

Combining Gene Expression Data and Prior Knowledge for Inferring Gene Regulatory Networks via Bayesian Networks Using Structural Restrictions

Luis M. de Campos, Andrés Cano, Javier G. Castellano, and Serafín Moral

Department of Computer Science and Artificial Intelligence
University of Granada, Granada, Spain
{lci,acu,fjgc,smc}@decsai.ugr.es

Abstract. Gene Regulatory Networks (GRNs) are known as the most adequate instrument to provide a clear insight and understanding of the cellular systems. One of the most successful techniques to reconstruct GRNs using gene expression data is Bayesian networks (BN) which have proven to be an ideal approach for heterogeneous data integration in the learning process. Nevertheless, the incorporation of prior knowledge has been achieved by using prior beliefs or by using networks as starting point in the search process. In this work, the utilization of different kinds of structural restrictions within algorithms for learning BNs from gene expression data is considered. These restrictions will codify prior knowledge, in such a way that a BN should satisfy them. Therefore, one aim of this work is to make a detailed review on the use of prior knowledge and gene expression data to inferring GRNs from BNs, but the major purpose in this paper is to research whether the structural learning algorithms for BNs from expression data can achieve better outcomes exploiting this prior knowledge with the use of structural restrictions. In the experimental study, it is shown that this new way to incorporate prior knowledge lead us to achieve better reverse-engineered networks.

Keywords: Bayesian networks, genetic regulatory networks, microarray data, structural restrictions, prior knowledge

1 Introduction

During past decades, several computational methodologies have been introduced to reconstruct gene regulatory networks (GRNs) using functional genomic data obtained by high-throughput technologies such as microarray or RNAseq (Chai et al., 2014; Oates, Amos, & Spencer, 2014; Y. R. Wang & Huang, 2014; Lee & Tzou, 2009; Esteves & Reis, 2018). This process of reconstructing GRNs using gene expression data is also known as network inference or reverse engineering. GRNs represent the regulatory interactions between genes and they provide useful information for drug-design or medical-related fields.

Gene-gene interactions can be inferred from gene expression data to reconstruct a gene regulatory network (Chai et al., 2014). For this purpose Bayesian networks (BNs) (Pearl, 1988) represent one of the most successful formalism (Wit & McClure, 2004) and constitute a widely accepted methodology to deal with uncertain knowledge.

BNs also have analytical and diagnostic capabilities that make them very suitable for inferring GRNs. A BN or belief network can be described as a probabilistic graphical representation of a joint probability distribution. The BN is composed of a directed acyclic graph (DAG), a qualitative part, and the numerical parameters of the model, usually conditional probability tables, a quantitative part. Their success is founded on their probabilistic nature and their ability to deal with uncertainty, being capable to handle noisy data (Chai et al., 2014) or missing data or find hidden variables (Linde, Schulze, Henkel, & Guthke, 2015). Another important aspect is the capacity of BNs to integrate prior biological knowledge (Linde et al., 2015) and to capture multiple types of relationships (Yu, Smith, Wang, Hartemink, & Jarvis, 2004).

A BN can be built manually from an expert but the common practice is to obtain it automatically from a data set. Therefore, the task of learning automatically a Bayesian network from a data set is to obtain the network that best represents the probability distribution in the data. Since a BN is composed of a qualitative part and a quantitative part, we distinguish two steps: (i) structural learning; and (ii) parameter learning.

Because parameter learning consists of estimating the conditional probabilities given by the structure of the graph using the observed frequencies on the data, we first must learn the topology of the network. The conditional probabilities can be computed by using a maximum likelihood estimation (Buntine, 1996), though it is normally done by using a Bayesian estimator based on the Dirichlet distribution (Heckerman, Geiger, & Chickering, 1995).

There is a lot of works on the automatic learning from data of the BN structures and, as a result, many structural learning methods have been developed that can be categorized into two broad groups: algorithms using conditional independence tests, and algorithms using a heuristic search and a metric or scoring function.

Learning methods based on conditional independence tests (also called constraint-based algorithms) (Spirtes, Glymour, & Scheines, 1993; Cheng, Greiner, Kelly, Bell, & Liu, 2002; de Campos & Huete, 2000) perform a qualitative study of the dependence and independence relationships between the variables. The aim of these methods is to find the network that best match these relationships by using conditional independence tests. The most telling example of this kind of structural learning is the PC algorithm (Spirtes et al., 1993) which, starting with a complete graph, first eliminates as many edges as possible, and then it gives direction to the edges. The elimination of edges is guided by conditional independence statistical tests which are applied to variables of the data.

The second type of structural learning algorithms attempt to find the graph that best represents the data by maximizing the selected metric and minimizing the number of arcs. The metric or scoring function is a measure of relation between the data and the developing graph. There are several proposals based on Bayesian scoring functions, such as BD/BDe metric (Heckerman et al., 1995), BDeu metric (Buntine, 1991) or K2 (Cooper & Herskovits, 1992) and other approaches based on information theory scoring functions, such as entropy (Chow & Liu, 1968) or the Minimum Description Length (Lam & Bacchus, 1994). Furthermore, a search procedure is needed to find the best structures according to the selected metric. Local search methods are commonly used (Cooper & Herskovits, 1992; Heckerman et al., 1995) due to the exponentially large size of the search space.

The nature of gene expression data and its acquisition is a task that is subject to *the curse of dimensionality* (Bellman, 1957): A huge number of variables (genes) and

very few cases (it is an expensive test) (Njah & Jamoussi, 2015). Thus, many BNs structures represent equally well this high dimensional data (Y. R. Wang & Huang, 2014). Furthermore, the structural learning of a BN is challenging due to the vast space of possible DAGs for even a moderate number of variables. For these reasons, several authors have paid attention on the use of additional biological knowledge for learning BNs from gene data. It has been observed that the combined use of gene expression data and prior knowledge enhances the accuracy and the quality of the gene regulatory network (Lee & Tzou, 2009) and improves the reliability of predicted gene interactions (Linde et al., 2015; Zhou & Zheng, 2014).

As will be discussed later, most of the existing works which employ prior knowledge in structural learning of the BN use a prior probability distribution (Heckerman et al., 1995) (i.e. prior beliefs) for each DAG as a function of the available biological knowledge, but also the prior knowledge is utilized to build an initial network structure used to initiate the learning search. But the use of prior knowledge in the manner that an expert would do it in a graphical network, that is, with the specification of structural restrictions that should verify the network, has not been thoroughly studied.

Hence, our contribution is two-fold. First, we will carry out a profound review of the literature about the use of prior knowledge and gene expression data for reverse engineering of GRNs using BNs. Second, we address the employment of additional prior knowledge by describing some types of structural restrictions, which must codify several kinds of biological knowledge and must be employed within structural learning algorithms for BN. In that regard, we are going to consider three types of structural constraints: first, existence restrictions which force the presence of arcs and/or edges; second, absence restrictions which force the prohibition of arcs and/or edges; third, ordering structural restrictions which force a partial order between some variables. This restrictions will be assumed as “hard” restrictions (in opposite to “soft” restrictions (Heckerman et al., 1995)); therefore, the BN always must satisfies them and all the intermediate networks obtained in the learning process have to satisfy them. An existence restriction may indicate that a gene expression is regulated positively or negatively by other, or may indicate that both genes are coregulated. An absence restriction can be viewed as an nonexistent direct interactions of any kind between two genes. An ordering restriction may represent a group of two or more genes in the same regulatory pathway.

By using prior knowledge as structural restrictions, our experimental results reveal an improvement in the networks obtained, moreover, the increase is proportional to the amount of prior knowledge utilized.

The paper is structured as follows: Section 2 presents the problem of obtaining GRNs from BNs. In Section 3 we study the use of prior knowledge and gene expression data to learn BNs. In Section 4 we shall present some preliminary background about structural learning of BNs and the types of structural restrictions to be considered. Section 5 discusses the experimental results. Finally, Section 6 is devoted to the conclusions.

2 Inferring gene regulatory networks from Bayesian networks

Learning BNs from gene expression data has several issues (Spirtes et al., 2000), being the main challenge the curse of dimensionality, but the gene expression is also influenced by missing values, noise, measurement errors and the fact that gene expression is not a nominal value, it is a non-normalized continuous value.

A gene regulatory network (GRN), or simply genetic network, is a graph representation of the regulatory interactions between genes. Genes can be viewed as nodes in the network and connections represent the intricate mechanisms that regulate gene expression. A link between two genes may indicate that a gene expression is regulated positively or negatively by the other, or may indicate that both genes are coregulated. Many different approaches have been developed to model and inferring GRNs (Styczynski & Stephanopoulos, 2005; Markowitz & Spang, 2007; Schlitt & Brazma, 2007; Chai et al., 2014; Aderhold, Husmeier, & Grzegorzczuk, 2014; Banf & Rhee, 2017). However BNs have been one of the most successful approach for learning GRNs from gene expression data (Wit & McClure, 2004; Sachs, Perez, Pe'er, Lauffenburger, & Nolan, 2005; Bansal, Belcastro, Ambesi-Impiombato, & di Bernardo, 2007; Friedman, 2004), although BNs have a higher computational cost (Werhli, Grzegorzczuk, & Husmeier, 2006).

Obtaining a simplified model of a GRN from a BN is immediate: in the BN the nodes represent the genes of the genetic network and the links represent regulatory relations (activation or inhibition) as in GRNs. Although there are some differences: first, a BN cannot have cycles which is a common feature in many pathways since they are acyclic but this limitation can be overcome by using dynamic Bayesian networks. Second, a link may exist in the BN but not in the GRN, in this case, the link indicates that both genes are expressed at the same time, i.e. they are coregulated. Third, an edge in the GRN may indicate different kinds of relations (Schlitt & Brazma, 2007).

We should bear in mind that BNs represent probabilistic dependencies among variables and not causality, that is, correlation does not imply causality. Nevertheless, in the case of inferring GRN, “we can interpret the edge as a causal link if we assume that the Causal Markov Condition holds” (Bansal et al., 2007). The Markov condition for a BN states that each variable is conditionally independent of its nondescendants given its parents in the BN.

3 On the use of prior knowledge for learning BNs from expression data

To handle the curse of the dimensionality some authors have developed adapted algorithms for learning BNs (Acid, de Campos, & Fernández, 2013; Gámez, Mateo, & Puerta, 2011). However, other authors have proposed to use expert knowledge or any other additional biological knowledge. This kind of knowledge is called *prior knowledge*.

For example, (Segal, Wang, & Koller, 2003) build a Gaussian naïve Bayes for gene expression data. After that, they build a Markov network for protein-protein interactions data. Finally, the two networks are joined and the parameters are learned with the EM algorithm.

(Gifford, 2001) intends to use graphical models for gene expression data as a parameter-free tool capable of modelling genetic networks. The practical application comes by (Hartemink, Gifford, Jaakkola, & Young, 2001, 2002a) where they build two BNs by hand using only expert knowledge. After that they use a Bayesian score to compare the two networks. Also they present annotated edges to represent additional biological information about the type of dependence relationship between variables. They modify the metric to include annotated edges. Same authors (Hartemink, Gifford, Jaakkola, & Young, 2002b) present a method to obtain GRNs using expression data and genomic location data. The location data is used to influence the BN prior and expression data to influence the network likelihood.

(Zhu et al., 2008) combine genotypic data and gene expression data to build BNs. Additionally they include transcription factor binding site and protein-protein interaction data. The BN is constructed using this additional data as priors.

Preliminary networks derived from the literature and/or protein-protein interaction data are used by (Djebbari & Quackenbush, 2008) to build a network which is used as starting point of a local search.

Using a continuous approach, (Tamada et al., 2003b, 2003a) integrate microarray gene expression data and DNA sequence information into a BN. Their basic idea is: if a parent gene is a transcription factor, its children may share a consensus motif in their promoter regions of the DNA sequences. The BN is learned using the approach presented by (Imoto, Kim, et al., 2003).

(Imoto et al., 2004; Imoto, Higuchi, et al., 2003) propose to use more sources of biological knowledge: microarray gene expression, protein-protein interactions, protein-DNA interactions, binding site information and existing literature. The biological knowledge is added into the prior probability of the metric. (Werhli & Husmeier, 2007b, 2007a) make a variation of (Imoto, Higuchi, et al., 2003), using other learning algorithm. (Nariai, Kim, Imoto, & Miyano, 2004) use only protein-protein interactions as additional knowledge following the methodology presented by (Imoto, Higuchi, et al., 2003). After the network is built, they evaluate it with a genetic network from KEGG (Kyoto Encyclopedia of Genes and Genomes) (Kanehisa et al., 2008). (Nariai, Tamada, Imoto, & Miyano, 2005) generate a GRN, and a network of protein-protein interactions.

In contrast, (Le Phillip, Bahl, & Ungar, 2004) do not use information from other data sets, a network is built by an expert based on the existing literature. With a score+search approach, the authors use hard constraints of absence and existence from the expert network, i.e. they limit the search space with links that should necessarily be in the candidate networks and those that are prohibited. The network obtained by the expert is used to generate the synthetic data for validating their proposal. (Chen, Cairelli, Kilicoglu, Shin, & Rindfleisch, 2014) utilized knowledge extracted from literature as a network of interactions and then apply expression data. Using also previous literature as prior knowledge, (Steele, Tucker, Hoen, & Schuemie, 2009) obtain gene association scores as prior beliefs in the BN and they also investigate the effect of weighting the effect of prior knowledge. Likewise (Li, Wu, & Zhang, 2006) also combine literature extracted knowledge and microarray data.

(Almasri, Larsen, Chen, & Dai, 2008) use a GRN made previously by another author (Larsen, Almasri, Chen, & Dai, 2007). They use the structure of the known network as the starting point for the learning algorithm. Also they use the expert GRN to reduce the search space.

(Mukherjee & Speed, 2008) incorporate prior knowledge into BN inference. (Isci, Dogan, Ozturk, & Otu, 2014) proposed a methodology to incorporate several sources of prior knowledge, regardless of its type, in the prior distributions of the BN learning algorithm. Similarly, (Sauta, Demartini, Vitali, Riva, & Bellazzi, 2017) proposed a two step procedure where the additional knowledge is integrated in the model as a prior probability.

4 Bayesian networks using structural restrictions

Formally, let us consider a finite set of discrete random variables $\mathcal{V} = \{x_1, x_2, \dots, x_n\}$, each variable taking on values from a finite set, a Bayesian network is a pair (G, θ) , where:

- G is a directed acyclic graph (DAG) $G = (\mathcal{V}, E_G)$, being E_G the set of arcs. In this graph, a link represents direct dependence relationships between the variables. For example, $x_j \rightarrow x_i$ indicates that the node x_i directly depends on the node x_j .
- θ is a set of conditional probability distributions. This set contains, for each node, a conditional probability distribution of the variables on which it depends directly, i.e, its parents. If a node has no parents, its distribution is simply the probability distribution of the node.

The structural learning task of a BN from a data set \mathcal{D} can be described as the search of the DAG that, in some ways, best represents the data set.

Structural learning methods can be categorized into two broad groups: conditional independence tests based algorithms, and score+search methods (for references, see (Acid & de Campos, 2003)). In both cases our proposal must reduce the corresponding search space (which is hyper-exponential) by introducing several types of restrictions that the elements in this space must verify.

The structural restrictions approach used in the experiments was presented properly by (de Campos & Castellano, 2007), where three kinds of structural constraints were described for the variables. These restrictions can be represented graphically using arcs and edges (an edge can be seen as an unoriented arc):

- *Existence restrictions* (E_e): if an arc $x \rightarrow y \in E_e$, it must be true in any DAG of the search domain. If an edge $x-y \in E_e$ then $y \rightarrow x$ or $x \rightarrow y$ have to exist in every graph structure of the search domain.
- *Absence restrictions* (E_a): if $x \rightarrow y \in E_a$, it is forbidden for every graph of the search domain. If $x-y \in E_a$, $y \rightarrow x$ or $x \rightarrow y$ can not exist.
- *Partial ordering restrictions* (E_o): if $x \rightarrow y \in E_o$, it means that $x < y$, and every DAG structure in the search domain should verify that x antecedes y in some total order of \mathcal{V} consistent with the DAG structure. These ordering restrictions can be used to express, for instance, some kind of functional or temporal precedence between variables. Take into account if $(x, y) \in \mathcal{R}_o$ then the DAG can not contain the arc $y \rightarrow x$, nor a directed path from the variable x to the variable y .

Therefore, for inferring GRNs, an existence constraint can be identified as some kind of regulatory relationship between genes. If the existence constraint is directed, it can indicate that a gene is regulated positively or negatively by the other gene in the restriction. An absence constraint may indicate that there is not any directed regulatory relationship between two genes. An order constraint may indicate some relation between genes, if $gen_1 < gen_2$ indicates that gen_1 is expressed before gen_2 , for example, the gen_1 is before gen_2 in the same genetic regulatory pathway.

In the following subsections, the use of structural restrictions will be presented for a local search algorithm, specifically hill climbing, guided by a scoring function and the PC algorithm. Both are prime examples of the two mentioned approaches for BNs structure learning.

4.1 Structural restrictions in a local search algorithm based on a scoring function

Structural learning algorithms used in the score+search paradigm are usually based on the application of operators to move within the search space. The same basic operators are used in many search algorithms: (1) addition of arcs, and (2) removal of arcs (another operator used is reversal of arcs, but this operator can be seen as remove the arc and add the arc in the opposite direction).

To use structural restrictions in a local search algorithm, we must start from a DAG G , which must verify the restrictions. The initial point in this kind of algorithms is an empty DAG G_\emptyset . As regards this specific case, G_\emptyset should be substituted by the DAG obtained from E_e . However, we should bear in mind that E_e does not necessarily represent a DAG, and thus E_e must be transformed into a DAG. Let G' be a DAG obtained from G as a result of the application of one of the earlier mentioned operators. We are going to consider what are the requirements necessary and sufficient to guarantee that G' also verifies the structural restrictions.

Let $G_e = (\mathcal{V}, E_e)$, $G_a = (\mathcal{V}, E_a)$ and $G_o = (\mathcal{V}, E_o)$ be graphs describing self-consistent existence, absence and ordering restrictions¹, respectively, and let $G = (\mathcal{V}, E_G)$ be a DAG which verifies them.

- (I) Arc addition: Let $G' = (\mathcal{V}, E'_G)$, $E'_G = E_G \cup \{x \rightarrow y\}$, with $x \rightarrow y \notin E_G$. In that case, G' is a graph which verifies G_e , G_a and G_o given
 - $x-y \notin E_a$ and $x \rightarrow y \notin E_a$,
 - there is no directed path from y to x in $G \cup G_o$.
- (II) Arc removal: Let $G' = (\mathcal{V}, E'_G)$, $E'_G = E_G \setminus \{x \rightarrow y\}$, with $x \rightarrow y \in E_G$. In that case, G' is a graph which verifies G_e , G_a and G_o given
 - $x \rightarrow y \notin E_e$ and $x-y \notin E_e$.

4.2 Structural restrictions in the PC algorithm

The PC algorithm (Spirtes et al., 1993) first eliminates as many edges as possible, and second, it gives direction to several of the remaining unoriented arcs, i.e. edges, trying to obtain head to head patterns. Both activities are guided by conditional independence statistical tests applied to the available dataset.

In this situation, an algorithm to use structural restrictions, appropriate to diminish the amount of necessary statistical tests to apply, is as follows: we should check whether the resulting graph after carrying out this operation verifies the structural restrictions before computing a conditional independence test; if the verification test fails, the statistical test should not be computed. Let us specify the verification tests for the operators being applied.

Let G_e , G_a and G_o be the aforementioned graphs, and let $H = (\mathcal{V}, E_H)$ be a partially directed acyclic graph (PDAG) which verifies G_e , G_a and G_o .

- (I) Arc removal: Let $H' = (\mathcal{V}, E'_H)$, $E'_H = E_H \setminus \{x \rightarrow y\}$, with $x \rightarrow y \in E_H$. In that case, H' is a PDAG which verifies G_e , G_a and G_o given

¹ Some interactions can occur among the structural restrictions and may give rise to incongruencies. Thus, it is needed to confirm that these structural restrictions can indeed be verified. In this regard, we shall state that a collection of restrictions is self-consistent if there is some DAG that verifies them (see (de Campos & Castellano, 2007) for more details).

- $x \rightarrow y \notin E_e$ and $x - y \notin E_e$.
- (II) Edge removal: Let $H' = (\mathcal{V}, E'_H)$, $E'_H = E_H \setminus \{x - y\}$, with $x - y \in E_H$. In that case, H' is a PDAG which verifies G_e, G_a and G_o given
- $x \rightarrow y \notin E_e$, $y \rightarrow x \notin E_e$ and $x - y \notin E_e$.
- (III) Head-to-head addition: Let $x, y, z \in \mathcal{V}$ and define a subset of links S as either $S = \{x \rightarrow z, z - y\}$, $S = \{x - z, y \rightarrow z\}$ or $S = \{x - z, z - y\}$. If x and y are not adjacent in H and $S \subseteq E_H$, let $H' = (\mathcal{V}, E'_H)$, with $E'_H = (E_H \setminus S) \cup \{x \rightarrow z, y \rightarrow z\}$. In that case, H' is a PDAG which verifies G_e, G_a and G_o given
- $x \rightarrow z \notin E_a$ and $y \rightarrow z \notin E_a$,
 - there is no directed path from z to x nor is there a directed path from z to y in $H \cup G_o$.
- (IV) Edge orientation: Let $H' = (\mathcal{V}, E'_H)$, $E'_H = (E_H \setminus \{x - y\}) \cup \{x \rightarrow y\}$, with $x - y \in E_H$. In that case, H' is a PDAG which satisfies G_e, G_a and G_o given
- $x \rightarrow y \notin E_a$,
 - there is no directed path from y to x in $H \cup G_o$.

The initial point for the PC algorithm is the complete undirected graph. In our case this starting graph should be converted giving direction to some edges in accordance with G_e, G_a and G_o and by removing the edges in accordance with G_a .

5 Experiments

In this section, the effects of prior knowledge in BNs learned from gene expression data will be studied. In order to do this, we rely on the fact that BNs are visually interpretable, consequently, we will try to incorporate prior knowledge graphically, and in accordance with the interpretability of the model, using structural constraints.

5.1 Gene expression data set

We are going to study the use of structural constraints from prior knowledge with DNA microarrays data of the budding yeast *Saccharomyces cerevisiae*. The data set (available at <http://genome-www.stanford.edu/cellcycle/>) that we are going to use comes from the study of the budding yeast cell cycle by (Spellman et al., 1998). The dataset also includes data from (Cho et al., 1998).

In the work of Spellman et al., the expressed genes during the yeast cell cycle were studied using DNA microarrays. In the experiments, 6200 gene expressions (virtually all yeast genes) were measured in total of 77 time points under four different experimental conditions, identifying gene expressions in different cell cycle phases. Spellman and coworkers identified 800 genes involved in the *Saccharomyces cerevisiae* cell cycle.

5.2 Prior knowledge

Without access to an expert in the field, the solution adopted to obtain prior knowledge was looking into previous research for support. Specifically, we have used the GRN of the yeast cell cycle obtained by (Nariai et al., 2005). To build this network, Nariai and coworkers used the gene expression data from the works of (Spellman et al., 1998) and (Hughes et al., 2000). They also obtained knowledge from the MIPS database (Mewes

et al., 2002)(MIPS is the database of genomes and protein sequences of the Munich Information Centre for protein sequences. Available at <http://mips.gsf.de/>).

We are going to suppose that the GRN obtained in the research of (Nariai et al., 2005), is the real genetic network that represents the yeast cell cycle for some genes. We will use this network, but the starting graph should be modified by giving direction to the arcs without orientation, i.e. the edges. This edge orientation was carried out without creating directed cycles and avoiding to produce extra head to head patterns as far as possible. The obtained network has 55 nodes and 100 arcs and will be called the *true network*.

5.3 Preprocess

To work with the yeast cell cycle data set provided by Spellman et al., it has been necessary a data preprocessing stage.

First, a *local least squares imputation method (LLSImpute)* have been applied (Kim, Golub, & Park, 2005; Stacklies, Redestig, Scholz, Walther, & Selbig, 2007) to deal with missing data.

A gene expression is a real number taht takes continuous values. In the literature, the most used discretization method for BNs and microarray data is the one proposed by (Friedman, Linial, Nachman, & Pe'er, 2000). They chose to discretize into three categories, depending on whether the expression rate is lower than, similar to, or greater than a control value.

Since our objective is not the study of unknown functions of genes involved in cell cycle, but rather studying the use of structural constraints in the analysis of gene expression data, we have decided to use only those genes that appear in the genetic network proposed by (Nariai et al., 2005). Therefore, the variables has been reduced to 54 genes.

5.4 Experimental results

In the experiments carried out, the score+search learning method studied is the standard local search (employing the operators: arc insertion, arc deletion and arc reversal), using the BDeu score metric (Heckerman et al., 1995), with a equivalent sample size set to 1 and an uniform structure prior. The learning algorithm using conditional independence tests utilized in this research is the PC algorithm (Spirtes, Glymour, & Scheines, 2001).

Once the structural learning has finished, the required marginal and conditional probabilities must be estimated from the yeast cell cycle dataset provided by Spellman et al. This applies both to learned networks and true network. For all BNs, we have utilized a smoothed parameter estimator, based on the Laplace law of succession (Good, 1965). The reason is to avoid problems of unreliability and overfitting of the maximum likelihood estimator in small datasets.

The measures used to compare the algorithms are:

- The BDeu score metric of the obtained BNs; this measure is useful because it is the metric used to guide the score+search method.
- Three measures that allow us to reflect the structural differences between the learned networks and the true one. These measures can estimate the ability to reconstruct

the DAG of the true network: the amount of added arcs (A), the amount of deleted arcs (D) and the amount of inverted arcs (I) in the learned network with respect to the true network. To remove inaccurate differences or similarities between the DAGs being compared due to different but equivalent BNs, before comparing two networks, they were transformed into their completed PDAG representation², using the algorithm proposed by (Chickering, 1995).

- The Kullback-Leibler divergence (KL), also called relative entropy, between the probability distributions related to the true BN and the learned BNs is also computed, which represents a measure of the capacity to rebuild the joint probability distribution.

The following levels of random restrictions were selected: 10%, 20%, 30%, and 40% from the full set of restrictions that can be obtained from the completed PDAG representation of the true network. Specifically, if $G = (\mathcal{V}, E_G)$ is the completed PDAG representation of the true network, then each arc $x \rightarrow y \in E_G$ and each edge $x - y \in E_e$ is a feasible existence restriction; each arc $x \rightarrow y \notin E_G$ is a feasible absence restriction and in case that also $y \rightarrow x \notin E_G$ we may utilize the restriction $x \rightarrow y \in E_a$ or $y \rightarrow x \in E_a$ or $x - y \in E_a$; at last, if there is a directed path from x to y in E_G then $x \rightarrow y \in E_o$ is a feasible ordering restriction.

We have studied the learning algorithms for each percentage of restrictions for each type exclusively, and also utilizing the three kinds of restrictions simultaneously. The results displayed show the average values of the benchmarking measures across 50 repetitions. All the algorithms are implemented in the Elvira system (Elvira Consortium, 2002) (available at <http://leo.ugr.es/elvira/>).

Table 1 displays the outcomes achieved using the score+search algorithm, while Table 3 displays the results achieved using the PC algorithm. These tables include the outcomes achieved by the learning algorithms without utilizing structural restrictions (0%), and the KL estimations of the true network, with parameters re-trained from Spellman et al. dataset, which is compared with the learned networks. Table 2 displays the corresponding BDeu average values for the learned networks using local search, as well as the estimation of the BDeu score of the true network.

%	G_e, G_a, G_o				exclusively G_e				exclusively G_a				exclusively G_o			
	KL	A	D	I	KL	A	D	I	KL	A	D	I	KL	A	D	I
10%	12.63	51.9	76.0	4.3	13.12	56.0	77.1	4.8	13.18	54.8	85.4	5.1	13.80	57.7	85.4	4.8
20%	11.71	45.9	67.3	3.2	12.48	52.1	68.6	4.2	12.51	51.2	84.0	5.7	13.66	58.9	85.0	4.6
30%	10.77	40.0	58.8	1.8	12.13	50.8	60.1	3.5	11.75	47.8	82.7	5.6	13.49	58.2	84.5	4.2
40%	9.80	34.9	50.0	1.3	11.74	49.1	51.2	2.9	10.88	43.2	81.4	5.2	13.52	58.7	84.4	3.8
0%	13.66	57.0	87.0	4.0	KL value of the true network: 7.76											

Table 1. Average outcomes achieved using local search.

² As was stated by (de Campos & Castellano, 2007, p. 244), “A completed PDAG (also called essential graph) is a partially directed acyclic graph which is a canonical representation of all the DAGs belonging to the same equivalence class of DAGs.”

	G_e, G_a, G_o	exclusively G_e	exclusively G_a	exclusively G_o
10%	-2509.46	-2474.52	-2440.86	-2432.29
20%	-2596.95	-2530.98	-2469.69	-2442.33
30%	-2689.36	-2587.79	-2504.13	-2448.65
40%	-2783.47	-2647.72	-2542.60	-2452.42
0%	-2418.38	BDeu value of the true network:-3420.68		

Table 2. Average BDeu values obtained using local search.

%	G_e, G_a, G_o				exclusively G_e				exclusively G_a				exclusively G_o			
	KL	A	D	I	KL	A	D	I	KL	A	D	I	KL	A	D	I
10%	8.10	5.4	85.7	0.5	8.13	6.0	86.2	0.8	8.66	5.4	96.5	0.0	8.73	5.9	96.9	0.0
20%	7.51	4.9	75.4	0.7	7.61	6.0	76.4	1.2	8.55	4.9	96.0	0.0	8.72	5.6	96.6	0.0
30%	7.04	5.4	65.1	1.0	7.14	6.0	66.5	1.4	8.41	4.8	92.2	0.0	8.72	5.3	96.3	0.0
40%	6.74	6.8	54.8	0.9	6.77	5.9	56.7	1.4	8.31	5.6	94.2	0.0	8.72	5.2	96.2	0.0
0%	8.74	6.0	97.0	0.0	KL value of the true network: 7.76											

Table 3. Average outcomes achieved using PC.

5.5 Analysis of results

From the outcomes obtained with the experiments carried out, we can say that the utilization of prior knowledge represented by structural restrictions help us to obtain better results regardless the proportion of structural restrictions used. Obviously, the higher the proportion of prior knowledge used, the better the obtained BN will be.

In order to examine these results in more depth, we will comment then according to the following aspects: structural differences, KL divergence and BDeu score.

- **Structural differences:** In this aspect, the local search algorithm works as planned, that is, as the proportion of existence restrictions, absence restrictions and ordering restrictions raises, then proportion of removed, added and inverted arcs, respectively, are reduced. This performance can be undoubtedly noticed in the outcomes. Furthermore, in the experiments, we can observe another less obvious behavior: the utilization of any type of structural restrictions also contributes to reduce other values of structural differences. For instance, absence restrictions reduce the quantity of added arcs, but also the amount of inverted and removed arcs. There is a couple of exceptions to this rule: we don't observe that inverted arcs decreases clearly when we use only absence restrictions nor observe that the added arcs decreases clearly when we use only ordering restrictions. In the case of the PC algorithm (Table 3), the utilization of any of the three kinds of structural restrictions also contribute to reduce the other values of structural differences. However, as exception, the inverted arcs increase slightly with the use of existence restrictions.
- **KL divergence:** According to this measure, there is no doubt about that the use of prior knowledge contributes to more accurate BN structures, and as the proportion of restrictions increases, the improvement increases almost systematically. In the PC algorithm case, the utilization of structural restrictions always reflects more faithfully the true network structure than the PC without structural restrictions.
- **BDeu Score:** Regarding the BDeu score, we only will take into account the algorithm which employs such measure: the local search. In this particular case, we can

note that the obtained BDeu scores are constantly better than the true network score; this may be due a higher level of overfitting to the given data. Furthermore, as we raise the number of restrictions, the BDeu values head to the true network score.

On analyzing the results offered by each type of structural constraints, we can observe that the better results are achieved consistently by existence constraints, that is, adding arcs obtains better results than forbidding arcs or imposing a variable order, regardless of the learning algorithm employed. Similar results are obtained by the absence constraints, but nevertheless, the number of deleted arcs is appreciably higher. The sole use of ordering constraints also obtains improved results but to a lesser degree.

In this data set, we do not observe any saturation effect when we increase the number of structural constraints being used. It is expected that with much more available data, the quality of the estimated networks improves and the value of prior structural knowledge decreases. In this situation of scarce data, is when prior knowledge is more useful in terms of the quality of the final result. In any case, prior knowledge is also important for reducing learning time, as the search space is smaller.

When no prior knowledge is employed, we should observe the poor results obtained by the different learning algorithms. Note that the true network has 100 arcs, while the unrestricted PC algorithm eliminates 97 arcs and the local search algorithm, also without restrictions, eliminates 87. The unrestricted local search algorithm adds 57 arcs which are not in the original network, and the unrestricted PC algorithm adds 6 arcs. The explanation for this poor behaviour of the learning algorithms is the small number of cases (only 77 samples) used for learning the structure. As example, if we focus on the PC algorithm, we observe that it learns only 9 arcs (6 added and 3 in the true network). The reason to learn so few arcs is because, when we are working with a small number of cases, the independence hypothesis states independence for most of the performed tests. The evolution of the Bayesian networks obtained by the PC algorithm as the additional knowledge used was gradually increased, can be found in Figures 1, 2 and 3.

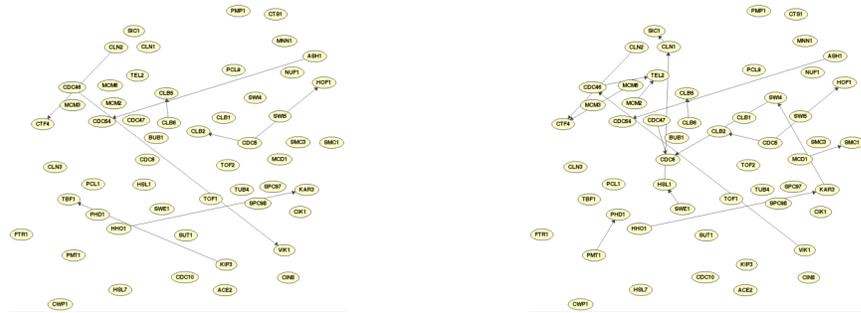


Fig. 1. Bayesian networks obtained by PC algorithm without additional knowledge (left) and using 10% of additional expert knowledge (right).

However, despite this awful starting point without prior knowledge, the use of additional knowledge tends always to better results for all the collected performance measures. Working with so few cases, the use of restrictions will improve the results and the enhancement increases with the proportion of structural restrictions being utilized.

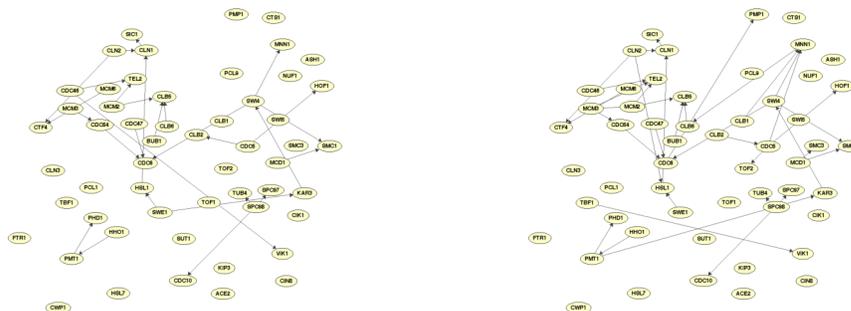


Fig. 2. Bayesian networks obtained by PC algorithm using 20% (left) and 30% of additional expert knowledge (right).

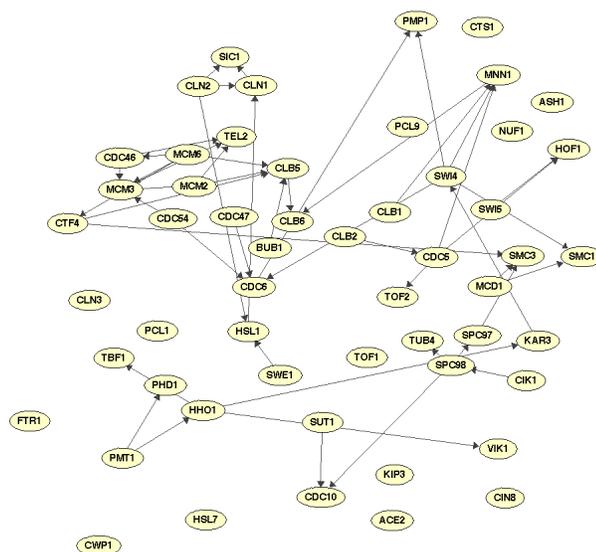


Fig. 3. Bayesian network obtained by PC algorithm using 40% of additional expert knowledge.

We would like to clarify that, unlike classical data, gene expression data have a high dimensionality, that is, a severe number of variables (gene expressions) and a slightly low number of instances. In the context of the PC Learning algorithm, the amount of samples in gene expression data is rarely enough to perform reliable conditional independence tests (M. Wang, Chen, & Cloutier, 2007), which will consequently lead to removing too many edges between variables, resulting Bayesian networks with few arcs. Furthermore, the PC algorithm is prone to obtain sparse networks (Acid et al., 2004). In the case of the search+metric algorithm, the high dimensionality of this data generates correlations between variables which can be simply a chance effect (Hoefsloot, Smit, & Smilde, 2008), causing undesirable links to appear.

6 Conclusions

The prior knowledge is generally employed to enhance the reliability of inferring genetic regulatory networks from Bayesian networks using gene expression data. However, Bayesian networks provide an easily interpretable graphical representation of the relations among the genes which had not been fully exploited for incorporating prior knowledge to reconstruct genetic regulatory networks.

As we have seen in the literature review, most authors employ prior knowledge as a prior probability distribution or as an initial network utilized as starting point of the heuristic search. Therefore we have presented the utilization of three structural restrictions types, i.e. existence, absence and ordering restrictions, to learn the DAG of Bayesian networks and considered their application to gene expression data. We have examined a local search learning algorithm using a score metric and the PC algorithm (a conditional independence paradigm).

The experimental outcomes indicate that the utilization of extra prior knowledge codified as structural restrictions enhances the learned Bayesian network structures even when there are few cases. Consequently the use of structural restrictions can easily implements prior knowledge as they can represent different types of gene regulatory relationships.

Acknowledgments

This work has been supported by the Spanish “Ministerio de Economía y Competitividad” and by “Fondo Europeo de Desarrollo Regional” (FEDER) under Project Projects TEC2015-69496-R and TIN2016-77902-C3-2-P.

References

- Acid, S., & de Campos, L. M. (2003). Searching for Bayesian network structures in the space of restricted acyclic partially directed graphs. *Journal of Artificial Intelligence Research*, 18, 445-490. doi: doi:10.1613/jair.1061
- Acid, S., de Campos, L. M., & Fernández, M. (2013). Score-based methods for learning markov boundaries by searching in constrained spaces. *Data Mining and Knowledge Discovery*, 26(1), 174-212. doi: 10.1007/s10618-011-0247-5
- Acid, S., de Campos, L. M., Fernández-Luna, J. M., Rodríguez, S., Rodríguez, J. M., & Salcedo, J. L. (2004). A comparison of learning algorithms for bayesian networks: a case study based on data from an emergency medical service. *Artificial Intelligence in Medicine*, 30(3), 215 - 232. (Bayesian Networks in Biomedicine and Health-Care) doi: 10.1016/j.artmed.2003.11.002
- Aderhold, A., Husmeier, D., & Grzegorzczak, M. (2014). Statistical inference of regulatory networks for circadian regulation. *Statistical Applications in Genetics and Molecular Biology*, 13(3), 227 - 273. doi: 10.1515/sagmb-2013-005
- Almasri, E., Larsen, P., Chen, G., & Dai, Y. (2008). Incorporating literature knowledge in Bayesian network for inferring gene networks with gene expression data. In I. I. Mandoiu, R. Sunderraman, & A. Zelikovsky (Eds.), *Bioinformatics research and applications, fourth international symposium, isbra 2008. proceedings* (Vol. 4983, p. 184-195). doi: 10.1007/978-3-540-79450-9_18

- Banf, M., & Rhee, S. Y. (2017). Computational inference of gene regulatory networks: Approaches, limitations and opportunities. *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms*, 1860(1), 41 - 52. doi: 10.1016/j.bbagr.2016.09.003
- Bansal, M., Belcastro, V., Ambesi-Impombato, A., & di Bernardo, D. (2007). How to infer gene networks from expression profiles. *Molecular Systems Biology*, 3. doi: 10.1038/msb4100120
- Bellman, R. E. (1957). *Dynamic programming*. Princeton University Press.
- Buntine, W. (1991). Theory refinement on Bayesian networks. In *Uncertainty proceedings 1991* (p. 52 - 60). San Francisco (CA): Morgan Kaufmann. doi: 10.1016/B978-1-55860-203-8.50010-3
- Buntine, W. (1996). A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, 8(2), 195-210. doi: 10.1109/69.494161
- Chai, L. E., Loh, S. K., Low, S. T., Mohamad, M. S., Deris, S., & Zakaria, Z. (2014). A review on the computational approaches for gene regulatory network construction. *Computers in Biology and Medicine*, 48, 55 - 65. doi: 10.1016/j.compbiomed.2014.02.011
- Chen, G., Cairelli, M. J., Kilicoglu, H., Shin, D., & Rindflesch, T. C. (2014). Augmenting microarray data with literature-based knowledge to enhance gene regulatory network inference. *PLoS computational biology*, 10(6), e1003666. doi: 10.1371/journal.pcbi.1003666
- Cheng, J., Greiner, R., Kelly, J., Bell, D., & Liu, W. (2002). Learning Bayesian networks from data: An information-theory based approach. *Artificial Intelligence*, 137(1), 43 - 90. doi: 10.1016/S0004-3702(02)00191-1
- Chickering, D. M. (1995). A transformational characterization of equivalent Bayesian network structures. In P. Besnard & S. Hanks (Eds.), *Uai '95: Proceedings of the eleventh annual conference on uncertainty in artificial intelligence* (p. 87-98). Morgan Kaufmann.
- Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., ... Davis, R. W. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2(1), 65-73. doi: 10.1016/S1097-2765(00)80114-8
- Chow, C., & Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3), 462-467. doi: 10.1109/TIT.1968.1054142
- Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4), 309-347. doi: 10.1007/BF00994110
- de Campos, L. M., & Castellano, J. G. (2007). Bayesian network learning algorithms using structural restrictions. *International Journal of Approximate Reasoning*, 45(2), 233 - 254. doi: 10.1016/j.ijar.2006.06.009
- de Campos, L. M., & Huete, J. F. (2000). A new approach for learning belief networks using independence criteria. *International Journal of Approximate Reasoning*, 24(1), 11 - 37. doi: 10.1016/S0888-613X(99)00042-0
- Djebbari, A., & Quackenbush, J. (2008). Seeded Bayesian networks: constructing genetic networks from microarray data. *BMC Systems Biology*, 2, 57. doi: 10.1186/1752-0509-2-57
- Elvira Consortium. (2002). Elvira: An environment for probabilistic graphical mod-

- els. In J. Gámez & A. Salmerón (Eds.), *Procs. of the 1st european workshop on probabilistic graphical models (pgm 2002)* (p. 222-230).
- Esteves, G. H., & Reis, L. F. L. (2018). A statistical method for measuring activation of gene regulatory networks. *Statistical Applications in Genetics and Molecular Biology*, *17*(3). doi: 10.1515/sagmb-2016-0059
- Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, *303*(5659), 799-805. doi: 10.1126/science.1094068
- Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, *7*(3-4), 601-620. doi: 10.1089/106652700750050961
- Gámez, J. A., Mateo, J. L., & Puerta, J. M. (2011). Learning Bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighborhood. *Data Mining and Knowledge Discovery*, *22*(1), 106-148. doi: 10.1007/s10618-010-0178-6
- Gifford, D. K. (2001). Blazing pathways through genetic mountains. *Science*, *293*, 2049-2051. doi: 10.1126/science.1065113
- Good, I. J. (1965). *The estimation of probabilities*. The MIT Press.
- Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., & Young, R. A. (2001). Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pacific Symposium on Biocomputing*, 422-33.
- Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., & Young, R. A. (2002a). Bayesian methods for elucidating genetic regulatory networks. *IEEE Intelligent Systems*, *17*(2), 37-43. doi: 10.1109/5254.999218
- Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., & Young, R. A. (2002b). Combining location and expression data for principled discovery of genetic regulatory network models. *Pacific Symposium on Biocomputing*, 437-49.
- Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, *20*(3), 197-243. doi: 10.1023/A:1022623210503
- Hoefsloot, H. C., Smit, S., & Smilde, A. K. (2008). A classification model for the leiden proteomics competition. *Statistical applications in genetics and molecular biology*, *7*(2). doi: 10.2202/1544-6115.1351
- Hughes, T. R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton, R., Armour, C. D., ... Friend, S. H. (2000). Functional discovery via a compendium of expression profiles. *Cell*, *102*(1), 109-126. doi: 0.1016/S0092-8674(00)00015-5
- Imoto, S., Higuchi, T., Goto, T., Tashiro, K., Kuhara, S., & Miyano, S. (2003). Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. In *2nd ieee computer society bioinformatics conference (csb 2003)* (p. 104-113). IEEE Computer Society.
- Imoto, S., Higuchi, T., Goto, T., Tashiro, K., Kuhara, S., & Miyano, S. (2004). Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. *Journal of Bioinformatics and Computational Biology*, *2*(1), 77-98. doi: 10.1142/S021972000400048X
- Imoto, S., Kim, S., Goto, T., Miyano, S., Aburatani, S., Tashiro, K., & Kuhara, S. (2003). Bayesian network and nonparametric heteroscedastic regression for non-linear modeling of genetic network. *Journal of Bioinformatics and Computational Biology*, *1*(2), 231-52.
- Isci, S., Dogan, H., Ozturk, C., & Otu, H. H. (2014). Bayesian network prior: network

- analysis of biological data using external knowledge. *Bioinformatics*, *30*(6), 860-867. doi: 10.1093/bioinformatics/btt643
- Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., . . . Yamanishi, Y. (2008). KEGG for linking genomes to life and the environment. *Nucleic Acids Research*, *36*, 480-484. doi: 10.1093/nar/gkm882
- Kim, H., Golub, G. H., & Park, H. (2005). Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics*, *21*(2), 187-198. doi: 10.1093/bioinformatics/bth499
- Lam, W., & Bacchus, F. (1994). Learning Bayesian belief networks: An approach based on the mdl principle. *Computational Intelligence*, *10*(3), 269-293. doi: 10.1111/j.1467-8640.1994.tb00166.x
- Larsen, P., Almasri, E., Chen, G., & Dai, Y. (2007). A statistical method to incorporate biological knowledge for generating testable novel gene regulatory interactions from microarray experiments. *BMC Bioinformatics*, *8*, 317. doi: 10.1186/1471-2105-8-317
- Lee, W. P., & Tzou, W. S. (2009). Computational methods for discovering gene networks from expression data. *Briefings in Bioinformatics*, *10*(4), 408-423. doi: 10.1093/bib/bbp028
- Le Phillip, P., Bahl, A., & Ungar, L. H. (2004). Using prior knowledge to improve genetic network reconstruction from microarray data. *In Silico Biology*, *4*(3), 335-353.
- Li, S., Wu, L., & Zhang, Z. (2006). Constructing biological networks through combined literature mining and microarray analysis: a lmma approach. *Bioinformatics*, *22*(17), 2143-2150. doi: 10.1093/bioinformatics/btl363
- Linde, J., Schulze, S., Henkel, S. G., & Guthke, R. (2015). Data- and knowledge-based modeling of gene regulatory networks: an update. *Experimental and Clinical Sciences*, *14*, 346-378. doi: 10.17179/excli2015-168
- Markowitz, F., & Spang, R. (2007). Inferring cellular networks - a review. *BMC Bioinformatics*, *8*(6). doi: 10.1186/1471-2105-8-S6-S5
- Mewes, H. W., Frishman, D., Güldener, U., Mannhaupt, G., Mayer, K. F. X., Mokrejs, M., . . . Weil, B. (2002). MIPS: a database for genomes and protein sequences. *Nucleic Acids Research*, *30*(1), 31-34. doi: 10.1093/nar/30.1.31
- Mukherjee, S., & Speed, T. P. (2008). Network inference using informative priors. *Proceedings of the National Academy of Sciences (PNAS)*, *105*(38), 14313-14318. doi: 10.1073/pnas.0802272105
- Nariai, N., Kim, S., Imoto, S., & Miyano, S. (2004). Using protein-protein interactions for refining gene networks estimated from microarray data by Bayesian networks. *Pacific Symposium on Biocomputing*, 336-47.
- Nariai, N., Tamada, Y., Imoto, S., & Miyano, S. (2005). Estimating gene regulatory networks and protein-protein interactions of *Saccharomyces cerevisiae* from multiple genome-wide data. *Bioinformatics*, *21*(2), 206-212. doi: 10.1093/bioinformatics/bti1133
- Njah, H., & Jamoussi, S. (2015). Weighted ensemble learning of Bayesian network for gene regulatory networks. *Neurocomputing*, *150*, 404-416. doi: 10.1016/j.neucom.2014.05.078
- Oates, C. J., Amos, R., & Spencer, S. E. (2014). Quantifying the multi-scale performance of network inference algorithms. *Statistical Applications in Genetics and Molecular Biology*, *13*(5), 611 - 631. doi: 10.1515/sagmb-2014-0012

- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann Publishers Inc.
- Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., & Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, *308*(5721), 523–529. doi: 10.1126/science.1105809
- Sauta, E., Demartini, A., Vitali, F., Riva, A., & Bellazzi, R. (2017). Data fusion approach for learning transcriptional Bayesian networks. In A. ten Teije, C. Popow, J. H. Holmes, & L. Sacchi (Eds.), *Artificial intelligence in medicine: 16th conference on artificial intelligence in medicine, aime 2017* (pp. 76–80). Springer International Publishing. doi: 10.1007/978-3-319-59758-4_8
- Schlitt, T., & Brazma, A. (2007). Current approaches to gene regulatory network modelling. *BMC Bioinformatics*, *8*. doi: 10.1186/1471-2105-8-S6-S9
- Segal, E., Wang, H., & Koller, D. (2003). Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics*, *19*(1), i264–i272. doi: 10.1093/bioinformatics/btg1037
- Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., . . . Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, *9*, 3273–3297. doi: 10.1091/mbc.9.12.3273
- Spirtes, P., Glymour, C., & Scheines, R. (1993). *Causation, prediction, and search* (Vol. 81). MIT press. doi: 10.1007/978-1-4612-2748-9
- Spirtes, P., Glymour, C., & Scheines, R. (2001). *Causation, prediction, and search, 2nd edition* (2nd ed., Vol. 1). The MIT Press.
- Spirtes, P., Glymour, C., Scheines, R., Kauffman, S., Aimale, V., & Wimberly, F. (2000). Constructing Bayesian network models of gene expression networks from microarray data. In *Proceedings of the atlantic symposium on computational biology, genome information systems and technology*.
- Stacklies, W., Redestig, H., Scholz, M., Walther, D., & Selbig, J. (2007). pcaMethods—a bioconductor package providing PCA methods for incomplete data. *Bioinformatics*, *23*(9), 1164–1167. doi: 10.1093/bioinformatics/btm069
- Steele, E., Tucker, A., Hoen, P. t., & Schuemie, M. (2009). Literature-based priors for gene regulatory networks. *Bioinformatics*, *25*(14), 1768–1774. doi: 10.1093/bioinformatics/btp277
- Styczynski, M. P., & Stephanopoulos, G. (2005). Overview of computational methods for the inference of gene regulatory networks. *Computers & Chemical Engineering*, *29*(3), 519–534. doi: 10.1016/j.compchemeng.2004.08.029
- Tamada, Y., Kim, S., Bannai, H., Imoto, S., Tashiro, K., Kuhara, S., & Miyano, S. (2003a). Combining gene expression data with DNA sequence information for estimating gene networks using Bayesian network model. In *Genome informatics* (Vol. 14, p. 352–353). doi: 10.11234/gi1990.14.352
- Tamada, Y., Kim, S., Bannai, H., Imoto, S., Tashiro, K., Kuhara, S., & Miyano, S. (2003b). Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. *Bioinformatics*, *19*(2), 227–236.
- Wang, M., Chen, Z., & Cloutier, S. (2007). A hybrid Bayesian network learning method for constructing gene networks. *Computational Biology and Chemistry*, *31*(5–6), 361–372.
- Wang, Y. R., & Huang, H. (2014). Review on statistical methods for gene network

- reconstruction using expression data. *Journal of Theoretical Biology*, 362, 53 - 61. doi: 10.1016/j.jtbi.2014.03.040
- Werhli, A. V., Grzegorzczak, M., & Husmeier, D. (2006). Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical Gaussian models and Bayesian networks. *Bioinformatics*. doi: 10.1093/bioinformatics/btl391
- Werhli, A. V., & Husmeier, D. (2007a). Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Statistical Applications in Genetics and Molecular Biology*, 6(1), 15. doi: 10.2202/1544-6115.1282
- Werhli, A. V., & Husmeier, D. (2007b). Reverse engineering gene regulatory networks with Bayesian networks from expression data combined with multiple sources of biological prior knowledge. In *11th annual international conference on research in computational molecular biology (recomb 2007)* (Vol. 49).
- Wit, E., & McClure, J. (2004). *Statistics for microarrays*. John Wiley & Sons.
- Yu, J., Smith, V. A., Wang, P. P., Hartemink, A. J., & Jarvis, E. D. (2004). Advances to Bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, 20(18), 3594-3603. doi: 10.1093/bioinformatics/bth448
- Zhou, H., & Zheng, T. (2014). Bayesian hierarchical graph-structured model for pathway analysis using gene expression data. *Statistical Applications in Genetics and Molecular Biology*, 12(3), 393 - 412. doi: 10.1515/sagmb-2013-0011
- Zhu, J., Zhang, B., Smith, E. N., Drees, B., Brem, R. B., Kruglyak, L., . . . Schadt, E. E. (2008). Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. *Nature Genetics*, 40, 854 - 861. doi: 10.1038/ng.167