we can see the result of unrolling the network for 3 time slices. Given a DBN, the joint distribution over $\mathcal{X}[0], \dots, \mathcal{X}[T]$ is

$$P(\mathcal{X}[0], \dots, \mathcal{X}[T]) = P_{B_0}(\mathcal{X}[0]) \prod_{t=0}^{T-1} P_{B_{\rightarrow}}(\mathcal{X}[t+1] | \mathcal{X}[t]). \quad (2)$$

2.2 Learning DBN from Complete Data

In this work, the reverse engineering of gene networks is treated as a problem of learning a DBN from complete data. The data in question is a time-series gene expression dataset \mathcal{D} consisting of K complete observation sequences

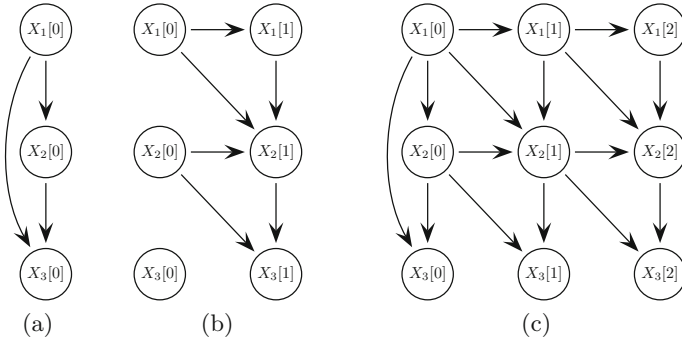


Fig. 1. (a) A prior network B_0 and (b) a transition network B_{\rightarrow} for three variables. In (c) is the corresponding “unrolled” network. Figure adapted from [4].

D_1, D_2, \dots, D_K . Each sequence D_ℓ has length N_ℓ and specifies values for the variables $\mathbf{x}^\ell[0], \mathbf{x}^\ell[1], \dots, \mathbf{x}^\ell[N_\ell]$. Thus, we have K initial instances of initial slices to learn B_0 and $N = \sum_\ell N_\ell$ instances of transitions to learn B_{\rightarrow} .

Formally, the reverse engineering problem is stated as follows. Given a dataset \mathcal{D} of instances of \mathcal{X} , find a DBN \mathcal{B} with structure $\mathcal{G} = (B_0, B_{\rightarrow})$ and its parameters Θ that best matches \mathcal{D} . We use a scoring function to measure how a DBN matches the data. The score combines the *likelihood* of the data according to the network, $L(\mathcal{B} : \mathcal{D}) = \log P(\mathcal{D} | \mathcal{B})$, with some penalty according to the complexity of the network.

BIC Score. We define the *Bayesian Information Criterion* score and the *BDe score* according to [3]. First, let us define

$$\theta_{i,j'_i,k'_i}^{(0)} = P(X_i[0] = k'_i | \mathbf{Pa}(X_i[0]) = j'_i) \tag{3}$$

and similarly

$$\theta_{i,j_i,k_i}^{\rightarrow} = P(X_i[0] = k_i | \mathbf{Pa}(X_i[0]) = j_i) \tag{4}$$

for $t = 1, \dots, T$. The notation for the sufficient statistics is given by

$$N_{i,j'_i,k'_i}^{(0)} = \sum_\ell I(X_i[0] = k'_i, \mathbf{Pa}(X_i[0]) = j'_i; \mathbf{x}^\ell) \tag{5}$$

and

$$N_{i,j_i,k_i}^{\rightarrow} = \sum_\ell \sum_t I(X_i[0] = k_i, \mathbf{Pa}(X_i[t]) = j_i; \mathbf{x}^\ell) \tag{6}$$

where $I(\cdot; \mathbf{x}^\ell)$ is an indicator function which takes on value 1 if the event \cdot occurs in \mathbf{x}^ℓ , and 0 otherwise.

Using Eq. 2 and rearranging the terms, the likelihood function is decomposed according to the DBN structure:

$$P(\mathcal{D} | \mathcal{G}, \Theta_{\mathcal{G}}) = \prod_i \prod_{j'_i} \prod_{k'_i} (\theta_{i,j'_i,k'_i}^{(0)})^{N_{i,j'_i,k'_i}^{(0)}} \times \prod_i \prod_{j_i} \prod_{k_i} (\theta_{i,j_i,k_i}^{\rightarrow})^{N_{i,j_i,k_i}^{\rightarrow}} \quad (7)$$

and the log-likelihood is given by

$$L(\mathcal{B} : \mathcal{D}) = \sum_i \sum_{j'_i} \sum_{k'_i} N_{i,j'_i,k'_i}^{(0)} \log \theta_{i,j'_i,k'_i}^{(0)} + \sum_i \sum_{j_i} \sum_{k_i} N_{i,j_i,k_i}^{\rightarrow} \log \theta_{i,j_i,k_i}^{\rightarrow}. \quad (8)$$

This decomposition facilitates the computation of the BIC and BDe scores.

Using the standard maximum likelihood estimate for multinomial distributions, we get the following expression for $\hat{\Theta}_{\mathcal{G}}$:

$$\hat{\theta}_{i,j'_i,k'_i}^{(0)} = \frac{N_{i,j'_i,k'_i}^{(0)}}{\sum_{k'_i} N_{i,j'_i,k'_i}^{(0)}}, \quad (9)$$

and similarly for the transition case. Thus, the BIC score is given by

$$\text{BIC}(\mathcal{G} : \mathcal{D}) = \text{BIC}_0 + \text{BIC}_{\rightarrow}, \quad (10)$$

where

$$\text{BIC}_0 = \sum_i \sum_{j'_i} \sum_{k'_i} N_{i,j'_i,k'_i}^{(0)} \log \hat{\theta}_{i,j'_i,k'_i}^{(0)} - \frac{\log K}{2} \#G_0 \quad (11)$$

and

$$\text{BIC}_{\rightarrow} = \sum_i \sum_{j_i} \sum_{k_i} N_{i,j_i,k_i}^{\rightarrow} \log \hat{\theta}_{i,j_i,k_i}^{\rightarrow} - \frac{\log N}{2} \#G_{\rightarrow}, \quad (12)$$

where $\#G_0$ and $\#G_{\rightarrow}$ are the number of parameters in B_0 and B_{\rightarrow} , respectively.

BDe Score. The BDe score is computed by evaluating the following integral:

$$P(\mathcal{D} | \mathcal{G}) = \int P(\mathcal{D} | \mathcal{G}, \Theta_{\mathcal{G}}) P(\Theta_{\mathcal{G}} | \mathcal{G}) d\Theta_{\mathcal{G}}. \quad (13)$$

Using Eq. 7 and assuming *Dirichlet* priors we can obtain a closed form. A Dirichlet prior for a variable X is specified by a set of *hyperparameters* $\{N'_x : x \in \text{Val}(X)\}$:

$$P(\Theta_X) = \text{Dirichlet}(N'_x : x \in \text{Val}(X)) \propto \prod_x \theta_x^{N'_x - 1}. \quad (14)$$

The hyperparameters can be interpreted as pseudo counts. Assuming that for each structure \mathcal{G} we have hyperparameters $N'_{i,j'_i,k'_i}{}^{(0)}$ and $N'_{i,j_i,k_i}{}^{\rightarrow}$ we can rewrite $P(\mathcal{D}|\mathcal{G})$ as a product of two terms

$$\prod_i \prod_{j'_i} \frac{\Gamma(\sum_{k'_i} N'_{i,j'_i,k'_i}{}^{(0)})}{\Gamma(\sum_{k'_i} N'_{i,j'_i,k'_i}{}^{(0)} + N'_{i,j'_i,k'_i}{}^{\rightarrow})} \times \prod_{k'_i} \frac{\Gamma(N'_{i,j'_i,k'_i}{}^{(0)} + N'_{i,j'_i,k'_i}{}^{\rightarrow})}{\Gamma(N'_{i,j'_i,k'_i}{}^{(0)})}, \quad (15)$$

and similar for the transition case where $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$ is the *Gamma* function satisfying $\Gamma(1) = 1$ and $\Gamma(x+1) = x\Gamma(x)$. Following [8], we assign the Dirichlet hyperparameters as

$$N'_{i,j'_i,k'_i}{}^{(0)} = \hat{N}^{(0)} \times P_{B'_0}(X_i[0] = k'_i | \mathbf{Pa}(X_i[0]) = j'_i) \quad \text{and} \quad (16)$$

$$N'_{i,j_i,k_i}{}^{\rightarrow} = \hat{N}^{\rightarrow} \times P_{B'_\leftarrow}(X_i[1] = k_i | \mathbf{Pa}(X_i[1]) = j_i), \quad (17)$$

given two *equivalent sample sizes* $\hat{N}^{(0)}$ and \hat{N}^{\rightarrow} .

2.3 Prior Biological Knowledge

The source of prior biological knowledge used is the STRING database (Search Tool for the Retrieval of Interacting Genes/Proteins) [25]. This database aims to provide a critical assessment and integration of protein-protein interactions, including direct (physical) as well as indirect (functional) associations. The current version of STRING is v10 and it covers more than 2000 organisms. Besides that, an API interface for the R computing environment is available, allowing us to easily retrieve the information about gene interactions.

The interactions are derived from multiple sources: (i) known experimental interactions are imported from primary databases, (ii) pathway knowledge is parsed from manually curated databases, (iii) automated text-mining is applied to uncover statistical and/or semantic links between proteins, based on Medline abstracts and a large collection of full-text articles, (iv) interactions are predicted *de novo* by a number of algorithms using genomic information as well as by co-expression analysis and (v) interactions that are observed in one organism are systematically transferred to other organisms, via pre-computed orthology relations [25].

For each pair of genes of a given organism, the STRING database provides a set of scores based on association evidence type: conserved neighborhood, gene fusions, phylogenetic co-occurrence, co-expression, database imports, large-scale experiments and literature co-occurrence. There is also a *combined score* expressing confidence when an association is supported by several types of evidence. For details see reference [20].

Let $s(X_i, X_j)$ be the combined score for a pair of genes obtained from STRING. First, we normalize the scores so that $0 \leq s(X_i, X_j) \leq 1$. Given that, the *biological score* is defined as the mean

$$\beta(X_i, \mathbf{Pa}(X_i)) = \frac{1}{|\mathbf{Pa}(X_i)|} \sum_{X_j \in \mathbf{Pa}(X_i)} s(X_i, X_j). \quad (18)$$

The decomposition in Eq. 8 facilitates the computation of the BIC and BDe scores. Notice that the likelihood is expressed as a sum of terms, where each term depends only on the conditional probability of a variable X_i given its parents $\mathbf{Pa}(X_i)$. Thus, when computing the BIC or BDe score, we add the term $\omega \cdot \beta(X_i, \mathbf{Pa}(X_i))$, where $\omega \in \mathbb{R}$ is the weight given to the prior biological knowledge. Obviously, $\omega = 0$ means that we are not taking into account the biological knowledge.

2.4 Validation

To validate and analyze the methodology, assume that \mathcal{G}^* is the gold standard network and \mathcal{G} is the output network. Then, it is possible to fill the entries of a *confusion matrix* [28] according to Table 1. According to this table, if an edge/connection is present in the gold standard and it is correctly inferred, then it counts as a true positive (TP).

Table 1. Confusion matrix. TP = true positive; FP = false positive; FN = false negative; TN = true negative.

Edge	Inferred in \mathcal{G}	Not inferred in \mathcal{G}
Present in \mathcal{G}^*	TP	FN
Absent in \mathcal{G}^*	FP	TN

To quantify the quality of the output network, we used a similarity [1] measure defined as

$$\text{Similarity}(\mathcal{G}^*, \mathcal{G}) = \sqrt{\text{PPV} \times \text{Specificity}}, \quad (19)$$

where

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{and} \quad \text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \quad (20)$$

2.5 Implementation

The methodology was implemented by extending an open source package named `Pgmpy` (<http://pgmpy.org>), which is a Python library for working with probabilistic graphical models. This is an ongoing project and for DBN only the parameters inference was implemented. We extended the library by implementing the learning of DBN structure given a gene expression dataset and a source of prior biological knowledge provided by the STRING database.

3 Results and Discussion

In order to apply the proposed methodology, we performed experiments using data from the DREAM4 Challenge (Dialogue for Reverse Engineering Assessments and Methods) [2] and a yeast (*Saccharomyces cerevisiae*) cell cycle study [24]. All datasets used were discretized using an algorithm called Bik-means [16].

3.1 DREAM4 Data Results

The DREAM4 *in silico* challenge data consists of a time series generated from networks of 10 and 100 genes. There are 5 networks of size 10 and for each of them there are 5 datasets of 21 time points. For the network of size 100 the number of networks is also 5, but there are 10 datasets of 21 time points each. This experiment validates the methodology without the use of prior biological knowledge, once it consists of synthetic data.

In Fig. 2 we compare the standard Bayesian Network (BN), the Bayesian Network considering the time factor (BN with time), and Dynamic Bayesian Network (DBN) by using the BIC score. The BN with time simply means that, when estimating the conditional probabilities for a gene $X_i[t]$, the values of its parents in the samples are taken at time $t+1$. Networks from 1 to 5 are composed by 10 genes and from 6 to 10 by 100 genes. We can observe that DBN performs better in most cases. In Fig. 3 we performed the same comparison, but using

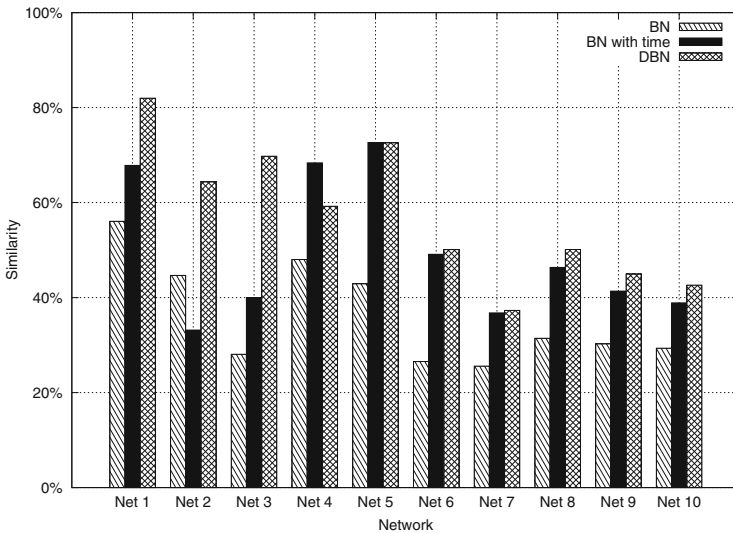


Fig. 2. Comparison between the standard Bayesian Network and the Dynamic Bayesian Network, using the BIC score.

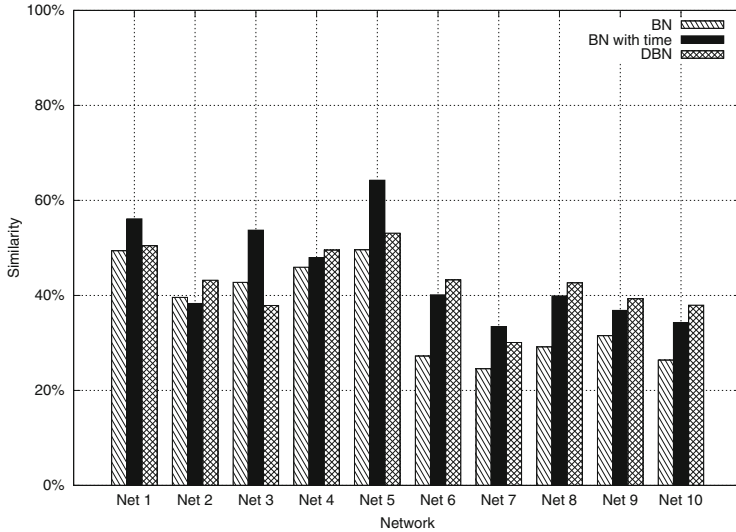


Fig. 3. Comparison between the standard Bayesian Network and the Dynamic Bayesian Network, using the BDe score.

the BDe score. In some cases DBN presented better results, but in others the BN with time was better.

We also compared the true positive rate between BIC and BDe scores. In Fig. 4 we can observe that the BDe score performs slightly better in most experiments. From 1 to 10 we used BN with time, where from 1 to 5 are the networks of size 10 genes and, from 6 to 10 are networks of size 100. From 11 to 20 we used DBN, where from 11 to 15 are networks of size 10 and, from 16 to 20 of size 100.

3.2 Yeast Cell Cycle Data Results

We tested the methodology using a dataset of yeast cell cycle gene expression provided by [24]. This dataset was obtained from six experiments, named *cln3* (2 time points), *clb2* (2), *alpha* (18), *cdc15* (24), *cdc28* (17) and *elutriation* (14). In this experiment, we selected 11 genes considered to be key regulators of the cell cycle process according to [15]. The gold standard network was obtained from the KEGG database.

In Fig. 5 we show the similarities values when using the BIC score and the prior biological knowledge for $\omega = 1, 2, 4$. The `completeData` bars was generated using all six experiments together. Because the experiments *cln3* and *clb2* are composed by only 2 samples each we did not use them individually. We can observe that the biological knowledge improves the quality of the output network specially in the *cdc28* and *alpha* experiments.

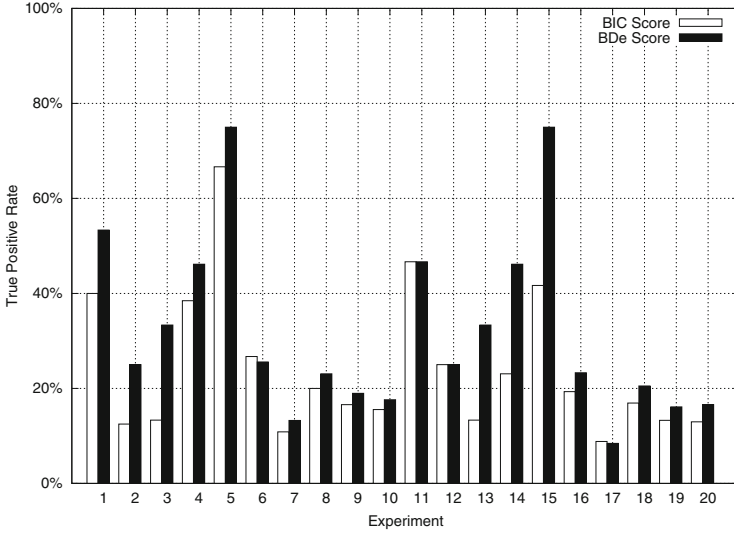


Fig. 4. Comparing the true positive rates obtained when using BIC and BDe score for the DREAM4 data.

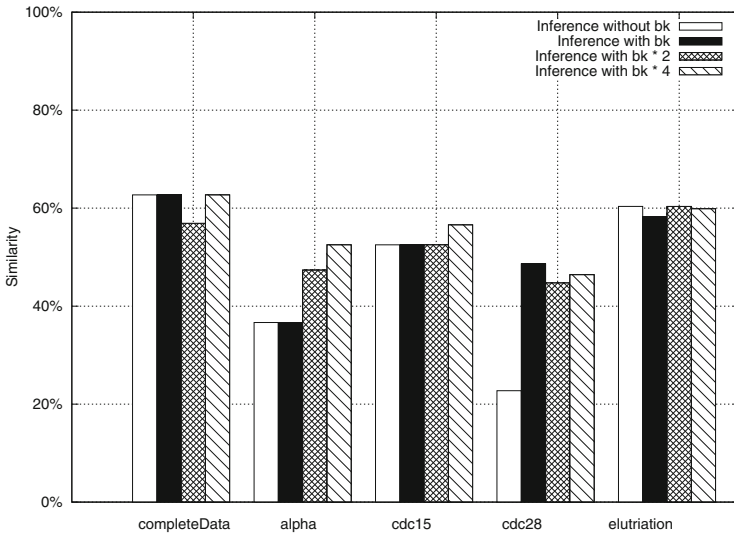


Fig. 5. Similarity of the yeast cell cycle network using BIC score and considering biological knowledge with $\omega = 1, 2, 4$.

In Fig. 6 the show the results when using the same data, but with the BDe score. We can observe that the similarity values are slightly worst when compared to the BIC score experiment. It is also possible to notice a soft improvement of the quality when considering the biological knowledge.

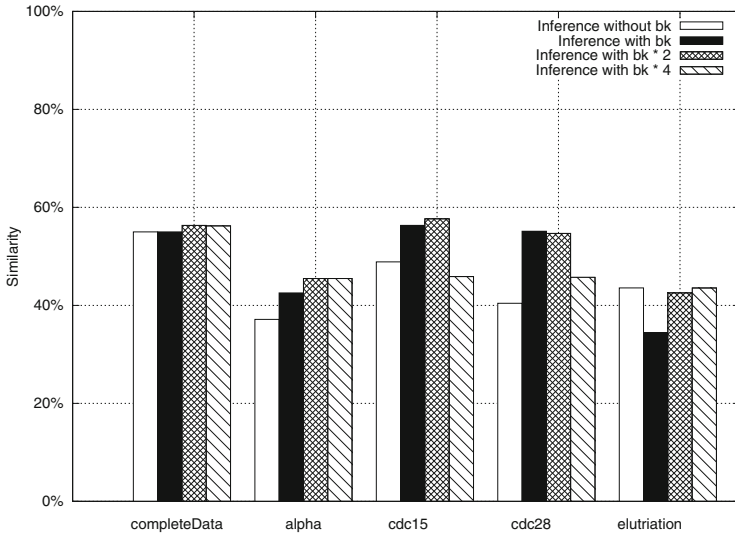


Fig. 6. Similarity of the yeast cell cycle network using BDe score and considering biological knowledge with $\omega = 1, 2, 4$.

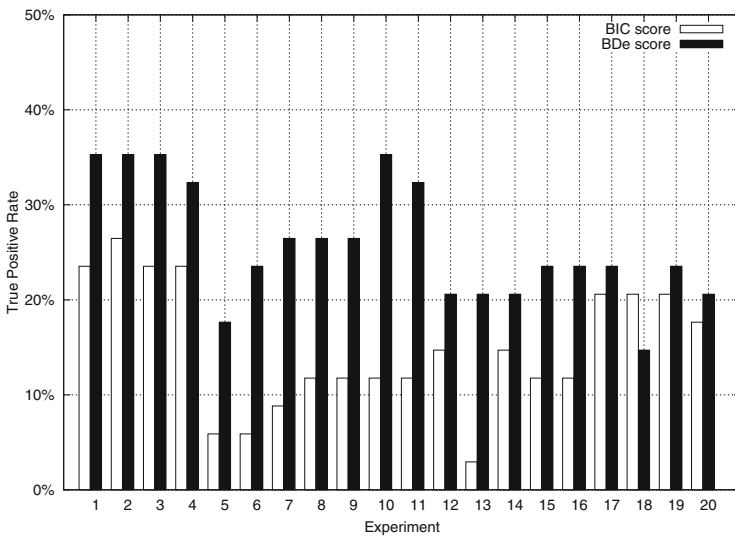


Fig. 7. Comparing the true positive rates obtained when using BIC and BDe score for the yeast cell cycle data.

Comparing the true positive rates between BIC and BDe scores shows that BDe performs better in most of our experiments, as we can observe in Fig. 7. In experiments 1 to 4 we used the `completeData` with biological knowledge plus $\omega = 0, 1, 2, 4$, respectively. In experiments 5 to 8 the `alpha` dataset was used,

following the same pattern of ω . Experiments 9 to 12 correspond to the cdc15 dataset, 13 to 16 correspond to the cdc28 dataset, and from 17 to 20 to the elutriation dataset.

4 Conclusion

In this work, we proposed a methodology for the reverse engineering of gene regulatory networks from time series gene expression using dynamic bayesian networks as well. We also proposed the use of a source of prior biological knowledge, known as STRING, to improve the quality of the inferred networks. The methodology was tested on two datasets: artificial dataset provided by the DREAM4 *in silico* challenge and a biological dataset from a yeast cell cycle study.

To learn the structure of the DBN we search for the network that best matches the data, based on a score function. We used two types of functions, BIC and BDe scores, and compared them. The results show that BDe score performs better in most of the experiments.

We also compared the standard Bayesian Network model against the Dynamic Bayesian Network, showing that the DBN provides better results than its standard version. One important point to highlight is that our methodology considers a source of prior biological knowledge, when available, unlike most of the reverse engineering algorithms. We showed through our experiments that this approach can improve the quality of the inferred networks. As a future work, we can explore how to use various sources of biological knowledge instead of only one.

Acknowledgement. This research was supported by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) and UFMS (Universidade Federal de Mato Grosso do Sul).

References

1. Dougherty, E.R.: Validation of inference procedures for gene regulatory networks. *Curr. Genomics* **8**(6), 351–359 (2007)
2. DREAM: DREAM: Dialogue for Reverse Engineering Assessments and Methods (2009). <http://wiki.c2b2.columbia.edu/dream/>
3. Friedman, N., Murphy, K., Russell, S.: Learning the structure of dynamic probabilistic networks. In: 15th Annual Conference on Uncertainty in Artificial Intelligence, pp. 139–147. Morgan Kaufmann (1999)
4. Friedman, N., et al.: Using Bayesian networks to analyze expression data. *J. Comput. Biol.* **7**(6), 601–620 (2000)
5. Gao, S., Wang, X.: Quantitative utilization of prior biological knowledge in the Bayesian network modeling of gene expression data. *BMC Bioinf.* **12**(1), 359+ (2011)
6. Goodwin, B.C.: *Temporal Organization in Cells; A Dynamic Theory of Cellular Control Process*. Academic Press, Cambridge (1963)
7. Hecker, M., et al.: Gene regulatory network inference: data integration in dynamic models - a review. *BioSystems* **96**, 86–103 (2009)

8. Heckerman, D., Geiger, D., Chickering, D.: Learning Bayesian networks: the combination of knowledge and statistical data. *Mach. Learn.* **2**(3), 197–243 (1995)
9. Huang, S., Ernberg, I., Kauffman, S.: Cancer attractors: a systems view of tumors from a gene network dynamics and developmental perspective. *Semin. Cell Dev. Biol.* **20**(7), 869–876 (2009)
10. Kanehisa, M., Goto, S., Kawashima, S., Nakaya, A.: The KEGG databases at GenomeNet. *Nucleic Acids Res.* **30**, 42–46 (2002)
11. Karlebach, G., Shamir, R.: Modelling and analysis of gene regulatory networks. *Nature* **9**, 770–780 (2008)
12. Kauffman, S.A.: Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* **22**(3), 437–467 (1969)
13. Koike, C.Y., Higa, C.H.A.: Inference of gene regulatory networks using coefficient of determination, Tsallis entropy and biological prior knowledge. In: *Proceedings of the IEEE 16th International Conference on Bioinformatics and Bioengineering* (2016)
14. Koller, D., Friedman, N.: *Probabilistic Graphical Models: Principles and Techniques*, 1st edn. MIT Press, Cambridge (2012)
15. Li, F., Long, T., Lu, Y., Ouyang, Q., Thang, C.: The yeast cell-cycle network is robustly designed. *PNAS USA* **101**(14), 4781–4786 (2004)
16. Li, Y., Liu, L., Bai, X., Cai, H., Ji, W., Guo, D., Zhu, Y.: Comparative study of discretization methods of microarray data for inferring transcriptional regulatory networks. *BMC Bioinf.* **11**(1), 520+ (2010)
17. Linde, J., Schulze, S., Henkel, S.G., Guthke, R.: Data and knowledge-based modeling of gene regulatory networks: an update. *EXCLI J.* **14**, 346–378 (2015)
18. Locke, J.C.W., Millar, A.J., Turner, M.S.: Modelling genetic networks with noisy and varied experimental data: the circadian clock in *Arabidopsis thaliana*. *J. Theor. Biol.* **234**(3), 383–393 (2005)
19. Madhamshettiwar, P.B., Maetschke, S.R., Davis, M.J., Reverter, A., Ragan, M.A.: Gene regulatory network inference: evaluation and application to ovarian cancer allows the prioritization of drug targets. *Genome Med.* **4**(5), 1–16 (2012)
20. von Mering, C., Jensen, L.J., Snel, B., Hooper, S.D., Krupp, M., Foglierini, M., Jouffre, N., Huynen, M.A., Bork, P.: STRING: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic Acids Res.* **33**(Database issue), D433–D437 (2005)
21. Russell, S.J., Binder, J., Koller, D., Kanazawa, K.: Local learning in probabilistic networks with hidden variables. In: *IJCAI*, pp. 1146–1152 (1995)
22. Schena, M., Shalon, D., Davis, R.W., Brown, P.O.: Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* **270**(5235), 467–470 (1995)
23. Schwarz, G.: Estimating the dimension of a model. *Ann. Stat.* **6**(2), 461–464 (1978)
24. Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell* **9**, 3273–3297 (1998)
25. Szklarczyk, D., Franceschini, A., Wyder, S., Forslund, K., Heller, D., Huerta-Cepas, J., Simonovic, M., Roth, A., Santos, A., Tsafou, K.P., Kuhn, M., Bork, P., Jensen, L.J., von Mering, C.: STRING v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.* **43**(Database issue), D447–D452 (2015)
26. Wang, Z., Xu, W., Lucas, F., Liu, Y.: Incorporating prior knowledge into gene network study. *Bioinformatics* **29**(20), 2633–2640 (2013)

27. Wang, Z., Gerstein, M., Snyder, M.: RNA-Seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.* **10**(1), 57–63 (2009)
28. Webb, A.R.: *Statistical Pattern Recognition*, 2nd edn. Wiley, Hoboken (2002)
29. Werhli, A.V., Husmeier, D.: Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Stat. Appl. Genet. Mol. Biol.* **6**(1), 15 (2007)