2014 IEEE 14th International Conference on Bioinformatics and Bioengineering

Gene Networks Inference through Linear Grouping of Variables

Carlos Fernando Montoya-Cubas, David Correa Martins-Jr, Carlos Silva Santos
Center of Mathematics, Computation and Cognition
Federal University of ABC
Santo André-SP, Brazil
Email: {fernando.montoya,david.martins,carlos.ssantos}@ufabc.edu.br

Junior Barrera
Institute of Mathematics and Statistics
University of São Paulo
São Paulo-SP, Brazil
Email: jb@ime.usp.br

***Abstract*—The inference of gene networks from gene expression data is an open problem due to the large dimensionality (number of genes) and the small number of data samples typically available, even considering the fact that the network is sparse (limited number of input genes per target gene). In this work we propose a method that alleviates the curse of dimensionality by grouping predictor gene configurations in their respective linear combination values. Each linear combination value results in an equivalence class. In this way, the number of configurations of predictor values becomes a linear function of the dimensionality (number of predictors) instead of an exponential function when considering the original configurations. The proposed method follows the probabilistic gene networks approach which applies local feature selection to obtain an adequate predictor gene set for each gene. Even considering that some information from the original configurations of predictors is lost after applying the grouping, the results indicate that the inference with linear grouping tends to provide networks with better topological similarities than those obtained without grouping in cases where the number of samples is quite limited and the inference involves a larger number of predictors per gene.**

I. Introduction

In recent years, the development of techniques such as DNA Microarrays [33], SAGE [38] and RNA-Seq [39], has made it possible to estimate the expression level (mRNA concentrations) of thousands of genes simultaneously and in several timepoints. With the availability of this data, several methods for analyzing the dynamical evolution of the gene expression levels have been proposed, with the goal of reverse engineering the regulatory control mechanisms [16].

Gene expression data poses a hard challenge for statistical analysis due to its high-dimensionality: since the typical scenario involves a few dozens of experiments and thousands of genes, the number of parameters to be estimated is several orders of magnitude higher than the available samples. In addition, gene expression data is inherently noisy and prone to the scarcity of prior information about many biological organisms [3]. Classical statistical methods are not able to cope with this restriction, thus motivating the appearance of methods that use additional assumptions, such as smoothness and sparsity, and/or try to incorporate prior knowledge and integration of other types of data sources [16], [25], [27]. Despite the availability of high volume data and the increasing interest garnered by the subject, the inference of gene regulatory networks (GRN) remains an open problem [16], [35], [21], [28], [31].

In fact, the inference of networks from gene expression data is an inverse and ill-posed problem, in the sense that many solutions may be capable to explain the observed data. This makes the problem quite complex, since the number of samples is limited and the data is subject to experimental noises as aforementioned. The inference process requires a good modeling framework combined with very well designed search or learning procedures.

There are essentially two main approaches to model the complex networks of gene interactions [34]: continuous and discrete. The continuous approach relies on differential equations to reach a quantitative detailed model of biochemical networks with cellular functions [18]. On the other hand, the discrete approach is based on the construction of qualitative discrete models of gene interactions, including the models based on graphs such as the Bayesian Networks [15], Boolean Networks [20] and the Probabilistic Boolean Networks [35] which includes the Probabilistic Gene Networks [5]. Although the continuous approaches offer a detailed understanding of the considered system, they require a significant number of samples and information about the characteristics of the reactions [19]. In its turn, the discrete approaches can be easily modeled computationally and have been successfully employed in the modeling and simulation of some biological process networks, such as *Drosophila melanogaster* [32], [2], yeast cell cycle [23], [40], [7], *Arabidopsis thaliana* [13], *Saccharomyces cerevisae* [23], mammal cell cycle [14], *Plasmodium falciparum* [5], among others.

In the context of discrete models, Boolean Networks represent an appropriate model to generalize and capture the global behavior of biological systems, specially when the number of experiments (samples) available is limited and the dimensionality (number of variables) is very large [20]. The main disadvantage of such model is the information loss as a consequence of the data quantization. However, the data quantization is exactly what makes the Boolean model simpler [36], [17]. Many methods were proposed to infer gene networks modeled as Boolean Networks [1], [22], [24], [30].

Although genes can have only two possible values in Boolean networks, it still difficult to infer their topologies based on a small number of samples. Even considering the probabilistic gene networks approach [5] whose assumptions provide a simplification of the inference process that allows to apply local feature selection (i.e., the best predictor gene subset is searched for each target gene), the curse of dimensionality is

978-1-4799-7502-0/14 $31.00 © 2014 IEEE
DOI 10.1109/BIBE.2014.10

still a major problem, since the number of statistical parameters (configuration values) to be estimated in a subset of cardinality k is 2^k for Boolean networks. For instance, if a gene has 5 predictors, the number of possible configuration values for the predictors is 32, which requires a data set with much more than 32 samples (timepoints) to reliably estimate the conditional probabilities of the target gene to be 0 or 1 given the configurations of the predictors (otherwise, many configurations would be non observed or poorly observed).

In order to amenize the curse of dimensionality, in this work we proposed a linear grouping method that maps the predictor configuration values into their corresponding linear combination values. As a consequence of this mapping, the number of configurations becomes a linear function of the cardinality of the predictor set, improving the statistical estimation at the expense of an information loss about the configuration values which occurs after such mapping. Results obtained from artificial data show that the linear grouping is sound in cases where the inference process involves large number of predictors per gene and small number of samples.

This manuscript is organized as follows. Section II defines the Boolean networks model and its stochastic version, the probabilistic Boolean networks model. Section III discusses the probabilistic gene networks (PGN) approach. Section IV introduces the proposed technique of linear grouping to reduce the number of possible predictor configuration values. Section VI shows preliminary results comparing the network similarities achieved by the PGN inference approach with linear grouping against the similarities obtained by the same approach without linear grouping. Some final remarks are provided in Section VII.

II. Boolean Networks

A Boolean Network (BN) is a model proposed for the study of complex systems dynamics, and of GRNs in particular [20]. In a Boolean Network, a set of n Boolean variables is represented by a set $V = \{v_1, v_2, \ldots, v_n\}$ of vertices in a graph; the other component of a BN is a set of Boolean functions $\Phi = \{\phi_1, \phi_2, \ldots, \phi_n\}$, each function corresponding to one vertex, called Boolean transition functions [8]. In GRNs modeling, each vertex v_i is associated with a certain gene, thus we will refer to v_i as either a gene or a vertex, indistinctly. Each gene $v_i \in \{0, 1\}, i = 1, 2, ..., n$ represents a binary variable for which its value in the next time instant $t + 1$ is completely determined by the values of its k_i predictor genes in the current time instant t. More concretely, this dynamic can be represented by $v_i(t + 1) = f_i(v_{1i}(t), v_{2i}(t), ..., v_{ki}(t))$, in which $v_{1i}, v_{2i}, ..., v_{ki}$ represents the k_i predictor (or regulatory) genes that influence the target gene v_i.

The BN model considers only discrete iterations in time, and all genes are synchronously updated at the same instant, according to some deterministic function ϕ_i. In this model, the dynamic of the system is deterministic, i.e., the k_i predictors and the respective logical function for each gene v_i remain the same during all timepoints.

A. Probabilistic Boolean Networks

Although the BNs are useful in many situations, an important limitation is its determinism, which assumes an enviroment without uncertainty. Moreover, it is important to consider a cell as an open system, which can receive external stimuli. In order to model the stochasticity of biological systems, the Probabilistic Boolean Networks model (PBN) has been proposed [35].

A PBN is a network in which each gene in a certain timepoint has its binary expression value determined by a set of Boolean functions of other genes in the previous timepoint, where each function has a probability of being applied [35]. Thus, a BN is a specific type of PBN where all genes are determined by a unique Boolean predictor function with probability equal to 1.

Normally, the quasi-determinism inherent in biological systems can be modeled by PBN simply assigning, for each gene, a probability close to 1 to a certain function and probabilities close to 0 to the remaining functions. The functions with very small probabilities can simulate perturbations (external stimuli) or changes between biological contexts [6], [11].

III. Probabilistic gene networks inference approach

The probabilistic gene networks approach relies on the application of the feature selection principle: for each target, a search for the subset of predictors that best describes the target behavior according to their expression signals is performed [5], [26], [25]. Barrera *et al* employed this approach to analyse the temporal expression signals of the Plasmodium *falciparum* (one of the agents of the malaria disease), displaying notable biological results [5]. Such approach assumes that the temporal samples follow a first order Markov chain in which each target gene value in a given instant of time depends only on its predictor values at the previous instant of time. The transition function is homogeneous (it is the same for every time step), almost deterministic (from any given state, there is one preferential state to go in the next time) and conditionally independent. These assumptions are important simplifications in small number of samples settings. An optinal assumption for further simplification is to consider that the genes are linearly dependent on other genes (linear dependence). We consider this assumption in our proposed method described in the following section.

IV. Linear Grouping of Variables for Dimensionality Reduction

Estimating an k-variable Boolean function from experimental data requires 2^k parameters: for each of the 2^k possible configurations of the input variables, it is necessary to say whether the output should be 0 or 1. Even for moderately large values of k, the number of parameters tend to be much higher than the number of samples. In this situation, most of the input configurations are never observed on experimental data and, even for an observed configuration, the number of occurrences might not be enough for high confidence estimation of the output value. This is one instance of the phenomenon called *curse of dimensionality*, in which the number of samples is insufficient for reliable estimation of the conditional probabilities [37].

In order to reduce the number of necessary parameters for the estimation of the conditional probability distributions $P(Y|\mathbf{Z})$, where Y is a binary variable and $\mathbf{Z}$ is a binary vector

in $\{0,1\}^k$, we propose a new method called "linear grouping". This goal is achieved by reducing the number of states from 2^k to $k+1$ with a linear mapping of input vector $\mathbf{Z} \in \{0,1\}^k$ into a integer number $L \in \mathbb{Z}$. The mapping is defined by a linear combination, expressed by Equation 1:

$$L = a_1 Z_1 + a_2 Z_2 + \ldots + a_k Z_k \quad (1)$$

in which $a_i \in \{-1, 1\}$ for $i \in \{1, 2, \ldots, k\}$. We define a coefficient vector $\mathbf{A} = \{a_1, a_2, ..., a_k\} \in \{-1, +1\}^k$ rewriting in vectorial format $L = \mathbf{A}^T\mathbf{Z}$. This modeling assumes that a predictor can be an activator (coefficient equal to $+1$) or an inhibitor (coefficient equal to -1) of the given target gene Y. As each coefficient a_i can assume two values, there exists 2^k possible linear combinations to be evaluated for a predictor set with dimension k. For a given predictor set $\mathbf{Z}$, the configuration of coefficients $\mathbf{A}^*$ which results in the best criterion function value is adopted as the linear combination that predicts Y based on $\mathbf{Z}$. Thus, the ranking of different predictor sets for a given target is based on the criterion function values corresponding to their best linear combinations.

As the predictors Z_i are discrete random variables (binary) and the coefficients a_i are integer numbers, L is a discrete random variable as well. The number of possible values for L is exactly $k+1$ for any coefficient vector configuration $\mathbf{A}$. The demonstration of this fact is as follows. Let k^- and k^+ be the number of negative coefficients (inhibitors) and the number of positive coefficients (activators) respectively. Then, the dimension of the predictor set is given by $k = k^- + k^+$. Since the gene expressions are binary values (0 or 1), the smallest value that L can assume is exactly $-k^-$, which happens when all inhibitor genes are on (1) and all activator genes are off (0). Analogously, the largest value that L can assume is k^+, occurring when all inhibitor genes are off (0) and all activator genes are on (1). Therefore $L \in \{-k^-, -k^- + 1, ..., k^+ - 1, k^+\}$ with cardinality given by $k^+ - (-k^-) + 1 = k + 1$, as we would like to prove. This result implies that the 2^k instances of the original predictor vector are mapped to $k+1$ equivalence classes, according to the resulting values of a given linear combination of the predictor values. In the proposed method, the direct estimation of $P(Y|\mathbf{Z})$ is replaced by the estimation of $P(Y|L)$. In this way, the number of parameters to be estimated from a samples set becomes significantly reduced.

Although there are 2^k distinct forms to define the coefficient vector $\mathbf{A}$ as aforementioned, only half of these forms ($2^{k+!}$) need to be evaluates, since the other half generates exactly the same groups with the signal of L inverted. In fact, if $\mathbf{A}' = -\mathbf{A}$, then $\mathbf{A}'$ and $\mathbf{A}$ generate the same groups where the magnitudes of their respective linear combinations are the same, but with the inverted signals. Table I shows an example which illustrates this fact.

For the example of Table I with dimension $k = 3$, there exists $2^3 = 8$ possible linear combinations, but only 4 of them needs to be investigated: $(-1,-1,-1); (-1,-1,+1); (-1,+1,-1); (-1,+1,+1)$. The other 4 remaining combinations generate just the same partitions. Figure 1 presents the partitioning of the Boolean lattice which is done for each of these linear combinations.

TABLE I. GROUPING OF INSTANCES IN PARTITIONS WITH $(a_1, a_2, a_3) = (-1, -1, -1)$ AND $(a_1, a_2, a_3) = (+1, +1, +1)$. NOTE THAT BOTH GROUPINGS GENERATE THE SAME PARTITIONS.

$(a_1, a_2, a_3) = (-1, -1, -1)$		$(a_1, a_2, a_3) = (+1, +1, +1)$	
$L_1 =$	$-1z_1 - 1z_2 - 1z_3$	$L_2 =$	$+1z_1 + 1z_2 + 1z_3$
-3	111	3	111
-2	011,101,110	2	011,101,110
-1	001,010,100	1	001,010,100
0	000	0	000

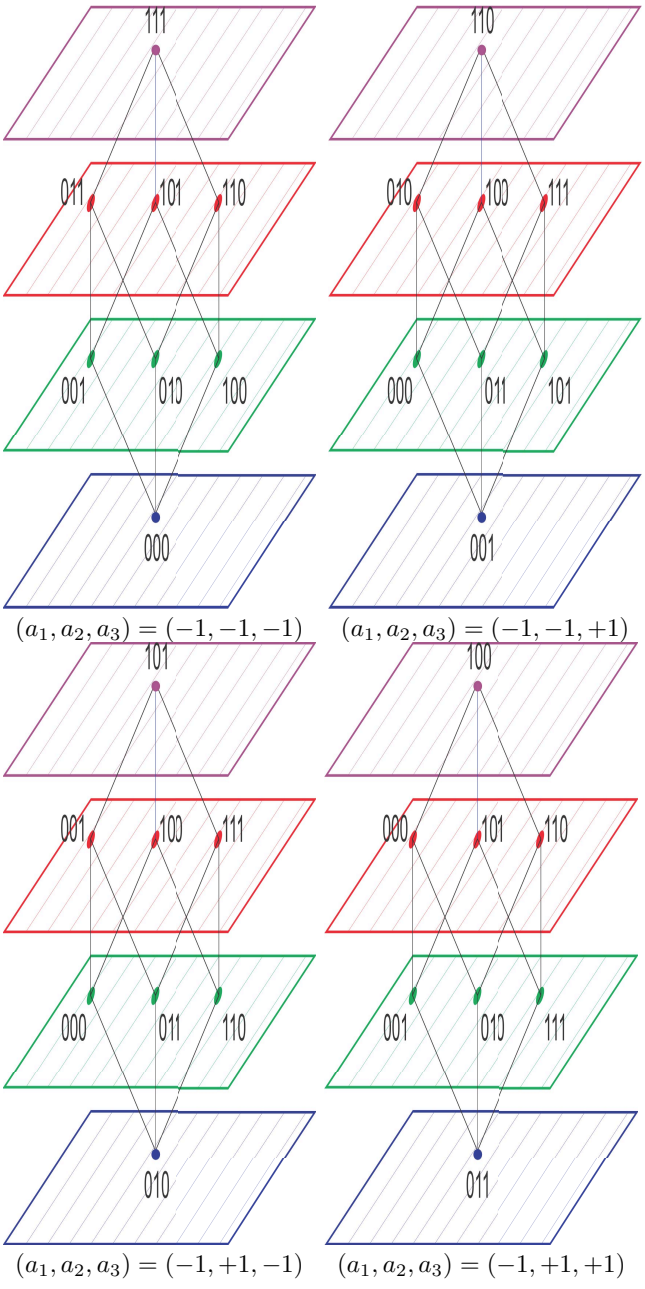

Fig. 1. Boolean lattice partitioning for the linear coefficients $(a_1, a_2, a_3) = \{(-1,-1,-1); (-1,-1,+1); (-1,+1,-1); (-1,+1,+1)\}$.

V. EXPERIMENTS

To evaluate the method proposed to infer gene networks from expression data, we conducted experiments with simu-

lated data. Artificial Boolean networks were randomly generated and the dynamic signals (expression profiles) produced by the networks were simulated along timepoints to create the input dataset. Section V-A describes the procedure to generate the artificial networks and the corresponding simulated gene expression data. The adopted topological similarity metrics for evaluation of the results are provided in Section V-B. The adopted feature search algorithm and the criterion function are presented in Sections V-C and V-D, respectively. The parametrization values adopted for the execution of the experiments are found in Section V-E. Finally, the experimental results involving the comparison of the feature search method (described in Sections V-C and V-D) without linear grouping and the same method with linear grouping are presented in discussed in Section VI.

A. Generation of probabilistic Boolean networks and simulated data

To generate a probabilistic Boolean network (PBN), we fixed the number of genes (n), the average number of predictors per gene ($\langle k \rangle$), and the network topology, chosen between the random networks model Erdös-Rényi (ER) [12] and the scale-free networks model Barabási-Albert (BA) [4]. First, the network topology is generated employing one of the two models aforementioned. Next, for each gene g_i, a set of Boolean functions ϕ_i is randomly chosen from the set of $2^{2^{k_i}}$ possible functions of k_i predictors, where k_i is the number of predictors of the gene $g+i$. The Quine-McCluskey [29] method was applied to verify wether a certain selected Boolean function is minimum (i.e, wether it really depends of all predictors assigned to it in the groundtruth network or not). If the function does not depends on all variables, new functions are randomly chosen until finding one which depends on all variables. At the end, each gene has a fixed number of predictor functions, in which each function has a probability to be applied. Considering Boolean networks, each gene has a unique predictor function (deterministic behavior). On the other hand, for probabilistic Boolean networks, 2 predictor functions per gene were fixed in the experiments, where one of them presents probability close to 1, simulating the quasi-deterministic behavior inherent to real biological systems.

Once we have the network topology and the logical dependencies, the expression profile data can be generated by fixing the number m of timepoints (number of samples) and, for each PBN, an initial state $\vec{s}_0$ is randomly chosen from the set of all 2^n possible states. Then, the evolution of the network states is simulated from $\vec{s}_0$ to $\vec{s}_{m-1}$ through repeated application of the functions set Φ. In case in which any state $\vec{s}_i$ be repeated in the simulation (this can happen if the simulation passes through all atractor states before the generation of m states), the dataset generated so far is discarded and the simulation resets to another state $\vec{s}_0$ randomly chosen. In this way, we guarantee that each simulation passes through m distinct states.

B. Evaluation metrics adopted

The inferred network were compared to groundtruth networks by using two topological similarity metrics based on the number of true / false positives and true / false negatives: Positive Predictive Value (PPV) and Similarity (SIM) [9]. A true-positive is an edge present in both groundtruth and inferred networks. A true-negative is an edge not present in both networks. A false-positive is an edge present only in the inferred network, while a false-negative is an edge present only in the groundtruth network. Letting TP, TN, FP e FN be the numbers of true-positives, true-negatives, false-positives and false-negatives, respectively, PPV is defined by:

$$PPV = \frac{TP}{TP + FP} \tag{2}$$

and SIM is defined by:

$$\sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \tag{3}$$

C. Feature search algorithm adopted

The feature search algorithm adopted first applies an exhaustive search for the best subset of degree k', where k' is a predefined parameter. Then, the Sequential Backward Search (SBS) is applied taking the result of the exhaustive search step as initial subset. SBS stops when the best subset of size $k-1$ be worse than the subset of size k. Algorithm 1 contains a more precise description of this procedure.

Algorithm 1 High level description of the feature search algorithm adopted.

Require: Set of all genes $\mathbf{X}$, target gene Y, gene expression dataset, criterion function $\mathcal{F}$ to be minimized, and initial degree k'

Ensure: Predictor set $\mathbf{Z} = \{z_1, \ldots, z_k\}$ for Y

1: $\mathbf{Z} \leftarrow$ best subset of size k' of $\mathbf{X}$ as predictor of Y by applying exhaustive search
2: $\mathbf{Z}' \leftarrow \mathbf{Z} - \{Z^*\}$ such that $Z^* = argmin_{Z \in \mathbf{Z}} \mathcal{F}(\mathbf{Z} - \{Z\}, Y)$
3: **while** $\mathcal{F}(\mathbf{Z}', Y) < \mathcal{F}(\mathbf{Z}, Y)$ and $|\mathbf{Z}| > 0$ **do**
4: $\quad \mathbf{Z} \leftarrow \mathbf{Z}'$
5: $\quad \mathbf{Z}' \leftarrow \mathbf{Z} - \{Z^*\}$ such that $Z^* = argmin_{Z \in \mathbf{Z}} \mathcal{F}(\mathbf{Z} - \{Z\}, Y)$
6: **end while**
7: **return** $\mathbf{Z}$

D. Criterion function adopted

Here we choose as criterion function the penalized mean conditional entropy [26] defined by:

$$H(Y|\mathbf{Z}) = \frac{\alpha(|\mathbf{Z}| - N)H(Y) + \sum_{i=1}^{N}(f_i + \alpha)H(Y|\mathbf{Z} = \mathbf{z_i})}{\alpha|\mathbf{Z}| + s} \tag{4}$$

where $|\mathbf{Z}|$ is the number of possible instances of the predictor vector $\mathbf{Z}$ (cardinality of $\mathbf{Z}$), N is the number of observed configurations (so, the number of non-observed configurations is given by $|\mathbf{Z}| - N$), f_i is the absolute frequency (number of observations) of $\mathbf{z_i} \in \mathbf{Z}$ and s is the number of samples. α is a positive penalization parameter. $H(Y|\mathbf{Z} = \mathbf{Z_i})$ is the conditional entropy of Y given $\mathbf{Z} = \mathbf{z_i}$ defined by $\sum_{y \in Y} P(y|\mathbf{z_i}) log P(y|\mathbf{z_i})$. If the linear grouping proposed

here described in Secion IV is applied, then the set of linear combination values L replaces $\mathbf{Z}$ and $l_i \in L$ replaces $\mathbf{z_i}$.

The penalized mean conditional entropy has been successfully applied as criterion function for feature selection in the context of gene networks inference [5], [26], [25], [27]. The entropy measures the disorder degree of a variable, so it should be minimized.

E. Parametrization adopted

Table II presents the parameter values used to perform the experiments.

TABLE II. PARAMETERS USED TO PERFORM THE EXPERIMENTS.

Parameter	Value
Network size (number of genes n)	20
Average degree of the groundtruth network $\langle k_{gt} \rangle$	3
Number of timepoints (signal length m)	{20,50}
Topological models	{ER,BA}
Functional models	{BN,PBN}
Probabilities of the PBN Boolean functions	$(0, 96; 0, 04)$
Initial degree to start the search (k')	5
Mean conditional entropy penalization (α)	1

According to the parameters described in Table II, eight experiments were performed, varying between two topological models (ER and BA), two functional models (BN and PBN), two signal lengths (20 and 50).

VI. RESULTS AND DISCUSSION

For each one of the eight experiments described in the previous section, we generated 30 different groundtruth networks, each one generating 30 distinct sample sets, resulting in a total of 900 gene expression datasets. We took into account these datasets to perform a comparative analysis of the feature selection method described in Algorithm 1 guided by the penalized mean conditional entropy with linear grouping (LG) versus without linear grouping (normal) to infer gene networks. Figures 2 and 3 present boxplots of the PPV and SIM values, respectively (each boxplot represents 900 values). Table III summarizes these results (averages and standard deviations). First, we note that the PPV values obtained by the method without linear grouping (normal) were markedly inferior to those obtained by the method with linear grouping (grouping) for all considered scenarios. Besides, in most cases, and specially for the cases with small number of samples ($m = 20$), the SIM values obtained by the normal method also were smaller to the obtained by linear grouping. Thus, it is notable that the linear grouping presents better estimation power for higher dimensionalities, since the method search starts from a predictor set with a relatively large dimension ($k' = 5$). For a large number of samples ($m = 50$), the difference in the performances decreases, as expected, although the linear grouping method still presents better PPVs.

An important issue in feature selection is to obtain the subset with the correct degree, i.e., the correct dimensionality of the feature set. This is an open problem which needs to be carefully analysed. In cases where the number of samples is very limited, it is difficult to obtain the exact dimensionality of the feature set, specially if this set possesses large dimension. Considering networks in general, a node is considered an

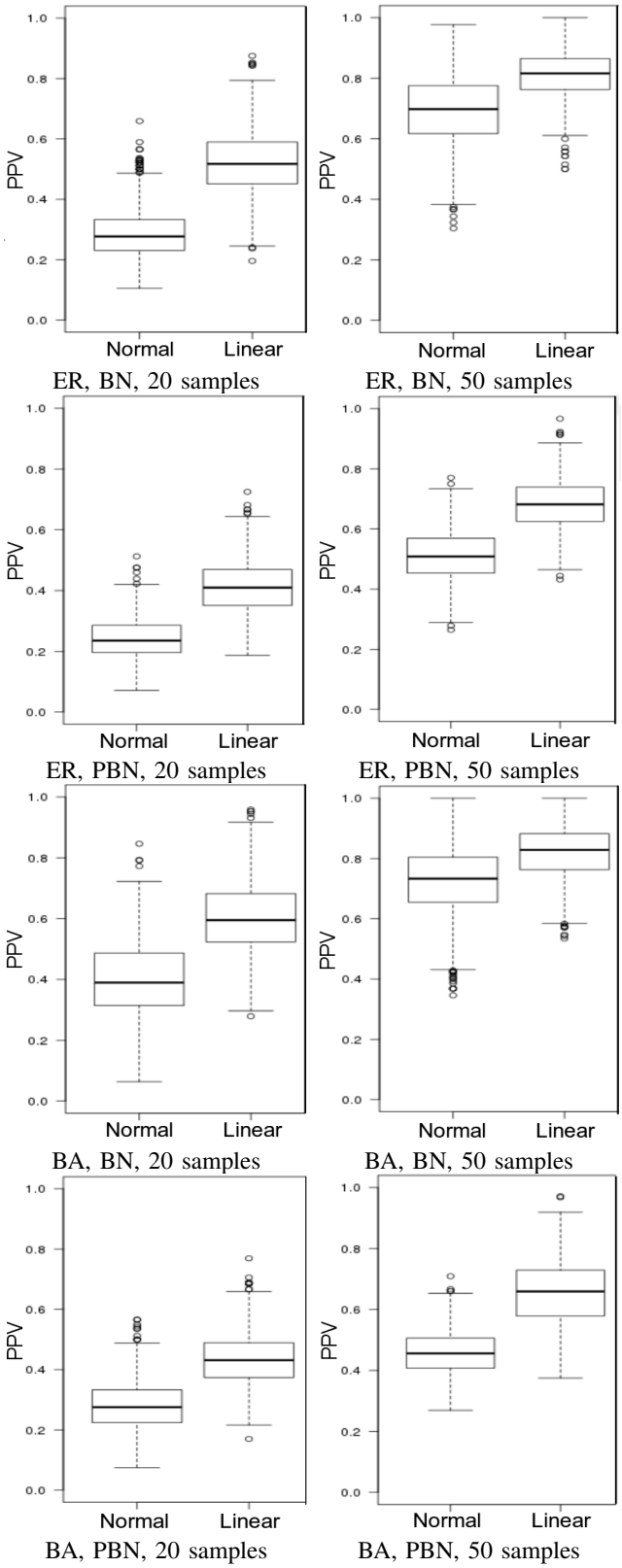

Fig. 2. Boxplots representing PPV values of 900 inferred networks. Each panel contains two boxplots: normal (left); linear grouping (right).

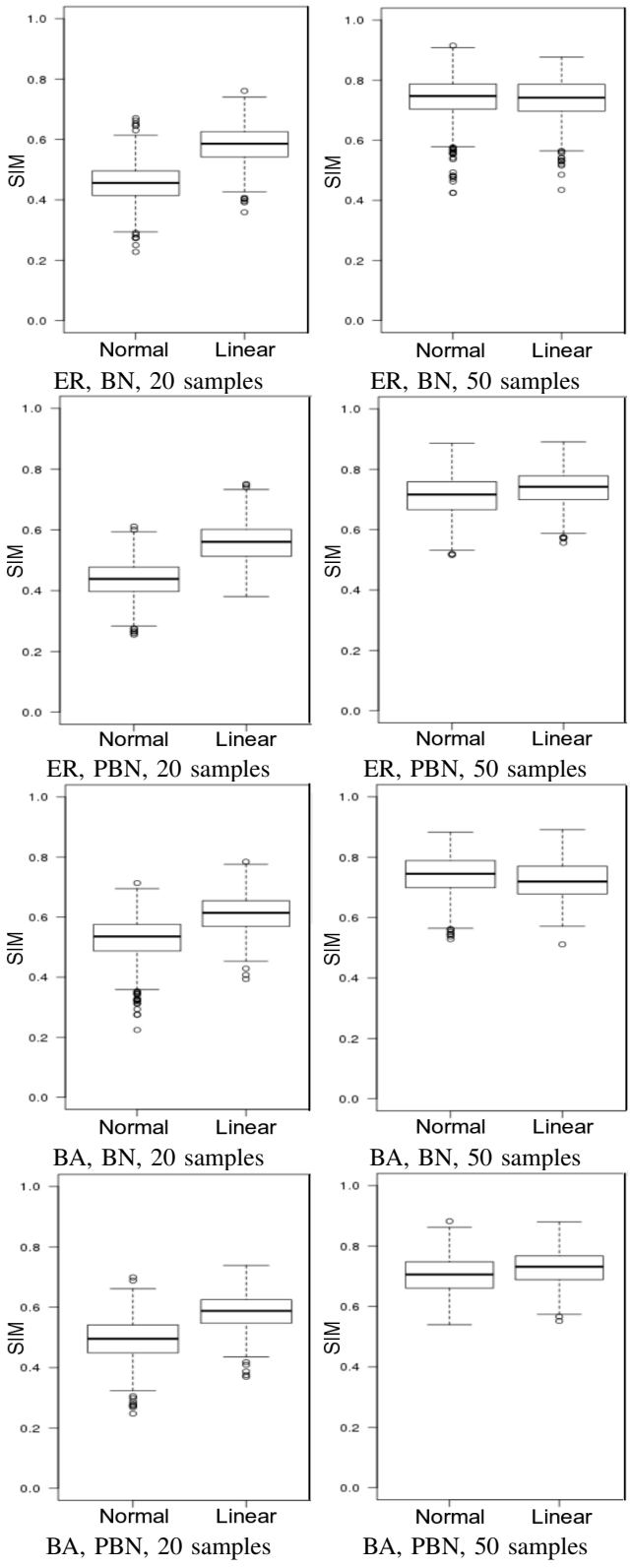

Fig. 3. Boxplots representing similarity (SIM) values of 900 inferred networks. Each panel contains two boxplots: normal (left); linear grouping (right).

TABLE III. AVERAGES AND STANDARD DEVIATIONS OF THE PPV AND SIM VALUES FOR THE RESULTS PRESENTED IN FIGURES 2 AND 3.

	m	method	PPV		SIM	
			BN	PBN	BN	PBN
ER	20	normal	0.29 ± 0.08	0.24 ± 0.06	0.46 ± 0.07	0.44 ± 0.06
		linear	0.52 ± 0.11	0.41 ± 0.09	0.58 ± 0.06	0.56 ± 0.06
	50	normal	0.69 ± 0.12	0.51 ± 0.08	0.74 ± 0.07	0.71 ± 0.06
		linear	0.81 ± 0.08	0.68 ± 0.08	0.74 ± 0.06	0.74 ± 0.06
BA	20	normal	0.40 ± 0.12	0.28 ± 0.08	0.53 ± 0.07	0.49 ± 0.07
		linear	0.60 ± 0.12	0.43 ± 0.09	0.61 ± 0.06	0.58 ± 0.06
	50	normal	0.72 ± 0.12	0.46 ± 0.07	0.74 ± 0.07	0.70 ± 0.06
		linear	0.82 ± 0.09	0.65 ± 0.11	0.72 ± 0.06	0.73 ± 0.06

input hub if it is influenced by a relatively large number of variables. Thus, estimating hubs from a limited number of samples is a challenge. By grouping the instances of the predictors, the number of parameters to be estimated decreases, which theoretically facilitates the detection of hubs. In this way, we analised the degree distribution of the nodes present in the networks inferred by the normal method and the linear grouping method, as well as the degree correlations taking the groundtruth network nodes as reference.

The histograms presented in Figure 4 show the degree distribution of the groundtruth networks and the networks inferred by normal and linear grouping methods. We can see that the linear grouping distributes the degrees more adequately than the normal method. This is due to the difficulty with which the normal method has to deal with non-observed instances. Since the search algorithm starts with degree $k' = 5$, initially the conditional probability tables present $2^5 = 32$ possible instances. For this situation with only 20 timepoint samples ($m = 20$) at least 12 instances are not observed, which implies that the best subset of size 5 obtained unlikely will be adequate in this case. In addition, by removing elements from this initial set, its properly contained subsets do not tend to be satisfactory, leading to a significant concentration in degree 4. This is exactly the degree for which the number of samples ($m = 20$) begins to exceed the number of instances ($2^4 = 16$). Such problem does not occur for linear grouping, since it tends to choose a good predictor subset with degree $k' = 5$ as starting point. Table IV presents the correlations between the degrees of the inferred network nodes and the degrees of the corresponding groundtruth nodes. These correlations corroborate the observations made for $m = 20$, since in this case the smallest correlations were observed for the normal method.

TABLE IV. CORRELATIONS BETWEEN THE INFERRED NETWORK NODE DEGREES AND THEIR CORRESPONDING GROUNDTRUTH NETWORK NODES DEGREES.

		$m = 20$		$m = 50$	
topology	method	BN	PBN	BN	PBN
Erdös-Rényi	normal	0.47	0.26	0.58	0.24
	linear grouping	0.53	0.36	0.56	0.30
Barabási-Albert	normal	0.52	0.27	0.58	0.16
	linear grouping	0.60	0.39	0.57	0.29

For a larger volume of samples ($m = 50$) in the Boolean networks scenario (noise free), the normal method estimates the conditional probability distribution tables more adequately, presenting degree correlations comparable to those obtained by the linear grouping method. However, in the probabilistic Boolean networks scenario (noisy), the linear grouping method presented better correlations.

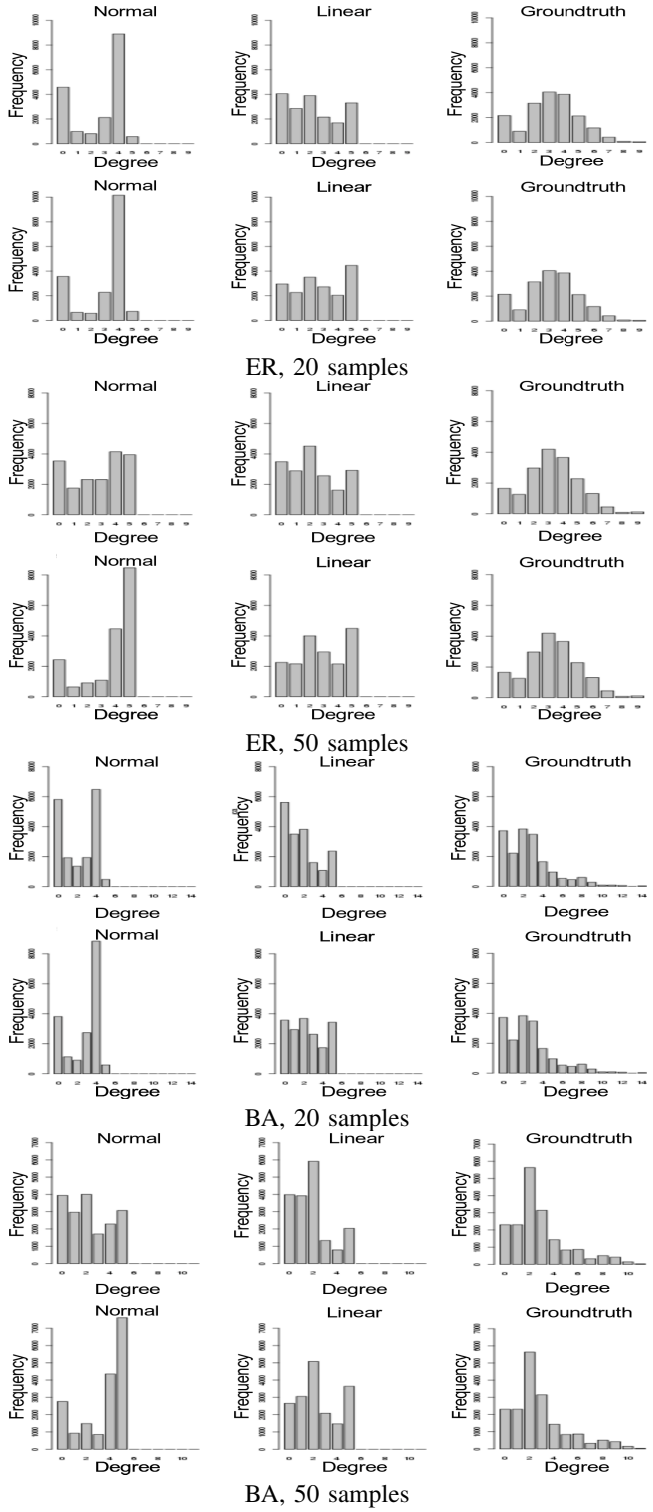

Fig. 4. Histograms of accumulated degree distribution for 900 inferred networks. Each panel presents two rows of histograms, where the first row corresponds to BN and the second row corresponds to PBN. The groundtruth histograms correspond to the degree distribuion accumulated for 30 networks multiplied by 30 just to maintain the scale compatible with the histograms of the methods.

VII. Conclusion

This paper presented a novel feature selection technique to infer gene networks modeled by discrete models such as Boolean networks. The principle of the proposed method is based on reduction of the number of estimation parameters (configurations) of the predictor values by grouping parameters in equivalence classes. The idea is to group in the same equivalence class parameters which lead to the same linear combination value according to the linear coefficients tbat optimize a given criterion function. Such clusters can be geometrically imagined as Boolean lattice cuts by parallel hyperplans. Two of these hyperplan necessarily intercept a unique vertex each, where those vertices have maximum Hamming distance (all bits flipped) between them. This approach effectively alleviates the curse of dimensionality, since the number of equivalence classes linearly grows with the dimension (number of predcitors), instead of growing exponentially as is the case of the original configurations. This is done at the cost of some information loss since the original configurations set is mapped to a smaller set (process similar to quantization).

The experimental results presented involving topological distances such as similarity (SIM) and positive predictive value (PPV) as inference evaluation criteria show that inferences involving larger number of predictors per gene tend to be benefited from linear grouping, specially if the number of samples is very small. On the other hand, for datasets with larger number of samples and inferences involving a smaller number of predictors per gene, there is no significant difference in the performances of the method without grouping (normal) and the linear grouping. This happens because the number of samples in this case is enough to allow a reasonable statistical estimation through the original configurations of the predictors. Such observation suggests that the method can be enhanced by a multiresolution analysis which consists on grouping only predictor configurations that are poorly observed in the data. This analysis has been successfully employed, for instance, to the binary filters design [10].

The linear grouping approach presented here creates unbalanced equivalence classes, which is a disadvantage. For instance, considering a Boolean lattice of 3 dimensions (3 features), two classes have only one configuration each, while the other two classes present three configurations each. Such unbalance is worsened with the dimension increasing. To mitigate this problem, one can consider a larger range of values for the linear coefficients, since in this paper only two possible values were considered: -1 (inhibition) and $+1$ (activation). The increasing on the number of possible coefficient values means to assign a variable weight for the activation or inhibition of a predictor. For example, a predictor with coefficient $+2$ would have twice the influence of a predictor with coefficient $+1$. Nevertheless, a larger range of coefficient values presents the disadvantage of having more equivalence classes to be considered, negatively impacting the statistical estimation power. In addition, the computational cost increases since a greater number of linear combinations need to be evaluated.

Other grouping modes can be investigated and evaluated. A possibility would be consider equivalence classes containing configurations close to each other according to some distance criterion. For example, each equivalence class could contain

only two neighbor configurations (Hamming distance equal to 1). From the geometrical point of view, such grouping partition the Boolean lattice in parallel edges. A disadvantage of this kind of grouping is that the number of classes is equivalente to half of the number of configurations, i.e., it increases exponentially with the dimension. Notwithstanding, such grouping still can be a more appealing alternative for a small dimension (smaller or equal to 5).

ACKNOWLEDGEMENT

We would like to thank FAPESP grant # 2011/50761-2, CNPq, CAPES and NAP eScience - PRP - USP for the financial support.

REFERENCES

[1] T. Akutsu, S. Miyano, and S. Kuhara. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In *Pacific Symposium on Biocomputing*, volume 4, pages 17–28, 1999.

[2] R. Albert and H. G. Othmer. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster. *Journal of Theoretical Biology*, 223(1):1–18, 2003.

[3] M. Angeletti, R. Culmone, and E. Merelli. An intelligent agents architecture for dna-microarray data integration. Technical report, U. of Camerino, Italy, 2001.

[4] A. L. Barabsi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[5] J. Barrera, R. M. Cesar-Jr, D. C. Martins-Jr, R. Z. N. Vencio, E. F. Merino, M. M. Yamamoto, F. G. Leonardi, C. A. B. Pereira, and H. A. del Portillo. Constructing probabilistic genetic networks of *Plasmodium falciparum* from dynamical expression signals of the intraerythrocytic development cycle. In *Methods of Microarray Data Analysis V*, chapter 2, pages 11–26. Springer, 2007.

[6] M. Brun, E. R. Dougherty, and I. Shmulevich. Steady-state probabilities for attractors in probabilistic boolean networks. *Signal Processing*, 85(10):1993–2013, 2005.

[7] M. I. Davidich and S. Bornholdt. Boolean network model predicts cell cycle sequence of fission yeast. *PLoS ONE*, 3(2):e1672, 2008.

[8] P. D'haeseleer, S. Liang, and Roland Somgyi. Tutorial: Gene expression data analysis and modeling. In *Pacific Symposium on Biocomputing*, Hawaii, January 1999.

[9] E. R. Dougherty. Validation of gene regulatory networks: scientific and inferential. *Briefings in Bioinformatics*, 12(3):245–252, 2011.

[10] E. R. Dougherty, J. Barrera, G. Mozelle, S. Kim, and M. Brun. Multiresolution analysis for optimal binary filters. *J. Math. Imaging Vis.*, 14(1):53–72, 2001.

[11] E. R. Dougherty, M. Brun, J. Trent, and M. L. Bittner. A conditioning-based model of contextual regulation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, August 2007.

[12] P. Erds and A. Rnyi. On random graphs. *Publ. Math. Debrecen*, 6:290–297, 1959.

[13] C. Espinosa-Soto, P. Padilla-Longoria, and E. R. Alvarez-Buylla. A gene regulatory network model for cell-fate determination during arabidopsis thaliana flower development that is robust and recovers experimental gene expression profiles. *Plant Cell*, 16(11):2923–2939, 2004.

[14] A. Faure, A. Naldi, C. Chaouiya, and D. Thieffry. Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22(14):e124–131, 2006.

[15] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7:601–620, 2000.

[16] M. Hecker, S. Lambeck, S. Toepfer, E. van Someren, and R. Guthke. Gene regulatory network inference: data integration in dynamic models-a review. *Biosystems*, 96:86–103, 2009.

[17] I. Ivanov and E. R. Dougherty. Modeling genetic regulatory networks: continuous or discrete? *Journal of Biological Systems*, 14(2):219–229, 2006.

[18] H. D. Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[19] Guy Karlebach and Ron Shamir. Modelling and analysis of gene regulatory networks. *Nat Rev Mol Cell Biol*, 9(10):770–780, 2008.

[20] S. A. Kauffman. Homeostasis and differentiation in random genetic control networks. *Nature*, 224(215):177–178, 1969.

[21] A. Kelemen, A. Abraham, and Y. Chen. *Computational Intelligence in Bioinformatics*. Springer, 2008.

[22] H. Lahdesmaki and I. Shmulevich. On learning gene regulatory networks under the boolean network model. *Machine Learning*, 52:147–167, 2003.

[23] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang. The yeast cell-cycle network is robustly designed. *Proc. Natl. Acad. Sci. USA*, 101(14):4781–4786, 2004.

[24] S. Liang, S. Fuhrman, and R. Somogyi. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. In *Pacific Simposium on Biocomputing*, volume 3, pages 18–29, 1998.

[25] F. M. Lopes, D. C. Martins-Jr, J. Barrera, and R. M. Cesar-Jr. A feature selection technique for inference of graphs from their known topological properties: revealing scale-free gene regulatory networks. *Information Sciences*, 272:1–15, 2014.

[26] F. M. Lopes, D. C. Martins-Jr, and R. M. Cesar-Jr. Feature selection environment for genomic applications. *BMC Bioinformatics*, 9(451), 2008.

[27] F. M. Lopes, S. S. Ray, R. F. Hashimoto, and R. M. Cesar-Jr. Entropic biological score: a cell cycle investigation for grns inference. *Gene*, 541:129–137, 2014.

[28] D. Marbach, R. J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, and G. Stolovitzky. Revealing strengths and weaknesses of methods for gene network inference. *Proceeings of the National Academy of Sciences*, 107(14):6286–6291, 2010.

[29] E. J. McCluskey. Minimization of boolean functions. *Bell Syst Tech, J*, 35(5):1417–1444, 1956.

[30] D. Nam, S. Seo, and S. Kim. An efficient top-down search algorithm for learning boolean networks of gene expression. *Machine Learning*, 65:229–245, 2006.

[31] B. Ristevski. A survey of models for inference of gene regulatory networks. *Nonlinear Analysis: Modelling and Control*, 18(4):444–465, 2013.

[32] L. Sánchez and D. Thieffry. A logical analysis of the drosophila gap-gene system. *Journal of Theoretical Biology*, 211(2):115–141, 2001.

[33] D. Shalon, S. J. Smith, and P. O. Brown. A dna microarray system for analyzing complex dna samples using two-color fluorescent probe hybridization. *Genome Res*, pages 639–45, 1996.

[34] I. Shmulevich and E. R. Dougherty. *Genomic Signal Processing*. Princeton University Press, New Jersey, 2007.

[35] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, 2002.

[36] M. P. Styczynski and G. Stephanopoulos. Overview of computational methods for the inference of gene regulatory networks. *Computers & Chemical Engineering*, 29(3):519–534, 2005.

[37] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Elsevier, Academic Press, Amesterdam, New York, 2006.

[38] V. E. Velculescu, L. Zhang, B. Vogelstein, and K. W. Kinzler. Serial analysis of gene expression. *Science*, 270:484–487, 1995.

[39] Z. Wang, M. Gerstein, and M. Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nat Rev Genet*, 10(1):57–63, 2009.

[40] Y. Zhang, M. Qian, Q. Ouyang, M. Deng, F. Li, and C. Tang. Stochastic model of yeast cell-cycle network. *Physica D*, 219(1):35–39, 2006.