Hub Gene Selection Methods for the Reconstruction of Transcription Networks

José Miguel Hernández-Lobato¹ and Tjeerd M.H. Dijkstra²

¹ Computer Science Department, Universidad Autónoma de Madrid,
Francisco Tomás y Valiente, 11, 28049, Madrid, Spain

`josemiguel.hernandez@uam.es`

² Institute for Computing and Information Sciences,
Radboud University Nijmegen, The Netherlands

`t.dijkstra@science.ru.nl`

Abstract. Transcription control networks have a scale-free topological structure: While most genes are involved in a reduced number of links, a few hubs or key regulators are connected to a significantly large number of nodes. Several methods have been developed for the reconstruction of these networks from gene expression data, e.g. ARACNE. However, few of them take into account the scale-free structure of transcription networks. In this paper, we focus on the hubs that commonly appear in scale-free networks. First, three feature selection methods are proposed for the identification of those genes that are likely to be hubs and second, we introduce an improvement in ARACNE so that this technique can take into account the list of hub genes generated by the feature selection methods. Experiments with synthetic gene expression data validate the accuracy of the feature selection methods in the task of identifying hub genes. When ARACNE is combined with the output of these methods, we achieve up to a 62% improvement in performance over the original reconstruction algorithm. Finally, the best method for identifying hub genes is validated on a set of expression profiles from yeast.

Keywords: Transcription network, ARACNE, Automatic relevance determination, Group Lasso, Maximum relevance minimum redundancy, Scale-free, Hub.

1 Introduction

High-throughput molecular technologies like DNA microarrays allow researchers to simultaneously measure the concentration of thousands of known molecules (e.g. mRNAs) in a cellular population. This technological breakthrough has encouraged a systems biology approach to the analysis of complex cellular processes [1,2]. The new paradigm focuses on the interactions between molecules, usually represented by a network, and how these interactions give rise to cellular behavior. In particular, functional properties are not in the molecules themselves but they appear as a result of their coordinated interactions. A challenging task is

then to reconstruct the network of molecule interactions from high-throughput experimental data [3,4,5].

Transcription networks are a particular class of interaction networks in which each node represents a different gene and each link corresponds to an interaction between two genes at the transcription level. The network captures the regulatory role of certain genes whose final products, e.g. transcription factors, govern the transcription rate of other genes. Transcription control networks display two typical characteristics. First, they have a sparse connectivity matrix [6] and second, they have a scale-free topological organization [7,4]. In particular, while most genes are involved in only a reduced number of interactions, a few hubs or key regulators are connected to a significantly large number of nodes. These features can be observed in the transcription network displayed in Fig. 1.

Under the assumption that mRNA concentration is predictive of regulatory molecule activity, several machine learning methods have been proposed for the reconstruction of transcription networks from steady-state gene expression data. Some examples are given by Bayesian network methods [8], mutual information approaches [4,5] and linear regression techniques [9,10]. While most of these methods enforce sparsity in the resulting interaction map, few of them actually exploit the scale-free topological structure of transcription control networks. An exception is the method described in [11] which is based on a scale-free prior within a framework of Gaussian graphical models. However, this latter method has a very high computational cost and is not feasible for large-scale problems. As a more efficient approach in terms of computational time, we concentrate on the hubs that commonly appear in transcription networks.

This paper is based on the assumption that reverse engineering methods can be significantly improved by predicting beforehand those genes that are more likely to be hubs or key regulators in the transcription network. For this task, we introduce a multiple linear regression model which describes the interactions between genes and transcriptional regulators. The model includes a coefficient matrix which is expected to be columnwise sparse. A set of hub genes can then be selected by identifying those columns of the regression matrix which have many non-zero entries. In practice, this is equivalent to selecting those features that are more relevant for solving the multiple regression problem. Three feature selection methods are proposed for this task: The group Lasso regularization method [12], the automatic relevance determination (ARD) approach [13] and the maximum relevance minimum redundancy (MRMR) criterion [14]. The performance of the different methods for identifying hub genes is evaluated in a series of experiments with simulated gene expression data. The program SynTREN is used in the data generation process [15]. These experiments indicate that the best performance is obtained by the hub gene selection method based on the ARD technique.

In a second step, we show how the list of hub genes generated by the feature selection methods can be used to significantly improve the results of ARACNE, a state-of-the-art technique for reverse engineering transcription networks [4,16]. Another series of experiments with simulated gene expression data indicate that a combination of ARACNE with the ARD feature selection method yields a 62%

Fig. 1. Approximation of the transcriptional regulatory network of *Escherichia coli*. Nodes represent genes and edges indicate a direct transcriptional interaction between two genes. Diamond shaped nodes correspond to those genes selected by ARD, most of them are key regulators or hubs in the network. Light-gray nodes are those genes selected by the group Lasso approach, most of them are connected to the same hub node and their expression levels are therefore highly correlated. The visualization of the network was implemented using the software Cytoscape [18].

improvement in performance over the results of the original algorithm. Finally, the ARD method for hub gene identification is run on a set of expression profiles from *Saccharomyces cerevisiae* collected under different experimental conditions [17]. This dataset contains 247 expression measurements for a total of 5520 genes. An analysis of the the top ten genes identified by the ARD technique reveals a large number of global transcriptional regulators.

2 A Linear Model of Transcription Control

First of all, we describe a linear regression model for steady-state gene expression data. Transcription control in biological systems is a dynamic process typically characterized by a set of differential equations. Michaelis Menten interaction kinetics and the Hill equation [19,20] model the rate of production of a transcript X by the enzyme RNA polymerase P when activator A is required for transcription:

$$\frac{d[X]}{dt} = V_m \cdot \frac{[A]^\alpha}{[A]^\alpha + K_A} \cdot \frac{[P]}{[P] + K_P} - \delta[X], \quad (1)$$

where δ is the degradation rate of transcript X , V_m is the maximum rate of synthesis, K_P and K_A are activation thresholds, α is the Hill coefficient for

cooperative binding and $[\cdot]$ stands for "concentration of". When competitive inhibition occurs between the enzyme RNA polymerase P and the repressor R , the rate of transcription is given by

$$\frac{d[X]}{dt} = V_m \cdot \frac{[P]}{[P] + K_P \left(1 + [R]^\beta / K_R\right)} - \delta[X], \quad (2)$$

where K_R is a repressor threshold and the parameter β is similar to α . When steady-state conditions hold and molecule concentrations are far from saturating, we obtain

$$[X] = \frac{V_m [A]^\alpha [P]}{\delta K_A K_P}, \quad [X] = \frac{V_m K_R [P]}{\delta K_P [R]^\beta}, \quad (3)$$

for the activation and repression cases, respectively. Taking logarithms, assuming that activation and repression are both possible at the same time and considering that mRNA concentration is predictive of final product concentration, we end up with a linear model of transcript log-concentration

$$\log[X] = \sum_{i=1}^m b_i \log[X_i] + \text{constant}, \quad (4)$$

where the b_i are real coefficients and the X_i are the transcripts of the activators, the repressors and the enzyme RNA polymerase. When X is a self-regulating gene, $\log[X]$ is included in the right-hand side of (4) with associated coefficient b_{m+1} . This autoregulatory term can be eliminated by algebraic simplification. For this, b_i is replaced by $b'_i = b_i / (1 - b_{m+1})$ where $i = 1, \dots, m$ and b_{m+1} is set to zero.

The previous linear model is easily extended to account for all the transcripts present in a biological system. Let $\mathbf{X}$ denote a $d \times n$ gene log-expression matrix where each row corresponds to a different gene and each column represents an independent sample obtained under steady-state conditions. Additionally, let us assume that the rows of $\mathbf{X}$ have been centered so that they have zero mean. Then, if the log-expression measurements are contaminated with additive Gaussian noise, (4) suggests that $\mathbf{X}$ should approximately satisfy

$$\mathbf{X} = \mathbf{B}\mathbf{X} + \sigma\mathbf{E}, \quad (5)$$

where $\mathbf{B}$ is a $d \times d$ matrix of regression coefficients that links each gene to its transcriptional regulators, σ is a positive constant that is determined by the level of noise in $\mathbf{X}$ and $\mathbf{E}$ is a $d \times n$ matrix whose elements are i.i.d. random variables that follow a standard Gaussian distribution. In this linear model, the diagonal elements of $\mathbf{B}$ are all zero since any autoregulatory term in (5) can be eliminated using the process described in the previous paragraph.

Note that the coefficient matrix $\mathbf{B}$ encodes the connectivity of the underlying transcription network. In particular, let b_{ij} be the element in the i -th row and j -th column of $\mathbf{B}$, then $b_{ij} \neq 0$ whenever gene j is a transcriptional regulator of

gene i and $b_{ij} = 0$ otherwise. Consequently, $\mathbf{B}$ is constrained by the topological structure of the network. Transcription networks are sparsely connected [6], with most genes having only a few connections. However, a few hubs or key regulators are connected to a significantly large number of genes [21]. Fig. 1 displays a sample transcription network with these characteristics. Consequently, we may expect $\mathbf{B}$ to be a sparse matrix whose rows typically have a few coefficients different from zero and these non-zero coefficients are often clustered on a few of the columns of $\mathbf{B}$, where each of these columns corresponds to a different hub gene or global regulator. We may say that $\mathbf{B}$ is a columnwise sparse matrix. Note that the column of $\mathbf{B}$ that corresponds to RNA polymerase is the one with more non-zero elements since this enzyme is involved in the production of all the transcripts in any biological system.

Given $\mathbf{X}$, the previous paragraph suggests a procedure for identifying those genes that are more likely to be hubs in the transcription network. The method consists in first, solving the multiple regression problem represented by (5) under the assumption that $\mathbf{B}$ is columnwise sparse and second, identifying the columns of this regression matrix with more non-zero coefficients. This is equivalent to selecting the rows of $\mathbf{X}$ that are more relevant for solving the multiple regression problem, i.e., we have to identify a reduced subset of genes whose expression pattern accounts for the expression pattern of all the other genes in the network.

3 Hub Gene Selection Methods

Let k be an approximation of the number of hub genes in the transcription network. We propose three linear methods for the identification of the k rows of $\mathbf{X}$ that are more relevant for solving the multiple regression task. The proposed techniques are based on extensions of standard sparse linear models [22,13] and standard feature selection methods [14] to the multiple regression case.

3.1 Automatic Relevance Determination

In this section, we extend the automatic relevance determination (ARD) framework [13] to enforce columnwise sparsity of matrix $\mathbf{B}$ in (5). The first step is to assign a Gaussian prior to the rows of $\mathbf{B}$. Let $\mathbf{b}_i = (b_i^1, \dots, b_i^{i-1}, b_i^{i+1}, \dots, b_i^d)$ be the i -th row in this matrix, then its prior distribution is

$$\mathcal{P}(\mathbf{b}_i|\mathbf{a}) = (2\pi)^{-(d-1)/2} \prod_{j \neq i} a_j^{1/2} \exp \left[-a_j (b_i^j)^2 / 2 \right], \quad (6)$$

where $\mathbf{a} = (a_1, \dots, a_d)$ is a d -dimensional vector of hyperparameters. The prior for the i -th column of $\mathbf{B}$ is then a zero-mean multivariate Gaussian distribution with independent components that have an inverse variance given by a_i . The ARD method works by optimizing the evidence of the resulting Bayesian model with respect to $\mathbf{a} = (a_1, \dots, a_d)$. During this process, some of the a_i take an infinite value and consequently, the resulting posterior for the corresponding

rows of $\mathbf{B}$ is a point mass at zero. However, a direct optimization of the evidence function under the ARD framework is very likely to get stuck into a sub-optimal solution. To avoid this problem, we propose a greedy optimization method in which all the a_i are initially set to infinite and only k of these parameters are optimized sequentially in a greedy manner.

The logarithm of the evidence of the resulting Bayesian model is

$$\mathcal{L}(\mathbf{a}) = -\frac{1}{2} \sum_{i=1}^d \log |\mathbf{C}_{-i}| - \frac{1}{2} \sum_{i=1}^d \mathbf{x}_i^t \mathbf{C}_{-i}^{-1} \mathbf{x}_i + \text{constant}, \quad (7)$$

where $\mathbf{x}_i$ represents the transpose of the i -th row of $\mathbf{X}$ and

$$\mathbf{C} = \sigma^2 \mathbf{I} + \sum_{i=1}^d a_i^{-1} \mathbf{x}_i \mathbf{x}_i^t, \quad \mathbf{C}_{-i} = \sigma^2 \mathbf{I} + \sum_{j \neq i}^d a_j^{-1} \mathbf{x}_j \mathbf{x}_j^t. \quad (8)$$

Following [23], we separate out the contribution of each a_j on the total log-evidence

$$\begin{aligned} \mathcal{L}(\mathbf{a}) &= \mathcal{L}(\mathbf{a}_{-j}) + \ell(a_j) \\ &= \mathcal{L}(\mathbf{a}_{-j}) + \sum_{i \neq j} \frac{1}{2} \left[\log a_j - \log(a_j + s_{ji}) + \frac{q_{ji}^2}{a_j + s_{ji}} \right], \end{aligned} \quad (9)$$

where $s_{ji} = \mathbf{x}_j^t \mathbf{C}_{-ij}^{-1} \mathbf{x}_j$ and $q_{ji} = \mathbf{x}_j^t \mathbf{C}_{-ij}^{-1} \mathbf{x}_i$. For the computation of s_{ji} and q_{ji} we use

$$s_{ji} = \frac{a_j S_{ji}}{a_j - S_{ji}}, \quad q_{ji} = \frac{a_j Q_{ji}}{a_j - S_{ji}}, \quad (10)$$

$$S_{ji} = D_{jj} + \frac{D_{ji}^2}{a_i - D_{ii}}, \quad Q_{ji} = \frac{a_i D_{ji}}{a_i - D_{ii}}, \quad (11)$$

where $S_{ji} = \mathbf{x}_j^t \mathbf{C}_{-i}^{-1} \mathbf{x}_j$, $Q_{ji} = \mathbf{x}_j^t \mathbf{C}_{-i}^{-1} \mathbf{x}_i$ and $D_{ji} = \mathbf{x}_j^t \mathbf{C}^{-1} \mathbf{x}_i$. Note that when $a_i = +\infty$, it is satisfied that $S_{ji} = D_{jj}$ and $Q_{ji} = D_{ji}$. Thus, it is only necessary to compute D_{ii} when $a_i < +\infty$ and with (10), (11) and matrix $\mathbf{C}^{-1}$ we can compute all the s_{ji} and q_{ji} very efficiently.

A greedy process allows us to select the k most relevant rows of $\mathbf{X}$. First, all the components of $\mathbf{a}$ are initialized to $+\infty$ and a copy of matrix $\mathbf{C}^{-1}$ is kept in memory with initial value $\sigma^{-2} \mathbf{I}$. For $j = 1, \dots, d$ such that $a_j = +\infty$, we compute the increment in the log-evidence that can be achieved by optimizing $\mathcal{L}(\mathbf{a})$ with respect to a_j when $\mathbf{a}_{-j} = (a_1, \dots, a_{j-1}, a_{j+1}, \dots, a_d)$ is kept fixed. The a_j that yields the highest increment is then updated so that $\mathcal{L}(\mathbf{a})$ is maximized. All the optimizations are performed using a combination of golden section search and successive parabolic interpolation. The main loop of the process is repeated until k components of $\mathbf{a}$ are finite. For a particular j , we compute D_{ji} for all i as the product $\mathbf{X} \mathbf{C}^{-1} \mathbf{x}_j$. Once an a_j becomes finite, we efficiently update $\mathbf{C}^{-1}$ using

$$\mathbf{C}_{\text{new}}^{-1} = \mathbf{C}_{\text{old}}^{-1} - \frac{\mathbf{C}_{\text{old}}^{-1} \mathbf{x}_j \mathbf{x}_j^t \mathbf{C}_{\text{old}}^{-1}}{a_j^{\text{new}} + \mathbf{x}_j^t \mathbf{C}_{\text{old}}^{-1} \mathbf{x}_j}. \quad (12)$$

Input: A gene log-expression matrix $\mathbf{X}$ and a positive integer k .
Output: A value for the regularization parameter M .

```

bottomM  $\leftarrow$  0.
topM  $\leftarrow$  100.
converged  $\leftarrow$  FALSE.
while not (converged)
    newM  $\leftarrow$  (bottomM + topM) / 2.
    numberOfNonZeroColumns  $\leftarrow$  groupLasso( $\mathbf{X}$ , newM).
    if (numberOfNonZeroColumns < k)
        bottomM  $\leftarrow$  newM.
    else if (numberOfNonZeroColumns > k)
        topM  $\leftarrow$  newM.
    else converged  $\leftarrow$  TRUE.
M  $\leftarrow$  newM.
return M.

```

Fig. 2. Pseudocode of the binary search process for finding M in the group Lasso method. The figure displays the steps that are followed to obtain a value for M in (13) so that the minimizer of this expression has exactly k non-zero columns. The function `groupLasso` returns the number of non-zero columns in the minimizer of (13) when $\mathbf{X}$ and M are equal to the arguments passed to this function.

To compute D_{ii} for all i such that $a_i < +\infty$, we keep in memory a matrix $\mathbf{F}$ whose components are $\mathbf{x}_i^t \mathbf{C}^{-1} \mathbf{x}_j$ for all a_i and a_j finite. When an a_j becomes finite, $\mathbf{F}$ is efficiently updated by using (12) and appending a new row and a new column to the matrix. The required D_{ii} values are then easily obtained from the main diagonal of $\mathbf{F}$. Once this greedy process is finished, the k hub genes selected by the ARD method are represented by the k components of $\mathbf{a}$ that are finite. The final computational cost of this technique is $\mathcal{O}(kd^2n)$.

3.2 Group Lasso

An extension of the Lasso regularization method [22] denoted group Lasso [12] is useful to obtain groupwise sparse linear models. Given (5), we can obtain a columnwise sparse estimate of the coefficient matrix $\mathbf{B}$ as the minimizer of

$$\| \mathbf{X} - \mathbf{B}\mathbf{X} \|_{\mathbf{F}}^2 \quad \text{subject to} \quad \sum_{i=1}^d \| \mathbf{b}_i \|_2 \leq M \quad \text{and} \quad \text{diag}(\mathbf{B}) = \mathbf{0}, \quad (13)$$

where $\| \cdot \|_{\mathbf{F}}$ and $\| \cdot \|_2$ stand for the Frobenius and ℓ_2 norms, respectively, $\mathbf{b}_i$ is the i -th column of matrix $\mathbf{B}$ and M is a positive regularization parameter. Since $\mathbf{B}$ is constrained by the sum of ℓ_2 norms, some of its non-relevant columns will shrink to zero during the optimization process [12]. In this regularization method, there is a direct relationship between M and the number of non-zero columns in the resulting estimate of $\mathbf{B}$. The larger M is, the more non-zero

columns in the coefficient matrix estimate. Consequently, to select the k most relevant rows of $\mathbf{X}$, we perform a binary search on M so that the corresponding minimizer of (13) has exactly k non-zero columns. This process is illustrated by the pseudocode displayed in Fig. 2. The optimization problem involved by (13) is efficiently solved using the gradient projection method described in [24]. The final computational cost of the group Lasso method is $\mathcal{O}(d^3)$. This is comparatively much more expensive than the cost of the ARD approach when $d \gg n$, which is often the case in practical applications.

3.3 Maximum Relevance Minimum Redundancy

The maximum relevance minimum redundancy (MRMR) criterion for feature selection [14] is generalized in this section to the particular case of the multiple regression problem displayed in (5). Let $\mathbf{x}_i = (x_{i1}, \dots, x_{in})$ be the i -th row of $\mathbf{X}$ and let $I(\mathbf{x}_i, \mathbf{x}_j)$ denote the empirical mutual information for genes i and j . Now, suppose that S is a set that contains those genes already selected by the method. The next gene to be selected is a gene j that is not yet in S and that maximizes

$$\frac{1}{d} \sum_{i=1}^d I(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{|S|} \sum_{i \in S} I(\mathbf{x}_i, \mathbf{x}_j). \quad (14)$$

The redundancy term is here the same as in standard MRMR. However, the relevance term is an average of the relevance terms for each of the d individual regressions in (5). For the computation of the empirical mutual information, $I(\mathbf{x}_i, \mathbf{x}_j)$, we assume that the marginal distribution of the pairs $\{x_{ik}, x_{jk}\}_{k=1}^n$ is bivariate Gaussian. In consequence, $I(\mathbf{x}_i, \mathbf{x}_j) = -0.5 \log(1 - \hat{\rho}_{ij}^2)$, where $\hat{\rho}_{ij}$ is the empirical correlation of the vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. The final computational cost of the MRMR method is $\mathcal{O}(d^2 n)$. Hence, MRMR is the most efficient of the three feature selection methods analyzed in this paper.

4 Evaluation of the Hub Gene Selection Methods

The performance of the ARD, group Lasso and MRMR methods for identifying hub genes is evaluated in a series of experiments with simulated steady-state gene expression data. The program SynTREN [15] is used to generate synthetic gene expression data from three different transcription control networks. Two of the networks are from the organism *Escherichia coli* and they have 423 and 1330 genes, respectively [25,26]. The last network is from *Saccharomyces cerevisiae* and has 690 genes [27]. 50 gene expression matrices with 250 measurements are generated from each network using the following configuration for SynTREN: All the noise levels are fixed to 0.3, transition functions are linear, the number of burn in-cycles is fixed to 1000, complex interactions are deactivated and the number of correlated externals is 0.

To reduce the negative impact of outliers, we map each row of each gene log-expression matrix to the standard Gaussian distribution. For this task, an approximation of the probability integral transform is used in combination with the standard Gaussian quantile function:

$$x_{ij}^{\text{map}} = \Phi^{-1} \left[F_n(x_{ij}) \frac{n}{n+1} \right] \quad \text{for } j = 1, \dots, n, \quad (15)$$

where x_{ij} is the j -th element in the i -th row of a $d \times n$ gene log-expression matrix, Φ^{-1} is the standard Gaussian quantile function, F_n is the empirical cumulative distribution of the elements in the i -th row of the log-expression matrix and the factor $n/(n+1)$ guarantees that the argument passed to Φ^{-1} is lower than one. The resulting data matrix is the input to the hub gene selection methods.

The ARD, group Lasso and MRMR methods are executed on each of the gene expression matrices generated by the program SynTREN (after applying the previous data preprocessing method). The parameter σ in the ARD approach is initialized to 1 and k , the number of hub genes that each method must select, is fixed to a reduced fraction of the total number of nodes in the underlying transcription network, e.g. 5%. Hub genes are by definition highly connected nodes in the network. Hence, as a measure of quality, we compute the average connectivity of the genes selected by each method. For comparative purposes, we also include in the analysis an extra method that only returns a list of k genes selected randomly. The highest possible average connectivities that a method can obtain are 17.27 and 37.31 for the small and large *E. coli* networks and 22.51 for the yeast network.

Table 1 summarizes the results of each feature selection method. The three proposed methods perform significantly better than the random approach, with ARD obtaining the best accuracy. For small networks, the results of MRMR and ARD are more or less similar. However, ARD is much better than MRMR when analyzing large transcription networks. Additionally, the Group Lasso method performs rather poorly when compared with the MRMR and ARD approaches. The reason for this poor performance is the propensity of the group Lasso method for selecting genes whose expression patterns are very correlated. By contrast, ARD and MRMR do not have this problem since these methods penalize the selection of redundant features [23,14].

Finally, Fig. 1 displays a fraction of the transcription control network from *Escherichia coli* with 423 genes. Those nodes with a diamond shape correspond to the genes selected by the ARD method when trained on one of the synthetic 423×250 gene expression matrix generated by SynTREN. The figure clearly shows that most of the genes selected by ARD are hubs or key regulators in the network. Light-gray nodes are those genes selected by the group Lasso method. In this method, most of the selected genes are connected to the same hub or key regulator and in consequence, their expression patterns are highly correlated. The nodes selected by MRMR, although not shown, are in this case very similar to those selected by ARD.

Table 1. Average connectivity and corresponding standard deviation for each method

Network	Genes	Random	Lasso	MRMR	ARD
Small E. coli	423	2.54±0.77	7.23±0.40	13.31±0.77	14.01±0.71
Yeast	690	3.14±0.91	9.46±0.51	13.91±1.27	16.51±0.74
Large E. coli	1330	4.22±1.94	8.56±0.72	14.70±2.48	23.48±1.62

5 Combining ARACNE with the Output of a Hub Gene Selection Method

In this section we describe how to improve the performance of ARACNE using the output of the previous hub gene selection methods. First, we introduce the original ARACNE algorithm. ARACNE is a state-of-the-art method for reverse engineering transcription networks using only steady-state gene expression data [4,16]. ARACNE belongs to a class of network reconstruction algorithms known as mutual information networks [28]. In these methods, a connection between two nodes is created if the corresponding transcript concentrations exhibit large mutual information. The first step of ARACNE is to compute the empirical mutual information for any two transcripts i and j , i.e. $I(\mathbf{x}_i, \mathbf{x}_j)$. The connection weight w_{ij} for any two genes i and j is initialized as $w_{ij} = I(\mathbf{x}_i, \mathbf{x}_j)$. After this, in a pruning step, ARACNE employs the data processing inequality to eliminate indirect interactions between genes. This inequality states that when genes i and j interact indirectly through gene k , it should be satisfied

$$I(\mathbf{x}_i, \mathbf{x}_j) \leq \min \{I(\mathbf{x}_i, \mathbf{x}_k), I(\mathbf{x}_j, \mathbf{x}_k)\}. \quad (16)$$

For each triplet of transcripts, ARACNE applies inequality (16) and sets w_{ij} to zero when (16) holds for some k , otherwise w_{ik} is not changed. The last step of ARACNE is to link each pair of genes i and j when w_{ij} is higher than threshold $\theta > 0$. Note that ARACNE enforces sparsity in the reconstructed network first, by making use of the data processing inequality and second, by means of the threshold θ . However, ARACNE does not take into account the scale-free topological structure of transcription control networks.

Prediction errors in ARACNE are mainly due to the large variance displayed by the mutual information estimates $I(\mathbf{x}_i, \mathbf{x}_j)$ for any two genes i and j . This excessive variance has its origin in the reduced number of measurements and the high level of noise in gene expression data. The performance of ARACNE can be significantly reduced by these random fluctuations in the estimation of the mutual information. In particular, ARACNE may fail to eliminate some of the indirect gene interactions. This occurs when genes i and j interact through gene k indirectly but fluctuations in the mutual information estimation process make $I(\mathbf{x}_i, \mathbf{x}_j)$ larger than $I(\mathbf{x}_i, \mathbf{x}_k)$ or $I(\mathbf{x}_j, \mathbf{x}_k)$. In a similar manner, ARACNE may mistakenly remove a direct interaction between genes i and j when these fluctuations in the estimation process make $I(\mathbf{x}_i, \mathbf{x}_j)$ lower than $I(\mathbf{x}_i, \mathbf{x}_k)$ and $I(\mathbf{x}_j, \mathbf{x}_k)$ for some gene k .

5.1 Modification of the Original ARACNE Method

Prior knowledge about the particular topology of the network can be very useful to reduce the negative impact of random fluctuations in the estimation of the mutual information. In particular, when the underlying network is scale-free any arbitrary node is more likely to be connected to a hub node than to any other ordinary node. This means that, on average, $I(\mathbf{x}_i, \mathbf{x}_j)$ should be larger when either i or j is a hub gene than when none of them is. Using this idea, we propose to improve the performance of ARACNE as follows. After ARACNE has computed the empirical mutual information for all genes, we update the mutual information estimates as

$$I_{\text{new}}(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} I_{\text{old}}(\mathbf{x}_i, \mathbf{x}_j) + \Delta_{ij} & \text{if } i \in S_H \text{ or } j \in S_H \\ I_{\text{old}}(\mathbf{x}_i, \mathbf{x}_j) & \text{otherwise} \end{cases} \quad (17)$$

for $i = 1, \dots, d, j = 1, \dots, d$ and $j \neq i$, where S_H is a set of hub genes and Δ_{ij} is a positive number that scales with the level of noise in the estimation of the mutual information for transcripts i and j . Here, S_H has been generated previously by either the ARD, group Lasso, or MRM methods. The purpose of Δ_{ij} is to generate an increment in the mutual information estimate between genes i and j when one of these genes is a hub. However, this increment has to be adjusted with respect to the level of noise in $I_{\text{old}}(\mathbf{x}_i, \mathbf{x}_j)$. In particular, when the number of available gene expression measurements increases, the amount of noise in $I_{\text{old}}(\mathbf{x}_i, \mathbf{x}_j)$ decreases and Δ_{ij} should decrease too. Otherwise, $I_{\text{new}}(\mathbf{x}_i, \mathbf{x}_j)$ would be asymptotically biased and the performance of the new ARACNE algorithm would not improve as more data samples are available.

If we assume that the elements of $\mathbf{x}_i$ and $\mathbf{x}_j$ have a bivariate Gaussian marginal distribution, then the first step of ARACNE will set

$$I(\mathbf{x}_i, \mathbf{x}_j) = -0.5 \log(1 - \hat{\rho}_{ij}^2), \quad (18)$$

where $\hat{\rho}_{ij}$ is the empirical correlation between $\mathbf{x}_i$ and $\mathbf{x}_j$. In this case, we can fix a probabilistic upper bound on the noise in the estimation of $I(\mathbf{x}_i, \mathbf{x}_j)$ using the delta method, namely

$$\Delta_{ij} = n^{-1/2} |\hat{\rho}_{ij}| \Phi^{-1}(1 - \gamma), \quad (19)$$

where we have approximated the sampling distribution of $\hat{\rho}_{ij}$ by a Gaussian with variance $n^{-1}(1 - \hat{\rho}_{ij}^2)^2$ and mean $\hat{\rho}_{ij}$, n is the number of available expression measurements, Φ^{-1} is the standard Gaussian quantile function and γ is a small positive number that determines the probability of the actual noise being greater than the upper bound.

The update rule described by (17) and (19) generates an increment in the mutual information estimates when one of the two transcripts is a hub gene. This increment is proportional to the uncertainty in the estimation of the mutual information so that we can achieve a balance between the connectivity structure given by the data and our prior assumption that the genes in S_H should be highly connected. The update rule guarantees that direct connections where one

Table 2. Average AUC-PR and corresponding standard deviation for each method

Network	Genes	Standard	New ARACNE with			
		ARACNE	Random	Lasso	MRMR	ARD
Small <i>E. coli</i>	423	0.41±0.03	0.28±0.05	0.41±0.04	0.57±0.04	0.56±0.03
Yeast	690	0.30±0.02	0.16±0.03	0.30±0.02	0.35±0.03	0.39±0.02
Large <i>E. coli</i>	1330	0.16±0.01	0.06±0.01	0.15±0.01	0.18±0.03	0.26±0.02

of the two transcripts is a hub gene are less likely to be mistakenly removed during the pruning step of ARACNE. Furthermore, indirect links between two transcripts regulated by a common hub gene are also more likely to be pruned out during this step. Consequently, the new ARACNE algorithm should have an improved performance over the original version of the method.

6 Evaluation of the New ARACNE Method

To evaluate the accuracy of the new ARACNE method based on a list of hub genes, we perform a series of experiments with synthetic gene expression data. The standard ARACNE algorithm is compared with four versions of the modified method, each one corresponding to a different technique for obtaining the set S_H of hub genes: ARD, group Lasso, MRMR and the random approach. The data used in these experiments are the 150 gene expression matrices generated for the experiments of Sect. 4. Again, the data are transformed using the mapping given by (15). The parameters of the gene selection methods are the same as in Sect. 4. Mutual information in ARACNE is computed as in MRMR and γ is fixed to d^{-1} , where d is the number of genes in the network under analysis.

The performance of each reverse engineering technique is evaluated using the area under the Precision-Recall curve (AUC-PR) which is obtained by altering threshold θ [29]. The precision and the recall rates are computed by comparing the edges of the true network with those edges inferred by the method under analysis. Precision-Recall curves are preferred as a measure of quality over other alternatives like Receiver Operating Characteristics (ROC) curves. The reason for this is that the target class distribution in the network reconstruction problem (edge, no edge) is very skewed and in this case Precision-Recall curves are more suitable for evaluating the performance of a method [29].

Table 2 displays the average and the standard deviation of the AUC-PR measures obtained by each method in the experiments. The random approach decreases the performance of the standard ARACNE algorithm and the group Lasso approach does not seem to provide any relevant improvement. However, the MRMR and ARD feature selection methods give a significant boost to the performance of the original ARACNE algorithm, with ARD obtaining the best results: up to a 62% improvement for the *E. coli* network with 1330 genes.

Finally, Fig. 3 displays the Precision-Recall curves obtained by the different network reconstruction methods when they are executed on a particular gene expression matrix sampled from each of the three transcription control networks:

Fig. 3. Precision-Recall curves for the network reconstruction methods when they are run on a specific gene expression matrix sampled from: The small *E. coli* network (top left), the yeast network (top right) and the large *E. coli* network (bottom). The best performing method is the ARACNE algorithm which employs the list of hub genes obtained by the ARD feature selection approach.

The small *E. coli* network, the yeast network and the large *E. coli* network. The best performing method is the ARACNE algorithm that employs the list of hub genes obtained by the ARD feature selection approach.

7 Experiments with Real Microarray Data

We evaluate the ability of the ARD method for selecting key regulators on a set of expression profiles from *Saccharomyces cerevisiae*. This dataset contains 247 expression measurements for a total of 5520 genes and it is publicly available at the Many Microbe Microarrays Database [17]. Before running the analysis, the gene expression matrix is transformed using (15) and σ is fixed to 1.

Table 3 displays the top ten genes selected by the ARD technique and their description at the *Saccharomyces* Genome Database. From the list of ten genes, YOR224C, YBR289W and YBR160W are involved in transcription or in the

Table 3. Top ten genes selected by ARD on the yeast dataset

Rank	Gene	Description
1	YOR224C	RNA polymerase subunit.
2	YPL013C	Mitochondrial ribosomal protein.
3	YGL245W	Glutamyl-tRNA synthetase.
4	YPL012W	Ribosomal RNA processing.
5	YER125W	Ubiquitin-protein ligase.
6	YER092W	Associates with the INO80 chromatin remodeling complex
7	YBR289W	Subunit of the SWI/SNF chromatin remodeling complex.
8	YBR272C	Involved in DNA mismatch repair.
9	YBR160W	Catalytic subunit of the main cell cycle kinase.
10	YOR215C	Unknown function.

regulation of transcription. YER092W is likely to be a regulator of transcription and YPL013C, YGL245W and YPL012W could have some type of role in post-transcriptional regulation. It is significant that the first gene selected by the ARD method is a component of the RNA polymerase enzyme. In particular, the linear model of transcription that is described in Sect. 2 precisely states that the transcript concentration of RNA polymerase is the most informative feature for solving the multiple regression problem given by (5). Since only 7 genes are described at the *Saccharomyces* Genome Database as RNA polymerase subunits, the probability of obtaining this result by chance is approximately 10^{-4} . Finally, note that YER125W is involved in the degradation of RNA polymerase [30] and also regulates many cellular processes, including transcription.

8 Conclusions

Several machine learning methods are available in the literature for the reverse engineering of transcription networks using only steady-state gene expression data. Most of these methods enforce sparsity in the resulting interaction map. However, few of them take into account the scale-free topology of transcription networks. In this paper, we have focused on the hubs that commonly appear in networks with a scale-free topological structure. Reverse engineering methods can be significantly improved by predicting beforehand those genes that are likely to be hubs or key regulators in the underlying network. For this task, we have introduced a multiple linear regression model which describes the interactions between genes and transcriptional regulators. The model includes a coefficient matrix which is expected to be columnwise sparse. A set of hub genes can then be selected by identifying those columns of the regression matrix which have many non-zero entries. In practice, this is equivalent to selecting those features that are more relevant for solving the multiple regression problem. Three feature selection methods have been proposed for this task. The group Lasso method [12], the automatic relevance determination (ARD) approach [13] and the maximum relevance minimum redundancy (MRMR) criterion [14]. A series of experiments with simulated gene expression data validate the capacity of these methods for

identifying hub genes. The best results are obtained by the feature selection method based on the ARD approach.

Additionally, the performance of ARACNE, a popular method for reverse engineering transcription networks, has been improved by taking into account the list of hub genes generated by the aforementioned feature selection methods. Experiments with simulated gene expression data indicate that a combination of ARACNE and the best feature selection method, ARD, yields up to a 62% improvement in performance with respect to the original algorithm.

Finally, the best feature selection method, ARD, is validated using expression profiles from yeast. This method obtains from an original set with thousands of genes a final set of ten genes that includes many global mRNA regulators.

Acknowledgements

This study was supported under project TIN2007-66862-C02-02 by the Spanish *Dirección General de Investigación* and by a Dutch NWO Computational Life Science grant awarded to Tjeerd M. H. Dijkstra (RUMPHI 600.635.100.08N28).

References

1. Kitano, H.: Systems biology: a brief overview. *Science* 295(5560), 1662–1664 (2002)
2. Stolovitzky, G., Califano, A.: Systems biology: Making sense of oceans of biological data. *The New York Academy of Sciences Update Magazine*, 20–23 (March/April 2006)
3. Tong, et al.: Global mapping of the yeast genetic interaction network. *Science* 303(5659), 808–813 (2004)
4. Basso, K., Margolin, A.A., Stolovitzky, G., Klein, U., Dalla-Favera, R., Califano, A.: Reverse engineering of regulatory networks in human B cells. *Nature Genetics* 37, 382–390 (2005)
5. Faith, J.J., Hayete, B., Thaden, J.T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J.J., Gardner, T.S.: Large-scale mapping and validation of *escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biology* 5, 54–66 (2007)
6. Thieffry, D., Huerta, A.M., Pérez-Rueda, E., Collado-Vides, J.: From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in *escherichia coli*. *BioEssays* 20(5), 433–440 (1998)
7. Albert, R.: Scale-free networks in cell biology. *Journal of Cell Science* 118(21), 4947–4957 (2005)
8. Friedman, N.: Inferring cellular networks using probabilistic graphical models. *Science* 303(5659), 799–805 (2004)
9. Yeung, M.K.S., Tegnér, J., Collins, J.J.: Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences of the United States of America* 99(9), 6163–6168 (2002)
10. Gardner, T., di Bernardo, D., Lorenz, D., Collins, J.J.: Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* 301(5629), 102–105 (2003)

11. Sheridan, P., Kamimura, T., Shimodaira, H.: On scale-free prior distributions and their applicability in large-scale network inference with gaussian graphical models. In: *Complex Sciences*, pp. 110–117. Springer, Heidelberg (2009)
12. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B* 68, 49–67 (2006)
13. Tipping, M.E.: Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research* 1, 211–244 (2001)
14. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(8), 1226–1238 (2005)
15. den Bulcke, T.V., Leemput, K.V., Naudts, B., van Remortel, P., Ma, H., Verschoren, A., Moor, B.D., Marchal, K.: Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics* 7(1), 43 (2006)
16. Margolin, A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R., Califano, A.: Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* 7(Suppl. 1), S7 (2006)
17. Faith, J.J., Driscoll, M.E., Fusaro, V.A., Cosgrove, E.J., Hayete, B., Juhn, F.S., Schneider, S.J., Gardner, T.S.: Many microbe microarrays database: uniformly normalized affymetrix compendia with structured experimental metadata. *Nucleic Acids Research* 36, D866–D870 (2008)
18. Shannon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B., Ideker, T.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research* 13, 2498–2504 (2003)
19. Gardner, T.S., Faith, J.J.: Reverse-engineering transcription control networks. *Physics of Life Reviews* 2(1), 65–88 (2005)
20. Alon, U.: *An introduction to systems biology*. CRC Press, Boca Raton (2006)
21. Barabási, A.L., Oltvai, Z.N.: Network biology: understanding the cell's functional organization. *Nature Reviews Genetics* 5(2), 101–113 (2004)
22. Tibshirani, R.: Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 58(1), 267–288 (1996)
23. Bishop, C.: *Pattern Recognition and Machine Learning*. Springer, Heidelberg (2006)
24. Kim, Y., Kim, J., Kim, Y.: Blockwise sparse regression. *Statistica Sinica* 16, 375–390 (2006)
25. Shen-Orr, S.S., Milo, R., Mangan, S., Alon, U.: Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics* 32, 64–68 (2002)
26. Ma, H.W., Kumar, B., Ditges, U., Gunzer, F., Buer, J., Zeng, A.P.: An extended transcriptional regulatory network of escherichia coli and analysis of its hierarchical structure and network motifs. *Nucleic Acids Research* 32(22), 6643–6649 (2004)
27. Guelzim, N., Bottani, S., Bourguin, P., Képès, F.: Topological and causal structure of the yeast transcriptional regulatory network. *Nature Genetics* 31, 60–63 (2002)
28. Meyer, P.E., Lafitte, F., Bontempi, G.: minet: A r/bioconductor package for inferring large transcriptional networks using mutual information. *BMC Bioinformatics* 9(1), 461 (2008)
29. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: *ICML 2006*, pp. 223–240 (2006)
30. Huibregtse, J.M., Yang, J.C., Beaudenon, S.L.: The large subunit of RNA polymerase II is a substrate of the Rsp5 ubiquitin-proteinligase. *Proceedings of the National Academy of Sciences of the United States of America* 94(8), 3656–3661 (1997)