Reverse Engineering of GRNs: An Evolutionary Approach based on the Tsallis Entropy

Mariana R. Mendoza
Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, RS, Brazil
mrmendoza@inf.ufrgs.br

Fabrício M. Lopes
Federal University of Technology - Paraná
Cornélio Procópio, PR, Brazil
fabricio@utfpr.edu.br

Ana L. C. Bazzan
Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, RS, Brazil
bazzan@inf.ufrgs.br

ABSTRACT

The discovery of gene regulatory networks is a major goal in the field of bioinformatics due to their relevance, for instance, in the development of new drugs and medical treatments. The idea underneath this task is to recover gene interactions in a global and simple way, identifying the most significant connections and thereby generating a model to depict the mechanisms and dynamics of gene expression and regulation. In the present paper we tackle this challenge by applying a genetic algorithm to Boolean-based networks whose structures are inferred through the optimization of a Tsallis entropy function, which has been already successfully used in the inference of gene networks with other search schemes. Additionally, wisdom of crowds is applied to create a consensus network from the information contained within the last generation of the genetic algorithm. Results show that the proposed method is a promising approach and that the combination of a criterion function based on Tsallis entropy with an heuristic search such as genetic algorithms yields networks up to 50% more accurate when compared to other Boolean-based approaches.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: Life and Medical Sciences – Biology and genetics

General Terms

Algorithms

Keywords

Gene Regulatory Networks, Inference, Mutual Information, Tsallis Entropy, Boolean Networks, Genetic Algorithms

1. INTRODUCTION

A recent and challenging goal in bioinformatics research is to understand the nature and control of cellular function and the reasons why cellular systems fail in disease. This can be accomplished (at least partially) by revealing the structure of the organisms' underlying gene regulatory networks. Nowadays scientists are attempting to perform this task by investigating the behavior of genes in a holistic way, based on large-scale gene expression profiles. A major motivation is the believe that genes' activity is not isolated or independent of each other [25]. Indeed, genes compose intricated networks through which they work in concert to promote life sustainability. Therefore, the discovery of gene regulatory networks' structure through reverse engineering is at the same time an appealing and complex approach.

Gene regulatory networks (GRNs) are graph models that reflect the mechanisms and dynamics of gene expression and regulation by mapping the physical or influence interactions between genes of a particular organism. The nodes of this graph represent genes or genes' products, while the wiring describe the regulatory interactions between these elements. Once modeled, gene networks explain how genes are over- or under-expressed in response to perturbation signals and environmental changes. This is particularly interesting, for instance, for new drugs and treatments development. Additionally, this model represents an important tool for in silico experiments. Thus, there is a strong motivation for improving and developing new reverse engineering algorithms.

Several approaches have been already explored for modeling and inference of GRNs. On the graph-based course, one may consider, for instance, Bayesian networks [9, 11], Boolean networks [2, 14, 15], weight matrices [4, 20], among others. Regardless the model applied, in general, reverse engineering algorithms try to reveal the regulatory interactions that compose the networks' topology from experimental data. The main goal is to recover gene interactions in a global and simple way, by identifying the most significant connections [18]. However, there is an important challenge involved in this context: as experiments usually produce sparse data sets, the interactions between hundreds of genes must be learned from a few available samples [11], which not only raises the complexity of the inference problem, but also impairs its accuracy. For this reason, the use of heuristic procedures has been recently explored.

Previous works [4, 7, 20, 23] have implemented a reverse engineering method based on genetic algorithms (GA) to reconstruct GRNs. In [4] and [20], authors represented the GRNs as weight matrices and aimed at minimizing the divergence between the expression patterns of the inferred network and the target network. Although the algorithms have

GECCO'12, July 7–11, 2012, Philadelphia, Pennsylvania, USA.

helped in the identification of networks with several dozen genes to significant accuracy, the large set of parameters to be inferred limits their application to realistic problems. In [7], the networks were modeled as Bayesian networks and evaluated based on metric scoring functions implemented by Weka[1], namely the Akaike Information Criterion (AIC) score and the Minimum Description Length (MDL) score. In this work, the correct topological ordering of nodes was predicted by the algorithm but the set of relationships between nodes could not be completely reconstructed. GAs were also explored in [23], in which a Boolean modeling framework was applied and the GAs' fitness function was implemented following the concepts of the *consistency problem*[2]. Authors have found the accuracy of the inferred networks to be satisfactory, but the generated models had the disadvantage of containing many false positives interactions. Furthermore, the fitness function applied in this work was not able to cope with self-loops, which are known to be prominent in real GRNs.

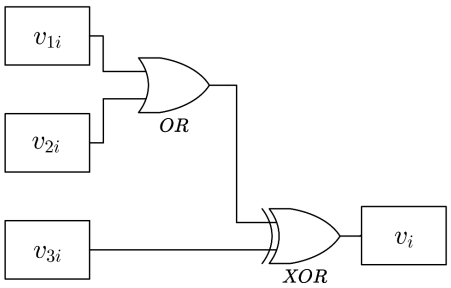
In the present paper we address some open questions raised by [23] and improve the previous approach in several different directions. First, a distinct encoding scheme is applied, based on integer representation, in order to reduce the length of GAs' individuals and therefore memory requirements. Second, a new, but still random initialization method is introduced, which aims at preventing the introduction of a high rate of false positives within the first generations of the algorithm. Third, GA operators of crossover and mutation are performed considering the network syntax. Finally, we evaluate candidate solutions by means of Tsallis entropy [26], which generalizes the Boltzmann entropy and has be shown to be a promising approach for inferring GRNs [18]. In contrast to the fitness function proposed by [23], the Tsallis entropy is able to cope with self-loops.

This paper is organized as follows. In the next section we will briefly describe the Boolean-based modeling framework and dynamics, the principles of Tsallis entropy, as well as the data set used in experiments. In the sequence, the proposed model is described. Finally, in Sections 5 and 6 we present and discuss the results of our work and point out future research directions.

2. METHODS

2.1 Random Boolean Networks

Random Boolean networks (RBNs) are one of the most well-known discrete modeling frameworks for GRNs. Albeit extremely simple, they are efficient in extracting meaningful biological information when the interest lies in a qualitative investigation of the regulatory interactions within a gene network [14]. A RBN is a directed graph $G(V, F)$ defined by a set of nodes $V = \{v_1, v_2, \ldots, v_N\}$ and a set of Boolean transitional functions $F = \{f_1, f_2, \ldots, f_N\}$. Each node v_i, $i = 1, \ldots, N$, is a Boolean device that stands for the state of variable (gene) i: in GRNs context, $v_i = 1$ denotes that gene i is expressed, while $v_i = 0$ means that it is not expressed The network state at time t is thus given by a N-dimensional vector $s(t) = [v_1(t), \ldots, v_N(t)]$. Since each node is a Boolean device, the system has a finite state space of size 2^N.



(a)

(t)			(t+1)
v_{1i}	v_{2i}	v_{3i}	v_i
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

(b)

Figure 1: An example of a Boolean function for node v_i, with predictors v_{1i}, v_{2i} and v_{3i} ($K_i = 3$), represented as a (a) logical circuit and (b) a state transition table. The state of v_i at time step $t + 1$ is obtained by applying the Boolean function to its predictors' state at the previous step, i.e., $v_i(t+1) = f_i(v_{ki}(t))$. For instance, this example indicates that when the predictors v_{1i}, v_{2i} and v_{3i} equal {0, 1, 1}, v_i will assume value 0. Adapted from [17].

Each node v_i has its value determined by a Boolean function $f_i \in F$, which represents the rules of regulatory interactions between nodes, and K_i specific inputs, denoting its regulatory factors or predictors. The function f_i is a logical circuit that, given the state of gene v_i's predictors, $v_{ki}(t)$ with $k = 1, \ldots, K_i$, at time t, generates the network states $s(t + 1)$ by mapping $v_i(t + 1) = f_i(v_{ki}(t))$. An example is given in Figure 1. Thus, being K_i the number of predictors of a given node, which can be either at state 1 or 0, the number of possible states for the set of K_i predictors is 2^{K_i}. Furthermore, for each of these combinations, the output of gene v_i defined by its Boolean function f_i must be either 1 or 0. Therefore, the total number of Boolean functions over K_i predictors is $2^{2^{K_i}}$. When $K_i = 2$, some of these functions are well-known (AND, OR, XOR, NAND, etc.), but in the general case functions have no obvious semantics.

The groundbreaker suggestion of studying the behavior of gene regulatory systems by means of networks of Boolean functions was introduced by Kauffman in [13]. Kauffman's experiments suggested that large randomly connected feedback networks of binary nodes behave with stability comparable to that in living organisms. The author asserts that "*It seems unlikely that Nature has made no use of such probable and reliable systems, both to initiate evolution and protect its progeny*". Since Kauffman's seminal work, most of research on Boolean networks has focused on unraveling the structure of GRNs from gene expression data.

[1]http://www.cs.waikato.ac.nz/ml/weka/

[2]Identify a network consistent with the observations in the given gene expression profile or determine if this network exists at all.

2.2 Inference by Tsallis entropy

In the context of information theory, Shannon's entropy [24] was considered a suitable measure for GRNs inference from expression data [15]. In general, when the entropy is adopted in the inference process, the conditional dependence of a target gene given its potential predictors and their expression profiles is performed and as a result, the mean conditional entropy is applied as a criterion function [19], i.e., to evaluate the subset of predictors and their suitability to predict a target gene. This process has been recognized as an appropriated statistical tool to infer direct interactions between genes [6]. However, the inference depends on the accuracy of the information available and the appropriateness of its use.

Moreover, the conditional entropy is commonly used to estimate the relationships among a group of multivariate predictors genes and a given target gene, i.e., N-to-1. The criterion functions based on information theory (entropy and mutual information) are also frequently applied in GRNs inference methods. However, these criterion functions are used to detect relationships of both 1-to-1, replacing the Pearson correlation, and N-to-1 relationships [15, 19, 22], in which the uniformity of the conditional probability distributions of a target gene, given its predictors, is considered as a whole.

On the other hand, C. Tsallis proposed a new entropy form in 1988, which became known as generalized Tsallis entropy or just Tsallis entropy, which is defined as follows:

$$H_q(X) = k\frac{(1-\sum_{x\in X} P(x)^q)}{q-1}, \quad (1)$$

in which k is a positive constant (which defines the size and scale), x is a possible configuration of the random variable X, $P(x)$ is the probability of x and $q \in \mathbb{R}$ is the entropic parameter.

The entropic parameter q characterizes the degree of nonextensivity of the system, which in the limit $q \to 1$ recovers the Shannon entropy. Therefore, the entropic form of H_q is not additive for any $q \neq 1$, and the connection between the entropic parameter q and the nonextensivity of the entropy is given by the rule [26]:

$$H_q(A+B) = H_q(A)+H_q(B)+(1-q)\times H_q(A)\times H_q(B), \quad (2)$$

in which A e B are two independent systems, i.e., $P(A,B) = P(A)\times P(B)$. From Equation 2 it was generated the expression "nonextensive entropy". Some properties can be observed in this equation such as nonnegativity for $H_q \geq 0$, superextensivity (superadditivity) for $q < 1$, extensivity (additivity) for $q = 1$ and subextensivity (subadditivity) for $q > 1$.

This new functional form of entropy allows the generalization of Boltzmann's statistical mechanics, which has been successful in presenting the properties of the statistical physics theory [27]. Besides that, the Tsallis entropy has emerged in recent years as a generalization of the Shannon entropy, because of their applications [12] and its theoretical foundation [1]. Its use becomes important in systems with long-range interactions, which cause long-range correlations, a particular feature of the nonextensive systems. In order to investigate the possibility of non-extensiveness of GRNs, and hence its interpretation in this context, it was proposed a new criterion function for the inference of GRNs based on the generalized Tsallis entropy [18], which produced better results compared to the Shannon entropy. Therefore, in order to infer the relationships among genes from its temporal expression profiles we adopt the criterion function proposed in [18], which is defined as follows:

$$S_q(v_i \mid g) = \frac{\alpha(m-n)}{\alpha m+d}S_q(v_i) + \sum_{g=1}^{n}\frac{r_g+\alpha}{\alpha m+d}\frac{1-\sum_{v_i}P(v_i|g)^q}{q-1}, \quad (3)$$

where $\alpha \geqslant 0$ is the penalty weight, m is the number of possible instances of the gene group g (predictors), n is the number of observed instances, d is the total number of samples, r_g is the number of each observed instance of g and $q \in \mathbb{R}$ is the entropic parameter of the Tsallis entropy.

2.3 Artificial gene networks

Despite the increasing availability of large-scale gene expression patterns, the reverse engineering of GRNs still suffers from an important limitation: the difficulty to evaluate results due to the restricted knowledge about the biological systems that generated these data sets. Therefore, the use of artificial networks and simulated expression signals is a common practice to assess algorithms performance. In the present paper, we resort to an Artificial Gene Network (AGN) validation and simulation model[3] [16, 17] to compose an artificial set of 100-node networks adopting the Boolean network approach. In order to verify the sensibility of the method on the topology class, we generate AGNs with the uniformly-random Erdös-Rényi (ER, [8]) and the Barabási-Albert (BA,[5]) models. The latter is currently known to be one of the most similar models to real gene networks [3].

Following the upper limit of stability for Boolean networks discussed in [13], we set the upper bound of nodes' average connectivity to $\langle k\rangle = 3$. Also, we consider two distinct approaches for the Boolean modeling: a deterministic (RBN) and a probabilistic (PBN) one. While the first approach considers a single Boolean function per gene to generate network dynamics, the former relax the deterministic rigidity by allowing each gene to have more than one Boolean function, each of which have a particular usage probability [25]. This probabilistic class of Boolean model offers a more flexible and powerful modeling framework at the cost of a greater difficulty of inference.

For each possible configuration of network topology, i.e., ER or BA, and network class, i.e., deterministic or probabilistic, we generated a random 100-node network and simulated 10 temporal expression signals of length 30, each one starting from a randomly chosen initial state [18]. The dynamics of the AGN is obtained by applying the Boolean transition functions to the network's initial state. Furthermore, we concatenated these signals generating a single time series of size 300, which is used for network inference.

3. PROPOSED MODEL

3.1 Representation

Following the approach described in [23], we model the GRNs as RBNs: genes are Boolean devices whose expression is regulated by a Boolean function and a set of predictors. However, we are not interested in recovering the whole set of transitional functions, but solely the network topology.

[3]http://code.google.com/p/jagn/

Thus, each individual of the GA codifies the network wiring of a candidate solution. An example is given in Figure 2. Figure 2(a) depicts a 5-node network, in which each node is regulated by two predictors at most. Node 1's predictor is node 5, node 2's predictor are node 4 and 5, and so on. The corresponding representation of this network as a GA individual is given in Figure 2(b).

A GA individual is codified as an integer string containing the full network wiring specification. The string length is given by $N \times K_{max}$ digits, in which N is the number of nodes in the network and K_{max} is an user-configurable upper bound limit for the cardinality of the nodes' predictor set. It is important to stress that the K_{max} parameter is independent of the average connectivity ($\langle k \rangle$) used for AGNs generation (see Section 2.3): it is applied during network inference to standardize individuals' size and restrict the search space, and may assume a value either equal, lower or higher than the latter. Each digit of the integer string contains either a zero or a non-zero value: while a non-zero value refers to the unique ID of a node's predictor, a zero value is used to allow a cardinality lower than K_{max}, i.e., $K_i < K_{max}$. Note that as node 5 has an empty predictor set (Figure 2(a)), both digits corresponding to its predictors assume a zero value (N5 in Figure 2(b)).

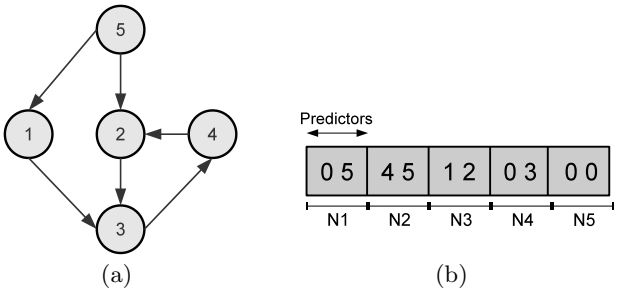


Figure 2: A 5-node network (a) topology and its corresponding (b) representation as a GA individual. In this example, nodes have a maximum in-degree of 2 ($K_{max} = 2$) and predictors are denoted by non-zero node IDs. Since we are interested in reconstructing the GRNs' topology, we do not encode the nodes' Boolean functions in candidate solutions.

3.2 Initialization

Each individual from the GA population is randomly initialized in such a way that every single node of the encoded network has one parent at most. This allows the heuristic search to start from a low-complex solution and gradually increase its complexity through GA operators, e.g., crossover and mutation, whenever this change results in a higher fitness value. Also, this strategy aims to reduce the frequency of false positive connections in the initial population.

3.3 Selection

The fittest individuals of a generation were selected based on the roulette wheel selection operator. The function to be optimized is implemented in terms of the Tsallis entropy, as defined in Equation 3, and aims to minimize the entropy of a single node regarding its predictors set. Elitism is applied in order to preserve the best found individuals in subsequent

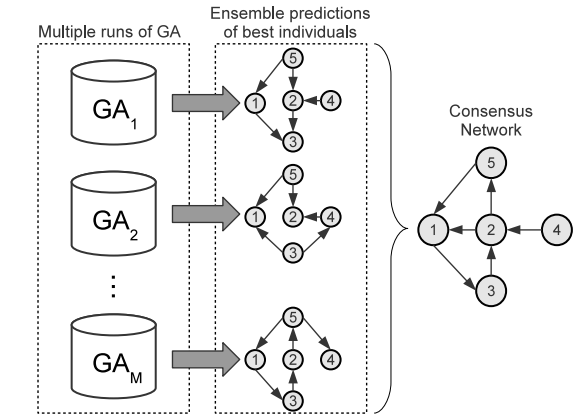


Figure 3: As multiple runs of GA may yield different results, wisdom of crowds is applied to construct a consensus network. The best individuals of the final generation perform a majority voting, generating an ensemble prediction for each simulation run. In the sequence, ensemble predictions are combined into a single consensus network.

generations and maintain the monotonic convergence of the algorithm.

3.4 Crossover and Mutation

In the context of exploring the solution space of RBNs with GAs, the operations of crossover and mutation are performed over the network wiring. As the number of nodes in the network is known and constant (it corresponds to the number of genes covered by the expression profile), the heuristic search must find the optimal connections between these nodes. This is achieved by varying the connections between the network's nodes and looking for the combination that maximizes the fitness function.

Therefore, in crossover operator, a pair of GA individuals exchange between themselves a set of connections placed between a randomly chosen point and the rightmost digit of the string (one-point crossover). Suppose we have two individuals of length 10 comprising a network of $N = 5$ nodes and $K_{max} = 2$, similar to the representation shown in Figure 2(b): `0545120300` and `2534102340`. If the random choice of the operator is to start the crossover on point 3, all connections regarding nodes 3 to N will be exchanged between the pair of mates to generate the offspring, which in this example will be `0545122340` and `2534100300`.

With respect to the mutation operator, which aims to promote diversity, the offspring may suffer eventual changes in their genetic material. In short, the network topology is varied by changing each digit of the integer string to a new random value with a small probability (P_{mut}). This operator may either remove (replace a non-zero digit by a zero digit) a node's connection, decreasing network complexity, or simply change (make a random swap between digits) a node's predictor. In order to allow some GA individuals to have its network complexity decreased and hence to explore in a controlled fashion sparse topologies within the search space, which are known to be GRNs' representative [11], the mutation operator performs a removal change in 10% of the mutations.

3.5 Wisdom of crowds

As genetic algorithms are a stochastic optimization approach [10], multiple runs of the algorithm may yield distinct results, such that an important question arises: how to compose the algorithm's final answer? One possible approach, applied in the present paper, is to combine the information carried by the best individuals in the last generation of each run [21]. Thus, we let the best individuals of the final generation to perform a majority voting on the network structure, generating an ensemble prediction. In the sequence, all ensembles are combined into a final consensus network, as depicted in Figure 3. This strategy has the benefit of improving results accuracy in the final network structure when compared to individual predictions [21].

4. EXPERIMENTS

In order to test the algorithm's sensibility to different network topologies, we run our method with networks based on the uniformly-random Erdös-Rényi (ER) and the Barabási-Albert (BA) models and on both deterministic (RBN) and probabilistic (PBN) classes. The simulations were set up by varying K_{max} between 2 and 3. These values were chosen based on the knowledge that biological gene networks are sparsely connected [11]. We performed 30 simulations for each network in our test set and combine their results in ensemble solutions, which in turn are used to build a final consensus network, as explained in Section 3.5. Also, we apply a reinforcement step to strengthen those interactions in the final consensus network whose inverse relationships are also present in the model with a lower confidence level (the ratio used is 0.5). We justify this operation with the fact that sometimes the reverse engineering algorithm is able to detect a significant correlation between two genes but the data or the score function are not fine enough to specify the sense of this influence.

The GA parameters were configured according to Table 1. We evolved a population of 30 individuals over 800 generations, applying one-point crossover and an elitist selection with elite size of four individuals. Mutation operator was applied with probability $P_{mut} = 0.001$ and edge removal operations were performed in 10% of cases. In what concerns the parameters from Tsallis entropy, defined in Equation 3, we used $\alpha = 1$ and $q = 2.5$. The choice of the q value is due to the good reconstruction accuracy attached to this configuration in [18].

5. RESULTS

The main results in terms of the ROC curve and the area under the curve (AUC score) are summarized in Table 2 and Figure 4. Table 2 shows the mean and standard deviation over 30 runs for the AUC score of ensemble predictions drawn from the final generation of each single simulation. Complementarily, the ROC curves for the final consensus network, i.e., the one assembled from the set of ensemble predictions, are plotted by varying the threshold used to create the consensus (Figure 4). Figures 4(a) to 4(d) refer to simulations for the ER model, while Figures 4(e) to 4(h) depict the results for simulations for the BA model. Furthermore, graphs on the right half concern the probabilistic Boolean model (PBN), while graphs on the left half are related to the deterministic one (RBN).

Observing the ROC curves in Figure 4, one can conclude that the proposed method has performed well relative to a random classifier. These graphs also bare the benefits of combining ensemble predictions into a single consensus network by means of wisdom of crowds: the AUC scores of the final consensus matrices are higher than the average AUC score for ensemble predictions in every case tested. Regarding the topology class, a comparison of results through 95% confidence intervals suggest there is no significant difference in the method performance in terms of the AUC score when comparing the ER and BA models. On average, the proposed method has achieved very similar scores for both topologies, 0.649 for ER against 0.647 for BA simulations, which let us conclude that the method's success does not depend on the network topology. Moreover, in general, simulations run with the probabilistic model (PBN, right half of Figure 4) have performed slightly poorer than those involving the deterministic ones (RBN, left half of Figure 4), but the difference is also not significantly different at the 0.05 significance level. This corroborates the idea that the reverse engineering of probabilistic models is harder due to the larger set of parameters to be inferred.

In what concerns the effects of the K_{max} parameter, except for simulations with the probabilistic ER model (Figures 4(c) and 4(d)), the use of $K_{max} = 2$ has yield better AUC scores. The improvement in performance was statistically significant ($p < 0.05$) and greater for networks based on the BA model, especially for the one with probabilistic nature. Table 3 shows a distinct results analysis, comparing the AGNs and the inferred consensus networks in terms of a similarity measure. This measure is obtained by combining both true positive rate (TPR, sensitivity) and true negative rate (TNR, specificity), as detailed in the following equations:

$$TPR = \frac{TP}{TP + FN} \qquad TNR = \frac{TN}{TN + FP} \tag{4}$$

$$similarity(N_1, N_2) = \sqrt{TPR \times TNR} \tag{5}$$

The analysis of results by the similarity measure confirms the better performance of $K_{max} = 2$ over $K_{max} = 3$. This indicates that high K_{max} values may degrade the method's performance, which is probably due to the inclusion of more

Table 1: Parameters used in simulations run.

Parameter	Description	Value
G_{max}	Number of generations	800
P_{size}	Population size	30
E	Elite size	4
P_{mut}	Mutation probability	0.001
P_{cross}	Crossover probability	1
K_{max}	Maximum predictors for node	{2, 3}

Table 2: Means and deviations of ensemble predictions' AUC scores over 30 runs of the GA.

Model	RBN				PBN			
	$K_{max} = 2$		$K_{max} = 3$		$K_{max} = 2$		$K_{max} = 3$	
	avg	std	avg	std	avg	std	avg	std
ER	.520	.0097	.524	.0080	.513	.0087	.524	.0114
BA	.519	.0085	.525	.0082	.515	.0078	.518	.0106

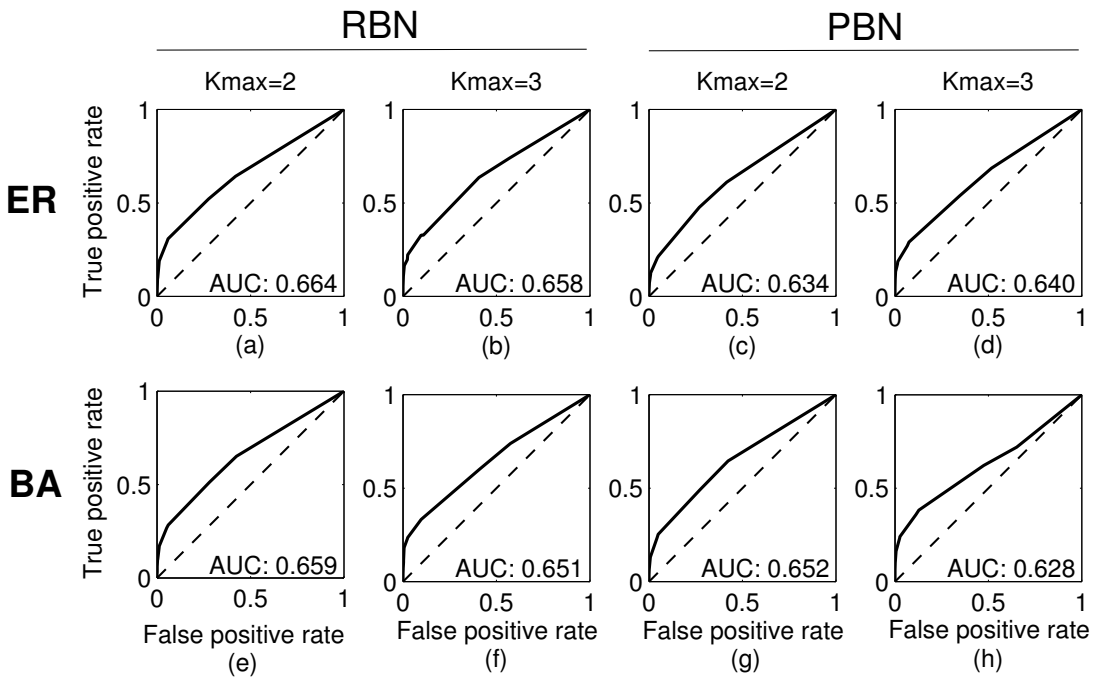


Figure 4: Results in terms of ROC curves and AUC scores for the different combinations of network topology, class and K_{max} value. Graphs on the top line (a to d) show the results for simulations with the ER model, while the graphs on the bottom line (e to h) depict the results for simulations with the BA model. Results suggest that the proposed method is not sensitive to network topology and that a more accurate inference is achieved with $K_{max} = 2$. Furthermore, the average AUC score for the probabilistic models is smaller than the average value found for the deterministic ones, which agrees with the common sense that probabilistic networks are of harder inference due to their more complex nature.

Table 3: Similarity between AGNs and inferred consensus networks for $q = 2.5$.

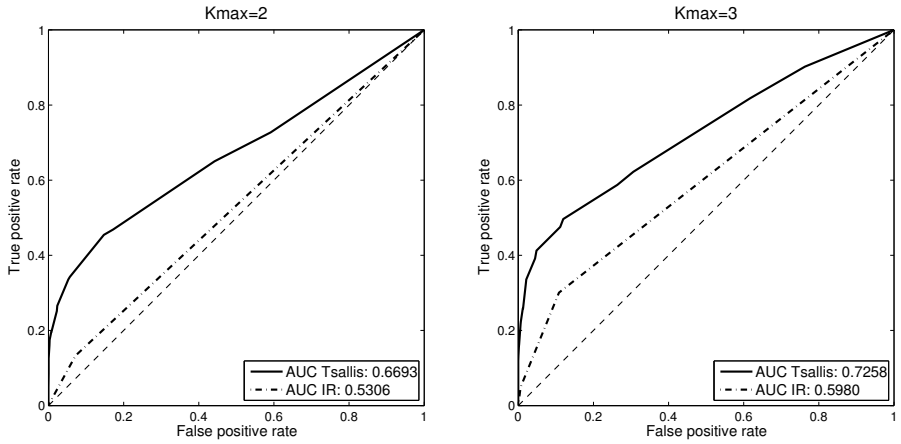
Model	RBN		PBN	
	$K_{max} = 2$	$K_{max} = 3$	$K_{max} = 2$	$K_{max} = 3$
ER	.6107	.5639	.5977	.5755
BA	.6123	.5609	.6098	.5005

false positive connections. Once again, the improvement was specially notable for the probabilistic BA model. In contrast, the impact over performance was modest for the probabilistic ER model. This finding agrees with the analysis by AUC score in the sense that the effect of K_{max} value is not very significant for the case of the probabilistic ER model. Moreover, the similarity values on Table 3 are close to results reported in [18] for ER and BA network models with average connectivity $\langle k \rangle = 3$.

In order to compare the reverse engineering method proposed in the current work with the solution discussed in [23], we followed the procedure described in Section 2.3 to generate an artificial 50-node probabilistic network and simulate its expression signal. Since previous results have shown that our method's performance is not dependent on the network topology, we compare solely the BA model, as this is currently known to be the closest topology to real gene regulatory networks [3]. At this point, we stress the main changes of the present work when compared to the approach in [23], namely the initialization procedure and the fitness function. These changes have the goal of smoothing false positive rates. For the opposite method, the fitness function was defined according to the Equation 4 in [23]. Regarding the GA parameters, simulations were set up following values on Table 1.

Table 4 shows the analysis of results by means of the similarity measure between inferred consensus networks and target GRNs, as defined in Equation 5. We identify our method by *Tsallis entropy*, while the approach suggested in [23] is referred to as *Inconsistency Ratio* (IR). The superior performance of the proposed method is clear: for a maximum connectivity of two ($K_{max} = 2$), the similarity measure is up to 50% higher than the results obtained with the IR fitness function – 0.5451, against 0.3512 for the latter. This is explained by the good balance between false positive and true positive connections recovered by our method when a maximum set of two predictors per gene is allowed. An improvement was also perceived for $K_{max} = 3$, although not in such a large scale: the similarity measure for networks obtained with the Tsallis entropy is equal to 0.5962, while the use of the IR fitness function has yield a consensus network with similarity rate equal to 0.5178.

An analysis by AUC scores confirms the higher robustness of the proposed method based on Tsallis entropy in relation to the minimization of an inconsistency ratio. For both $K_{max} = 2$ and $K_{max} = 3$, our method outperforms the approach introduced by [23], as shown in Figure 5. Ad-



(a) (b)

Figure 5: Network inference accuracy in terms of ROC curve and AUC scores for the two approaches compared: Tsallis entropy (Tsallis) and Inconsistency Ratio (IR). The left graph refers to the results obtained with $K_{max} = 2$, while the right graph reports the results for $K_{max} = 3$. The improvement achieved with the Tsallis criterion function is clear: for both values of K_{max} it has outperformed the inference by the minimization of an Inconsistency Ratio.

Table 4: Comparison between network inference by means of Tsallis entropy and through an inconsistency ratio fitness function for a 50-node AGN.

Fitness Function	Similarity	
	$K_{max} = 2$	$K_{max} = 3$
Tsallis entropy	.5451	.5962
Inconsistency Ratio	.3512	.5178

ditionally, both methods have achieved a better score for networks inferred with $K_{max} = 3$ in contrast to simulations with $K_{max} = 2$. This is explained by the fact that a higher maximum connectivity increases the chance of recovering the whole set of true positive connections. However, this also makes the algorithm more vulnerable to false positive interactions, which in turn increases the need for more powerful criterion functions and GA operators. This balance between power and robustness has been better achieved by the proposed method: the evaluation of candidate solutions by Tsallis entropy not only recovers more interactions from the target network, but also causes a significant reduction of false positive rates.

6. CONCLUSION

In the present paper we introduced the application of the Tsallis entropy as a fitness function to reverse engineering gene networks by means of genetic algorithms. Although the idea of using entropy or mutual information to infer gene networks is not new, the inference of gene networks by means of Tsallis entropy was recently proposed in [18]. Authors have defined a new criterion function, which aimed at extending the Boltzmann entropy and allowing the recovery of networks whose genes do not follow linear relationships and short-range interactions. Moreover, while recent papers have covered the topic of applying genetic algorithms to uncover gene networks [4, 7, 20, 23], they all have issues that limit their application: bad scalability, need for many parameters tuning, high vulnerability to false positive connections and not so accurate results. Thus, improvements in the area are still necessary.

The proposed solution combines the robustness of Tsallis entropy, which is known for reducing false positive rates [18], with the benefits of ensemble predictions provided by the wisdom of crowds, which have been previously shown to yield more accurate results in the context of gene network inference [21]. Furthermore, as gene networks are known to have a sparse topology [11], GA operators are implemented in order to prioritize the exploration of sparser networks within the search space. For results assessment, we have generated deterministic and probabilistic artificial Boolean networks following two distinct models: uniformly-random Erdös-Rényi (ER) and the Barabási-Albert (BA). Simulations were run by varying the number of maximum allowed predictors per node, given by parameter K_{max}, for each network within the test set.

Results have shown that the performance of the proposed method does not hold any dependence on network topology. No significant difference in terms of the AUC score was noticed for simulations with the ER model in contrast to the BA model, as shown in Figure 4. Furthermore, simulations with $K_{max} = 2$, which allow a set of two predictors per gene at most, have yield more accurate results. This is due by the fact that when $K_{max} = 3$, as the set of predictors is larger, more false positives may be contained within the inferred networks, which in turn degrades reconstruction accuracy. In what concerns network classes, deterministic networks were more accurately inferred than probabilistic ones, due to the nonlinearity and higher complexity attached to the latter. When compared to similar approaches, such as the inference by minimization of an inconsistency ratio as described in [23], the proposed method has been shown to be more robust and have a better balance between true positive and false positive rates, as depicted in Figure 5.

Despite the satisfactory performance, there is still space and need for improvements. For future works, we believe

that a promising direction for enhancement is to review and improve implementation details of genetic algorithm, specially regarding crossover and mutation operators. The goal is to bring them even closer to network syntax and thus promote better convergence. Additionally, there are still some parameters to be tested and optimized, such as the percentage of removal operations in mutations. As this has a direct influence over the rate of false positive connections, it may improve results when tuned. Moreover, we would like to investigate how the introduction of partial consensus networks within the generations would help to improve convergence to accurate networks. We are confident that these refinements would make the proposed solution even more robust and useful for knowledge discovery from gene expression patterns.

7. ACKNOWLEDGMENTS

We thank the reviewers for their comments and suggestions and the CNPq for its support to the authors.

8. REFERENCES

[1] S. Abe. Tsallis entropy: how unique? *Continuum Mechanics and Thermodynamics*, 16(3):237–244, 2004.

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

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

[4] S. Ando and H. Iba. Inference of gene regulatory model by genetic algorithms. In *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, pages 712–719. IEEE CS, 2001.

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

[6] C. Charbonnier, J. Chiquet, and C. Ambroise. Weighted-LASSO for Structured Network Inference from Time Course Data. *Statistical Applications in Genetics and Molecular Biology*, 9(1):15, 2010.

[7] C. Davidson. Identifying gene regulatory networks using evolutionary algorithms. *J. Comput. Small Coll.*, 25:231–237, May 2010.

[8] P. Erdös and A. Rényi. On random graphs, I. *Publicationes Mathematicae*, 6:290–297, 1959.

[9] N. Friedman, M. Liniar, I. Nachman, and D. Peer. Using Bayesian networks to analyse expression data. In *Journal of Computational Biology*, volume 7, pages 601–620. Mary Ann Liebert, Inc., 2000.

[10] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1989.

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

[12] S. Issue. Nonextensive statistical mechanics: new trends, new perspectives. *Europhysics News*, 36(6):185–231, 2005.

[13] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J Theor Biol*, 22(3):437–467, March 1969.

[14] H. Lähdesmäki, I. Shmulevich, and O. Yli-Harja. On learning gene regulatory networks under the Boolean network model. In *Machine Learning*, volume 52, pages 147–167. Kluwer Academic Publishers, 2003.

[15] S. Liang, S. Fuhrman, and R. Somogyi. REVEAL, A Generak Reverse Engineering Algorithm for Inference of Genetic Network Architectures. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 3, pages 18–29, 1998.

[16] F. M. Lopes, R. M. Cesar-Jr, and L. da F. Costa. AGN Simulation and Validation model. In *Proceedings of Advances in Bioinformatics and Computational Biology*, volume 5167 of Lecture Notes in Bioinformatics, pages 169–173. Springer-Verlag Berlin, 2008.

[17] F. M. Lopes, R. M. Cesar-Jr, and L. da F. Costa. Gene Expression Complex Networks: Synthesis, Identification, and Analysis. *Journal of Computational Biology*, 18(10):1535–1367, 2011.

[18] F. M. Lopes, E. de Oliveira, and R. Cesar. Inference of gene regulatory networks from time series by Tsallis entropy. *BMC Systems Biology*, 5(1):61, 2011.

[19] F. M. Lopes, D. C. Martins-Jr, and R. M. Cesar-Jr. Feature selection environment for genomic applications. *BMC Bioinformatics*, 9(1):451, October 2008.

[20] M. Mamakou, C. Sirakoulis, I. Andreadis, and I. Karafyllidis. Adaptive reverse engineering of gene regulatory networks using genetic algorithms. In *Computer as a Tool, 2005. EUROCON 2005.The International Conference on*, volume 1, pages 401–404, nov. 2005.

[21] D. Marbach, C. Mattiussi, and D. Floreano. Combining Multiple Results of a Reverse Engineering Algorithm: Application to the DREAM Five Gene Network Challenge. *Ann N Y Acad Sci*, 1158:102–113, 2009.

[22] A. Margolin, K. N. Basso, C. Wiggins, G. Stolovitzky, R. Favera, and A. Califano. ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7(Suppl 1):S7, 2006.

[23] M. R. Mendoza and A. L. C. Bazzan. Evolving Random Boolean Networks with Genetic Algorithms for Regulatory Networks Reconstruction. In *Proceedings of the 13th annual Conference on Genetic and Evolutionary Computation*, GECCO '11, pages 291–298, New York, NY, USA, July 2011. ACM.

[24] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.

[25] 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.

[26] C. Tsallis. I. Nonextensive Statistical Mechanics and Thermodynamics: Historical Background and Present Status. *Lecture Notes in Physics*, 560(3):3–98, 2001.

[27] C. Tsallis. What should a statistical mechanics satisfy to reflect nature? *Physica D: Nonlinear Phenomena*, 193(1-4):3–34, 2004.