2014 IEEE 14th International Conference on Bioinformatics and Bioengineering

Gene networks inference through one genetic algorithm per gene

Ray Dueñas Jimenez, David Correa Martins-Jr, Carlos Silva Santos
Center of Mathematics, Computation and Cognition
Federal University of ABC (UFABC)
Santo André-SP, Brazil
Emails: {ray.jimenez,david.martins,carlos.ssantos}@ufabc.edu.br

***Abstract*—Gene regulatory networks (GRN) inference from gene expression data is an important problem in systems biology field, in which the main goal is to comprehend the global molecular mechanisms underlying diseases for the development of medical treatments and drugs. This problem involves the estimation of the gene dependencies and the regulatory functions governing these interactions to provide a model that explains the dataset (usually obtained from gene expression data) on which the estimation relies. In this work a method based on genetic algorithms to infer gene networks is proposed. The main idea behind the method consists in applying one genetic algorithm for each gene independently, instead of applying a unique genetic algorithm to determine the whole network as usually done in the literature. Besides, we propose the application of a network inference method to generate the initial populations to serve as more promising starting points for the genetic algorithms than random populations. To guide the genetic algorithms, we propose the use of Akaike information criterion (AIC) as fitness function. Results obtained from inference of artificial Boolean networks show that AIC correlates very well with popular topological similarity metrics even in cases with small number of samples. Besides, the benefit of applying one genetic algorithm per gene starting from initial populations defined by a network inference technique is evident according to the results.**

I. Introduction

The vital maintenance of an organism depends on several metabolic pathways regulated by gene expression networks. Nowadays, the gene regulatory networks (GRN) inference problem has increasingly attracted the attention of researchers, due to the huge volume of gene expression data generated for many species and specific conditions. Nevertheless, the inference of gene regulatory networks (GRN) is an open problem [25]. A common challenge presented by gene expression analysis is the large number of genes (variables) with just a few dozens of samples (experiments), which demands the development of statistical and computational techniques to alleviate the estimation error committed in the presence of small number of samples and high dimensionality. Other factors that contribute to the difficulty of this task are associated to the large degree of imprecision inherent to the gene expression measurements (noisy data), the large complexity of inter-relationship networks, and lack of prior information about many biological organisms.

There are essentially two main approaches to model the complex networks of gene interactions: 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 Bayesian Networks [13], Boolean Networks [20] and the Probabilistic Boolean Networks [32]. Although the continuous approaches offer a detailed comprehension 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 for modeling and simulation of many biological process networks [14], [25], [29].

In the context of discrete models, Boolean Networks represent an appropriate model to generalize and capture the global bahavior of biological systems, especially 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 loss of information as a consequence of the data quantization. However, the data quantization is exactly what makes the Boolean model simpler [16]. Many methods were proposed to infer gene networks modeled as Boolean Networks [3], [1], [21], [27]. Such methods perform an exhaustive search over the state transition matrix of discretized gene expression data.

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, especially when the number of samples is small. This makes the problem quite complex, since the number of samples is limited and the data is subject to experimental noises. The inference process requires a good modeling framework combined with very well designed search or learning procedures. In recent years, evolutionary algorithms have been employed in a multitude of situations that present a very large solution space [28]. For instance, genetic algorithms have been applied to learn the network structure of regulatory pathways of continuous models [30]. Recently, Mendoza *et al* [26] proposed a genetic algorithm to infer GRNs, modeled as Boolean networks, from experimental data. In this work, the topologies are codified by integers representing the indices of the predictors for each gene in which the maximum number of predictors per gene needs to be determined *a priori*, the initial population is randomly generated with the restriction of one predictor per gene, and the fitness function is based on the Tsallis entropy with a penalty factor that needs to be appropriately tuned to achieve networks with good balance between complexity (number of links) and

978-1-4799-7502-0/14 $31.00 © 2014 IEEE
DOI 10.1109/BIBE.2014.9

consistency with the data.

In this paper we propose a GRN inference method based on genetic algorithms, where it is applied one genetic algorithm per gene independently with the aim to obtain the best predictor genes for each target gene. Such strategy is different from the applied by existing methods for the same purpose, which employ a unique genetic algorithm to infer the whole network. Another strategy proposed here consists on generating the initial population based on existing GRN inference methods (for instance, the probabilistic gene networks approach proposed in [6]) to obtain starting points more promising than those provided by randomly generated initial population, as usually done by genetic algorithms proposed in the literature. Besides, we adopt the Akaike information criterion (AIC) as fitness function, which is based on likelihood of the dataset be generated by the network, embedding a factor to penalize the model complexity, which increases with the dimensionality of the predictor subsets [2], [9].

With the objective of analysing the results of the proposed method, experiments involving artificial Boolean networks generated by complex network models, such as the random model Erdös-Rényi (ER) [12] and the scale-free model Barabási-Albert (BA) [5], were performed. Moreover, the proposed method was compared to a recent technique proposed by Mendoza *et al*, which is also based on genetic algorithm to infer gene networks [26].

This manuscript is organized as follows. Next section presents the Boolean Networks model foundations, including the Probabilistic Boolean Networks model as its stochastic version. Section III discusses the Probabilistic Gene Networks (PGN) approach. Section IV describes the proposed GA method for GRN inference. Some experimental results are discussed in Section V. This text is concluded in Section VI.

II. Boolean Networks

A Boolean network (BN) $B = (V, \mathbf{F})$ of n variables (genes) is defined by a set of n nodes $V = \{g_1, ..., g_n\}, g_i \in \{0, 1\}$, and a vector of n Boolean functions $\mathbf{F} = (f_1, f_2, ..., f_n), f_i : \{0, 1\}^n \rightarrow \{0, 1\}$. Each node g_i represents the state or expression of the gene i and each function f_i is a predictor function of g_i. The states of all genes in $B = (V, \mathbf{F})$ are synchronously updated in each step (or timepoint t) according to their predictor functions, i.e., $g_i(t + 1) = f_i(g_1(t), ..., g_n(t)) = \mathbf{F}_i(\mathbf{g}(t))$. In other words, the next state $\mathbf{g}(t+1)$ is obtained by application of the vector $\mathbf{F}$ of n functions to the current state $\mathbf{g}(t)$. The vector $\mathbf{F}$ is the transition function of the Boolean network $B = (V, \mathbf{F})$.

As each component f_i of $\mathbf{F}$ is a function that can depend of all n variables at most, the number of possible functions for a given gene i is 2^{2^n}, since the number of possible input values is 2^n where each input leads to one of the two possible outputs (0 or 1). Hence, the number of possible transition functions $\mathbf{F}$ (i.e., the number of different Boolean networks of size n) is $2^{n(2^n)}$, which is the total search space of the inference problem. Of course in most scenarios the network topologies are sparse, meaning that each variable depends on just a small fraction of other variables. Biological systems such as GRNs compose one of these scenarios [15].

Thus, the main concern regarding GRN inference is to find the correct network topology based on experimental data. Even considering that each gene depends on a fixed small number of genes (predictors), *e.g.* 2, the number of possible topologies is $\binom{n}{2}^n$, which is still huge even for $n \leq 10$. Besides, although the average number of predictors per gene is small, such number varies greatly from gene to gene. In fact, some genes may act as hubs, possessing prominently large number of predictors [5]. This feature makes the GRN inference problem even more challenging, requiring very well designed methods to address it.

A. Probabilistic Boolean Networks

The cell is an open system prone to receive external stimuli. Depending on the external conditions in a given time instant, the cell can change its dynamics [31]. Thus, to embed a stochastic character in the Boolean networks, the Probabilistic Boolean Networks model (PBN) has been proposed [32]. This model considers genes as binary values which are described by a set of Boolean predictor functions, where each function has a specific probability to be applied. Consequently, a BN is a specific type of PBN where each gene presents 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 [8], [11].

III. Probabilistic gene networks

The expression profile of a given target gene in a GRN usually is determined by the expression profile of a subset of genes called *predictors*. To search for the gene subset (predictors), feature selection methods can be applied to select the subset with the largest information content about the values of a target gene. In particular, the probabilistic gene networks (PGN) approach [6], [23], [24], [22] follows the feature selection principle: for each target gene, it is performed a search for the predictor subset that best describes the target behavior according to a criterion function which evaluates the prediction quality based on gene expression signals. Barrera *et al* discuss this approach in the context of the analysis of dynamic expression signals of Plasmodium *falciparum* cell cycle (one of the malaria agents), providing interesting biological results [6]. Each target gene in a given time instant depends only on its predictor values in the previous timepoint, since this approach assumes that the temporal samples follow a first order Markov chain. The transition function is homogeneous (the function does not change with time), almost deterministic (from any state, there exists a preferential state to which the system transits at the next timepoint) and conditionally independent (the value of a given gene depends only on the values of its predictors). These assumptions are important simplifications due to the limited number of samples typically available in real data. All GRN inference methods discussed from now on, including our proposed method based on genetic algorithms (Section IV), follow the axioms of the PGN model.

IV. Genetic Algorithms for GRN inference

A. Overview

Letting n be the number of genes present in the network, the problem of the search for the ideal topology presents a super-exponential search space ($2^{n \times n}$, since each adjacency matrix cell is a binary value, indicating presence or absence of an edge between two genes). One of the advantages of following the PGN model (Section III) is due to the fact that each gene depends only on its predictors. Thus, it is possible to divide the problem in n feature selection subproblems (one subproblem for each gene), where the search space of each subproblem is exponential (2^n possible candidate predictor sets). In this way, since the exhaustive search is still unfeasible even for a moderate number of genes, approximation algorithms such as genetic algorithms are required.

Here we propose a method to infer gene interaction networks based on genetic algorithms. The method takes as input the gene expression data and returns a network topology which tries to satisfactorily describe the data. Once retrieved the network topology, the logical dependencies among genes can be derived from the expression data. In this work, the Boolean Networks (BN) model was adopted, which follows the PGN axioms.

The procedure of the proposed method is outlined in Figure 1. It consists on dividing the gene network inference in n genetic algorithms (one for each gene), where n is the number of genes in the network. First, for a particular gene, an initial population of predictor sets is inferred through the application of feature selection by exhaustive search guided by mutual information (from now on, such method is called Exhaustive Search by Mutual Information - ESMI) and follows the PGN axioms (Section IV-B). Next, these predictor sets are codified and taken as input by the genetic algorithms (Section IV-C). These algorithms are guided by the Akaike Information Criterion (AIC), which evaluates the predictor set of each gene based on the input data (Section IV-D). Finally, the genetic operators crossover and mutation are applied to the predictor sets of the network genes to produce the next populations (generations), one new population for each gene (Sections IV-E and IV-F). Each generation is submitted to a new round of evaluation, crossover and mutation iteratively until a stop criterion be satisfied, obtaining the final populations. For each generation, the best chromosome (with the least AIC) of the current generation is compared with the best chromosome obtained so far. If the first is the best, then it is considered as comparison reference for the next generations. Thus, the best chromosome (predictor subset) obtained for each gene composes the final network. The following sections detail the aspects of the proposed method.

B. Initial populations generation

Usually genetic algorithms start with a randomly generated initial population. In [26], for instance, each gene starts with only one randomly chosen predictor, forming an initial network that composes the random initial population of networks. Here, we propose to obtain the initial population of predictor subsets for each gene by appling a feature selection method. The subsets with the higest information content about the values of a given gene compose its initial population. We apply the ESMI method in which all predictor subsets of fixed size k previously defined are examinated and ranked according to their mutual information with the target gene,

After the ranking, the best c predictor subsets compose the initial population of the target gene (chromosomes). Such process is repeated for every gene, thus creating the initial populations of the chromosomes of all genetic algorithm processes (one per gene).

C. Chromosomal codification

In genetic algorithms, chromosomes are representations of possible solutions of the search problem. A solution to the GRN inference problem can be represented by a graph of dependences among genes and their logical rules. Following the genetic algorithm approach described in [26], GRNs are modeled as Boolean networks in which each gene is expressed or not (1 or 0) based on a Boolean function of other genes which may include the target itself (self-loops). The chromosomes need to contain information about the graph topology, which can be codified as adjacency matrix, adjacency vector or predictor sets. Once the topology is defined, the logical rules can be estimated directly from the data, which can be done by prediction (classification) error minimization (likelihood maximization).

Nevertheless, unlike the method proposed in [26], our proposed technique consists of applying isolated genetic algorithms (one for each gene). For this, the chromosome representation is given by a set of indices, in which each index corresponds to a given predictor gene. By using this representation, a target gene in a given generation contains a collection of predicor sets (chromosomes) composing its current population. More formally, let PS_i be a predictor set (chromosome) of gene i. If the element $j \in PS_i$, then gene i depends on gene j. Otherwise ($j \notin PS_i$), the gene i does not depend on gene j. As a consequence of this representation, there is no restriction on the number of predictors that can be associated to a certain target gene.

D. Fitness function

A fitness function is needed to evaluate the current populations of solutions and guide the genetic algorithms to approximately optimal solutions. For each solution, it is assigned a fitness value indicating the capability of the solution to survive, reproduce and maintain its genetic characteristics along the next generations. In this case each predictor set needs to be evaluated based on how well it explains the considered gene expression profile. We adopted the Akaike Information Criterion (AIC) [2] as fitness function. AIC estimates the probability of the gene values to be explained by a given predictor set based on the data, including a factor to penalize the sets of larger dimensionality (the number of parameters to be estimated increases with the number of predictors). AIC offers a compromise between the complexity and the model fitting. Formally, the AIC for a given target gene G_i and a predictor set $\mathbf{X_i}$ is defined by Equation 1:

$$AIC(G_i, \mathbf{X_i}) = 2K(\mathbf{X_i}) - 2ln(L(G_i, \mathbf{X_i})) \qquad (1)$$

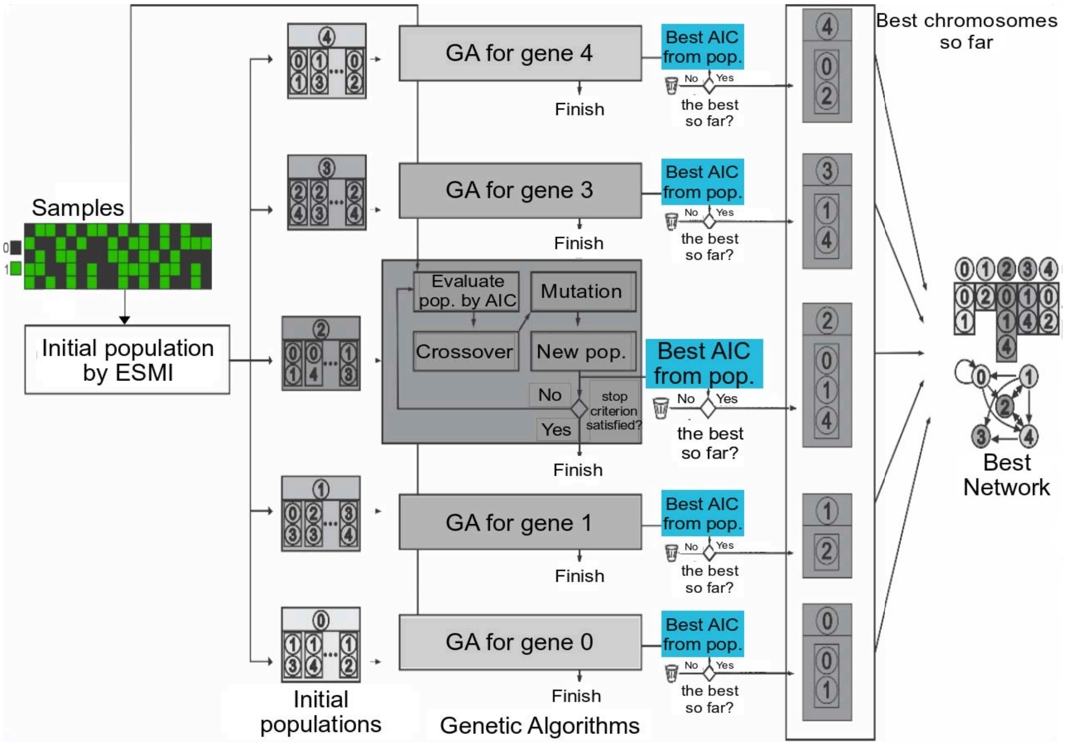


Fig. 1. Outline of the proposed method for gene networks inference. In this example, $n = 5$.

where K is the number of parameters present in the statistical model, and $L(G_i, \mathbf{X_i})$ is the maximum likelihood function for the gene G_i and the predictor set $\mathbf{X_i}$. Given m samples and the target gene $G_i = \{0, 1\}$ which depends on the subset of predictors $\mathbf{X_i} = \{0, 1\}^{k_i}$, where k_i is the number of predictors of the gene G_i, L can be estimated from data as described in Equation 2:

$$L(G_i, \mathbf{X_i}) = \prod_{\mathbf{x_i} \in \{0,1\}^{k_i}} \prod_{g_i \in \{0,1\}} P(g_i|\mathbf{x_i})^{mP(\mathbf{x_i}, g_i)} \tag{2}$$

where $P(g_i|\mathbf{x_i})$ is the conditional probability of $G_i = g_i$ given $\mathbf{X_i} = \mathbf{x_i}$, and $P(\mathbf{x_i}, g_i)$ is the joint probability of $\mathbf{X_i} = \mathbf{x_i}$ and $G_i = g_i$. These probabilities are estimated from the data.

Besides, $K(\mathbf{X_i})$ is the number of statistical parameters to be estimated for gene G_i, given by Equation 3:

$$K(\mathbf{X_i}) = 2^{1+k_i} \tag{3}$$

i.e., given that a gene G_i depends on k_i binary predictors, there are 2^{k_i} possible values for these predictors. As G_i is binary, two conditional probabilities need to be estimated for each possible value (instance) which the predictors can assume, one for $G_i = 0$ and another for $G_i = 1$. Thus, the number of statistical parameters to be estimated is $2 \times 2^{k_i} = 2^{1+k_i}$.

The best feature subsets tend to have smaller AIC values, since a larger number of predictors implies in a greater K (the risk of overfitting increases), while larger values of L indicate a better explanation of the data by the predictor set.

As the inferred network is obtained by the composition of the best subset (chromosome) of each gene obtained so far, the AIC of a network can be obtained by $\sum_i AIC(G_i, \mathbf{X_i^*})$, where $\mathbf{X_i^*}$ is the best subset obtained for the gene G_i so far.

E. Crossover

The crossover phase should simulate the natural selection mechanism, in which the most adapted parents according to the chosen fitness function tend to generate more children, but allowing least capable parents to generate descendants as well, since individuals with small fitness may have peculiar genetic characteristics which lead to produce individuals with better solutions for a given problem. Crossover presents two main steps: selection and recombination. In our proposed technique for GRN inference, we adopted the roulette wheel method for selection of the individuals, where each individual (chromosome) is assigned a slice proportional to its fitness. Once the individuals are selected, they exchange parts of their chromosomes by a recombination operation.

1) Selection: In selection step, a subset of individuals (chromosomes) of a given population is chosen to reproduce, generating the next population. We adopted the roulette wheel method, in which the probability of a chromosome to be chosen is inversely proportional to its AIC. In this way, the best chromosomes tend to compose the majority of the individuals to be recombined. The roulette requires the sum of of AIC inverses, using this value to determine the probability of each individual to be chosen, as in Equation 4:

$$R(i) = \frac{\frac{1}{AIC_i}}{\sum_{i=1}^{n} \frac{1}{AIC_i}} \quad (4)$$

2) Recombination: In recombination phase, the selected individuals form random pairs. Each pair recombines their genetic information to generate two children which will compose the next generation of individuals. In the present work, the recombination of a pair of individuals (parents) starts from the creation of a new set resulting from the union with duplicates of their indices. Then, the elements of this union are shuffled, generating a vector containing these elements in arbitrary order. Finally, one of the vector positions is randomly chosen as one-point cut, dividing the vector in two, one for each child. If there are elements (predictors) duplicated for a given child, one of these predictors is transferred to the other child.

F. Mutation

Aiming to escape from the local minimum solutions, mutation is an important genetic algorithm operation to change some punctual features of the individuals. Since the proposed method applies one genetic algorithm for each gene, where a gene presents a population composed by predictor subsets (chromosomes), a small percentage of these chromosomes should be changed. The number of chromosomes to be changed is calculated in each interaction according to a decreasing linear function which goes from max_{mut} (maximum number of mutated chromosomes) to 0 in 100 iteractions (generations). The variable it indicates the current iteraction. When r repetitions of the best AIC obtained are reached (convergence), it is resetted to 0, consequently making MC (number of chromosomes to be mutated) to be resetted to max_{mut}. MC is given by Equation 5:

$$MC = \max\{0, \lceil \frac{(100 - it) \times max_{mut}}{100} \rceil\} \quad (5)$$

Once determined the number of chromosomes to be mutated in a certain iteraction, the chromosomes are randomly chosen. Letting C be one of the chromosomes selected to be mutated, in a traditional mutation, any gene G_i from the network would be randomly chosen, either to be included to C if $G_i \notin C$, or to be excluded from C otherwise ($G_I \in C$). The problem with this mutation strategy is that the number of predictors of a chromosome is usually much smaller than the number of genes in the network, which implies that the majority of mutations include the randomly chosen gene instead of excluding it, inducing a degree increase in almost all chromosomes. In order to avoid such undesirable behavior, we propose a strategy based on genes swapping. That is, if $G_i \in C$, then G_i will be swapped by another $G_j \notin C$ randomly chosen. Otherwise ($G_i \notin C$), it replaces one of the genes $G_j \in C$ randomly chosen. Only one gene of C is swapped by another gene, i.e., only one index of C is changed at a time.

V. Experimental results

In this section we perform four sets of experiments: (A) evaluation of AIC as fitness function to guide the genetic algorithms (GA); (B) assessment of the benefits of generating the initial populations with the ESMI method as discussed in Section III; (C) comparison of the proposed method (GAs starting from random initial populations and from initial populations generated by ESMI method) with the ESMI method itself; (D) comparison involving the Mendoza *et al* method [26].

All experiments were performed with simulated data. We generated random networks according to two topological models: the random model Erdös-Renyi (ER) and the scale-free model Barabási-Albert (BA). Both network models require the average degree (number of preditors) parameter $\langle k \rangle$. In case of scale-free networks another parameter γ is required as decay factor of the power-law. We fix $\gamma = 2.5$, since the degree distribution of the elements of biological networks usually follows a power-law with $2 < \gamma < 3$ [17], [4], [22]. The network dynamics follows the PBN model: a small set of Boolean functions is assigned to each gene, where each function has a pre-defined probability of being applied. Such functions are randomly defined from the set of all k-variable Boolean functions, where k is the number of predictors of a given gene. At each timestep, one function from this set is selected to generate the considered gene value at the next instant, according to a given probability distribution (which we call "PBN functions probability distribution").

The parameter settings of the experiments are shown in Table I. The purpose was to analyze situations in which the number of data samples (timepoints) is very limited, since such situations are typical in real datasets.

TABLE I. Parameters used in the experiments.

Network topology models	{ER; BA}
Groundtruth network size (number of genes)	100
Average number of predictors (groundtruth)	3
Average number of predictors (initial populations)	3
Number of regulatory functions per gene (PBN)	3
PBN functions probability distribution	(0.98, 0.01, 0.01)
Number of samples (m)	{30; 60}
Number of GA iteractions (generations)	1000
Populations size	100
Minimum probability of each chromosome to be mutated	0
Maximum probability of each chromosome to be mutated	0.1
Convergence criterion for mutation (r)	10

In order to evaluate the results, we compare the inferred networks to the groundtruths using two topological similarities based on the number of true/false positives and true/false negatives: positive prediction value (PPV) and similarity (SIM) [10]. A true-positive edge is a directed edge present in both groundtruth and inferred networks, and a true-negative edge is a directed edge that is absent in both networks. Let TP, TN, FP and FN be the numbers of true-positive, true-negative, false-positive and false-negative edges, respectively. PPV is defined by $TP/(TP + FP)$, and SIM is defined by $\sqrt{TP/(TP + FN) \times TN/(TN + FP)}$.

For each combination of topological model, network size, and sample size, we generated 10 different groundtruth networks and then performed 10 simulations (100 datasets total). To account for random effects in GA execution, we performed 10 runs of the proposed method for each dataset, thus resulting 1000 inferred networks for each experiment. The same observation applies to the ESMI method: although it is deterministic, it is possible to become it stochastic due to the possibility of returning more than one subset tied in terms of mutual information (the ties are broken by randomly selecting one of the tied best subsets).

A. *AIC versus PPV and SIM*

Here we analyze the correlation between AIC and both PPV and SIM for four different parameter setups varying the number of samples and the size of the groundtruth networks, applying our proposed method with random initial populations.

Figure 2 illustrates the evolution of PPV and SIM in terms of AIC for a particular execution (most of the executions displayed similar behavior) along 1000 iteractions (generation), with BA topology and 30 samples (for 60 samples and for ER with 30 and 60 samples, the results were similar). The dots corresponds to the pairs (AIC, SIM) and (AIC, PPV) reached by the best inferred network in terms of AIC at a given generation. The dot colors indicate the generation ranges in which the corresponding networks were obtained: generations 1-200 (blue), 201-400 (dark green), 401-600 (light green), 601-800 (brown), 801-1000 (red). The corresponding Pearson correlations between AIC and SIM and between AIC and PPV are indicated in the panel titles. Based on these results, we can note that the method presents fast convergence, since after 200 generations, the AIC values become close to the minimum AIC obtained. The same happens to the topological similarities: SIM and PPV reach values close to the maximum obtained after 200 generatons.

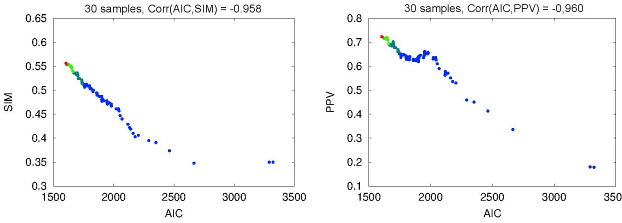


Fig. 2. Values of (AIC, SIM) and (AIC, PPV) obtained at each generation for a single execution of the proposed method starting from random initial populations considering BA topologies and 30 samples. The results were similar for 60 samples and for ER topology with 30 and 60 samples. Dot colors indicate the generations range in which the networks were obtained: generations 1-200 (blue), 201-400 (dark green), 401-600 (light green), 601-800 (brown), 801-1000 (red).

Considering all 1000 performed executions, the boxplots of the Figure 3 represents the distribution obtained for their corresponding correlations (SIM in the left panel and PPV in the right panel) considering BA topology (similar results were obtained for ER topology). The correlation results obtained by varying both topologies and number of samples are summarized in Table II. It is important to note that a larger number of samples implies in larger averages of absolute correlation values.

It is clear that all correlations are highly negative (absolute values larger than 0.9) for all experiments. Since the objective is to minimize AIC, these results indicate that the AIC has a great potential to drive the genetic algorithms toward better inference when the number of samples is minimally sufficient.

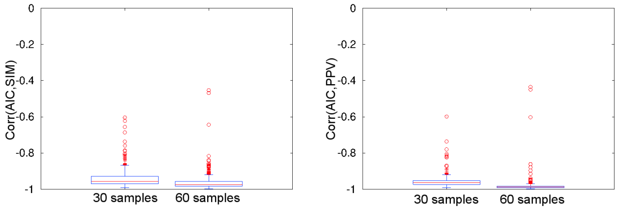


Fig. 3. Boxplots representing the distribution of the correlaes between (AIC, SIM) and (AIC, PPV) obtained from 1000 executions involving BA networks (similar results were obtained for ER topology).

TABLE II. AVERAGE (*avg*) AND STANDARD DEVIATION (*std*) OF THE CORRELATION COEFFICIENT BETWEEN AIC AND PPV/SIM.

		SIM		PPV	
		$m = 30$	$m = 60$	$m = 30$	$m = 60$
ER	avg	-0.96	-0.98	-0.94	-0.96
	std	0.03	0.03	0.05	0.04
BA	avg	-0.96	-0.99	-0.92	-0.96
	std	0.03	0.02	0.06	0.03

B. *Initial populations obtained by ESMI method*

The inference results can be greatly improved when the initial population is created by an inference method such as the ESMI. Figure 4 presents boxplots conrresponding to the distribution of SIM and PPV values obtained from 1000 inferred networks (10 groundtruth networks, 10 datasets simulated per groundtruth, and 10 executions per dataset) by our proposed method starting with random initial populations (called GA.random from now on) and starting with initial population obtained by ESMI searching for triplets (called GA.ESMI from now on). Although these results are shown only for BA topology, similar behavior was observed for ER topology. Table III shows the averages and standard deviations of SIM and PPV values for both ER and BA topologies. We can observe that the results obtained by GA.ESMI were superior to those obtained by GA.random, which shows the benefit of starting the genetic algorithms from subsets obtained from an inference method (ESMI in this case).

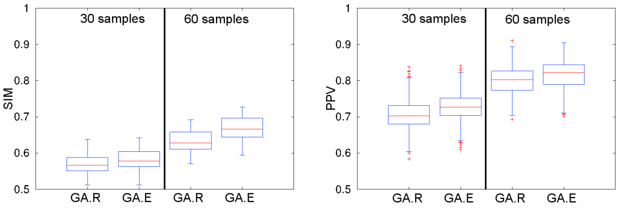


Fig. 4. Boxplots corresponding to SIM (left) and PPV (right) values from 1000 inferred networks considering BA topology. GA.R means GA.random while GA.E means GA.EMSI.

C. *Comparative analysis involving Mendoza et al method*

This section presents a comparative analysis of the methods GA.random, GA.ESMI, ESMI itself and the method proposed by Mendoza *et al* [26]. The experimental protocol in this section follows Table I, except for the number of gene expression samples (m). Here $m = 300$ wih 10 concatenations of 30

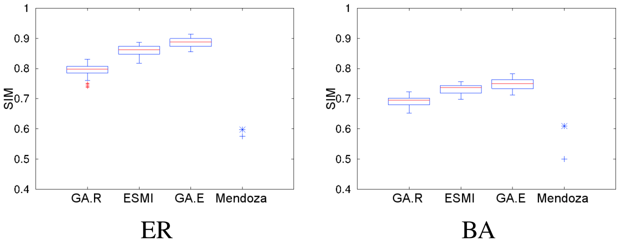
TABLE III. AVERAGES AND STANDARD DEVIATIONS FOR 1000 NETWORKS INFERRED BY THE PROPOSED METHOD WITH RANDOM INITIAL POPULATION AND WITH INITIAL POPULATION INFERRED BY ESMI. GA.R MEANS GA.RANDOM WHILE GA.E MEANS GA.ESMI

		SIM				PPV			
		$m = 30$		$m = 60$		$m = 30$		$m = 60$	
		GA.R	GA.E	GA.R	GA.E	GA.R	GA.E	GA.R	GA.E
ER	avg	0.572	0.585	0.68	0.747	0.661	0.682	0.841	0.873
	std	0.029	0.029	0.028	0.034	0.043	0.04	0.036	0.029
BA	avg	0.57	0.583	0.633	0.669	0.705	0.727	0.801	0.819
	std	0.025	0.027	0.027	0.032	0.038	0.038	0.037	0.039

samples, as done in [26]. The generation of each subset of 30 samples follows the PBN procedure previously described.

The Mendoza *et al* method executes the genetic algorithm 30 times to obtain a unique consensus network composed by the edges that are most frequent along these networks, following the "wisdom of crowds" principle [26]. Such consensus networks were evaluated only in terms of similarity (SIM). An important restriction presented by this method refers to the maximum degree k_{max} of the genes in the network that needs to be defined *a priori*. This method (called "Mendoza" from now on) was evaluated with $k_{max} = \{2, 3\}$.

Figure 5 presents boxplots representing the distribution of 1000 SIM values (from 10 executions taking as input 10 datasets generated by 10 groundtruth networks) obtained by the methods GA.random, GA.ESMI and ESMI, and two points illustrated by "$*$" and "$+$" corresponding to the results obtained by Mendoza method with $k_{max} = 2$ and $k_{max} = 3$ respectively. Table IV summarizes these results presenting averages and standard deviations. The best results for both ER and BA topologies were obtained by our proposed GA method with initial populations obtained by ESMI (GA.ESMI), while the Mendoza method presented inferior similarities.



ER

BA

Fig. 5. The boxplots correspond to the distributions of 1000 SIM values obtained by GA.random, GA.ESMI and ESMI methods. The symbols $*$ and $+$ correspond to the results obtained by Mendoza method with $k_{max} = 2$ and $k_{max} = 3$ respectively.

TABLE IV. AVERAGES (AVG) AND STANDARD DEVIATIONS (STD) OF THE RESULTS PRESENTED IN FIGURE 5.

		SIM				
		300 samples (concatenation of 10 sets of 30)				
		G.random	ESMI	G.ESMI	$k_{max} = 2$	$k_{max} = 3$
ER	avg	0.794	0.86	0.886	0.598	0.576
	std	0.021	0.02	0.016		
BA	avg	0.691	0.731	0.746	0.61	0.5
	std	0.016	0.019	0.02		

VI. CONCLUSION

In this paper, we propose an approach based on genetic algorithms to infer gene interaction networks modeled by Boolean networks and probabilistic Boolean networks. Their main advantages over other similar techniques are: (i) application of several independent genetic algorithms, one for each target gene; (ii) generation of initial populations with exhaustive search for subsets of fixed size (degree) guided by mutual information; (iii) the adoption of Akaike Information Criterion (AIC) as fitness function to guide the algorithm. AIC basically provides a measurement of the probability of the gene expression data samples to be generated by the inferred network according to its topology and logical rules, including a factor that penalizes topologies with an excessive number of edges (which implicates in an expressive increasing on the number of statistical parameters to be estimated) given the available data samples, avoiding overfitting.

The experiments performed based on data generated by artificial gene networks constructed by the complex network models Erdös Rényi (random) and Barabási-Albert (scale-free) indicate that the three strategies aforementioned jointly applied produce superior results, in terms of topological similarity, when compared to a recently published genetic algorithm based gene network inference [26]. The results of our proposed method were kept superior even without the application of the second strategy (initial populations randomly chosen, instead of obtaining them by an inference method). In addition, the initial population generation by the inference method (Exhaustive Search by Mutual Information - ESMI) presented benefits over the random generation, specially for a larger volume of samples.

With regard to the adopted fitness function, the experimental results showed that the AIC exhibits a good balance between the complexity and the quality of fit in situations with a limited number of samples, effectively guiding the genetic algorithms toward networks presenting good topological similarites with the grondtruth. In all presented scenarios by applying genetic algorithms from random initial populations, the average absolute correlations between the AIC and the topological metrics of similarity (SIM) and positive predictive value (PPV) were greater than 0.9. Moreover, the algorithms converge quickly, requiring about 200 iteractions (generations) to obtain the networks with the smallest AIC values and, consequently, the largest SIM and PPV values.

Although the proposed method showed much promise in terms of topological similarity, its validation can be furthered by taking into account not only topological features, but also dynamical aspects of the signals produced by the inferred networks, comparing them to the signal generated by the groundtruth network [10]. Thus, simpler topologies (smaller average number of edges) may explain the dynamics of the samples produced by the groundtruth almost as well as more complex topologies (larger average number of edges).

The known topological aspects about biological networks can be a valuable prior information to enhance the network inference methods. For example, Lopes *et al* [22] presented a feature selection method guided to obtain gene networks with scale-free topologies. In the case of our proposed genetic algorithms based method, something similar can be done, for instance, with regard to the crossover strategy. In the recombination step, instead of randomly choosing an one-point cut with uniform probability distribution, implying that the union of the parent chromosomes be divided close to the midpoint in most cases, resulting in two child chromosomes with similar lengths, such probability distribution could be changed to privilege cuts at the extremities, thus leading to many chromosomes with small degree and some chromosomes with very large degree. This could lead to a final network possessing scale-free characteristics.

Finally, the proposed method can be easily parallelizable, since each genetic algorithm (one per gene) can be processed independently of the others. Thus, it is possible that each processing core be responsible of obtaining the best predictor subset for one target gene. In this way, real sized networks (with thousands of genes) can be quickly inferred. Currently there are low cost parallel architectures such as the Graphical Processing Units (GPUs), which have been increasingly employed for general purposes in scientific applications. Regarding the gene networks inference application, Borelli *et al* implemented the exhaustive search by mutual information (ESMI) method discussed here in GPUs, obtaining speedups of the order of hundreds when compared to conventional multicore CPUs [7].

Acknowledgment

We would like to especially thank Carlos Fernando Montoya Cubas for the program codes to generate the groundtruth networks. We also thank FAPESP grant # 2011/50761-2, CNPq, CAPES and NAP eScience - PRP - USP for the financial support.

References

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

[2] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[3] 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.

[4] R. Albert. Scale-free networks in cell biology. *J Cell Sci*, 118(21):4947–4957, 2005.

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

[6] 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.

[7] F. F. Borelli, R. Y. de Camargo, D. C. Martins-Jr, and L. C. S. Rozante. Gene regulatory networks inference using a multi-gpu exhaustive search algorithm. *BMC Bioinformatics*, 14(S5), 2013.

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

[9] K. P. Burnham and D. R. Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer-Verlag, 2nd edition, 2002.

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

[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. Erdös and A. Rényi. On random graphs. *Publ. Math. Debrecen*, 6:290–297, 1959.

[13] 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.

[14] 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.

[15] D. Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks. *Bioinformatics*, 19:2271–2282, 2003.

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

[17] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabasi. The large-scale organization of metabolic networks. *Nature*, 467:651–654, 2000.

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

[19] G. Karlebach and R. 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] 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.

[22] 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(0):1–15, 2014.

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

[24] 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(2):129–137, 2014.

[25] 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.

[26] M. R. Mendoza, F. M. Lopes, and A. L. C. Bazzan. Reverse engineering of grns: An evolutionary approach based on the tsallis entropy. In *Proceedings of the 14th international conference on Genetic and evolutionary computation (GECCO)*, pages 185–192, Philadelphia, 2012.

[27] 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.

[28] S. K. Pal, S. Bandyopadhyay, and S. S. Ray. Evolutionary computation in bioinformatics: A review. *IEEE Transactions on Systems, Man and Cybernetics*, 36:601–615, 2006.

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

[30] A. Shin and Hitoshi Iba. Construction of genetic network using evolutionary algorithm and combined fitness function. *Genome Informatics*, 14:94–103, 2003.

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

[32] 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.