IRIS: a method for reverse engineering of regulatory relations in gene networks
Sandro Morganella¹,², Pietro Zoppoli¹,² and Michele Ceccarelli*¹,²

Address: ¹Department of Biological and Environmental Sciences, University of Sannio, Benevento, Italy and ²Biogem s.c.a.r.l., Ariano Irpino (Avellino), Italy

Email: Sandro Morganella - morganellaalx@gmail.com; Pietro Zoppoli - zoppoli@biogem.it; Michele Ceccarelli* - m.ceccarelli@gmail.com
* Corresponding author

Abstract

Background: The ultimate aim of systems biology is to understand and describe how molecular components interact to manifest collective behaviour that is the sum of the single parts. Building a network of molecular interactions is the basic step in modelling a complex entity such as the cell. Even if gene-gene interactions only partially describe real networks because of post-transcriptional modifications and protein regulation, using microarray technology it is possible to combine measurements for thousands of genes into a single analysis step that provides a picture of the cell’s gene expression. Several databases provide information about known molecular interactions and various methods have been developed to infer gene networks from expression data. However, network topology alone is not enough to perform simulations and predictions of how a molecular system will respond to perturbations. Rules for interactions among the single parts are needed for a complete definition of the network behaviour. Another interesting question is how to integrate information carried by the network topology, which can be derived from the literature, with large-scale experimental data.

Results: Here we propose an algorithm, called inference of regulatory interaction schema (IRIS), that uses an iterative approach to map gene expression profile values (both steady-state and time-course) into discrete states and a simple probabilistic method to infer the regulatory functions of the network. These interaction rules are integrated into a factor graph model. We test IRIS on two synthetic networks to determine its accuracy and compare it to other methods. We also apply IRIS to gene expression microarray data for the Saccharomyces cerevisiae cell cycle and for human B-cells and compare the results to literature findings.

Conclusions: IRIS is a rapid and efficient tool for the inference of regulatory relations in gene networks. A topological description of the network and a matrix of gene expression profiles are required as input to the algorithm. IRIS maps gene expression data onto discrete values and then computes regulatory functions as conditional probability tables. The suitability of the method is demonstrated for synthetic data and microarray data. The resulting network can also be embedded in a factor graph model.
Background
Although all cells of an organism contain the same DNA, each cell only transcribes and translates a fraction of that DNA. Each cell has a particular interaction pattern that involves genes, proteins and molecules; these complex schema are known as gene regulatory networks. Full understanding of gene interactions can be used to identify methods to control the behaviour of genes directly involved in disease processes. Methods used to infer patterns of interaction among molecular components from observed data are called reverse engineering algorithms. Even if gene-gene interactions only partially describe real networks because of post-transcriptional modifications and protein regulation, using microarray technology it is possible to combine measurements for thousands of genes into a single analysis step that provides a picture of cell gene expression. Therefore, many reverse engineering algorithms rely on gene expression data [1]. These methods differ in the type of expression profiles used, so there are algorithms for time-series data [2], algorithms for steady-state data [3,4] and algorithms that work on both types of data [5]. A graphical representation of the gene regulatory network is often used, with Bayesian networks probably the most popular graphical models used in this scenario [1,5-8]. The major limitation of Bayesian networks is that they cannot represent cyclic structures. To overcome this limitation, methods based on dynamic Bayesian networks have been proposed [9-13].

However, the network topology alone is not enough to perform simulations and predictions of how a molecular system will respond to perturbations. The rules for interactions among the single parts are needed for a complete definition of the network behaviour. Another interesting question is how to integrate the information carried by the network topology, which can be derived from the literature, with large-scale experimental data. Much attention has recently been focused on the modelling [14] and inference [15] of activation rules between molecular components in the cell. Although this can be considered a simpler problem than inference of the network topology, it is important to point out the following:

- Many interaction patterns between molecular components can be obtained from the literature (using, for example, databases such as Ingenuity Pathway Analysis) and integrated in the inference algorithm.
- Owing to the limited amount of experimental data, it can be convenient to solve a simplified problem and exploit as much as possible prior knowledge about the phenomenon being investigated.
- Modelling of the interaction pattern between molecular components can be useful when performing large-scale simulations and deriving hypotheses about the behaviour of biological systems under different conditions [14].

In the present study we follow this research direction. However, instead of considering continuous expression levels, as in [15], we use a discrete representation of gene activation. Indeed, methods based on graphical models often work on discrete data obtained from real-valued gene expression profiles. The discretisation step is of fundamental importance for good accuracy of subsequent computational steps. Śležak and Wróblewski proposed a discretisation approach based on rough set theory in which quality functions are used for roughly discretised data, with inexact dependence between attribute rankings [16]. Friedman et al. considered two different approaches to discretise real-valued data [1]. In the first approach they discretised expression levels to several discrete states according to a fixed discretisation rule and demonstrated that this approach is sensitive to the discretisation procedure. In the second method they combined a linear regression model with the model dependence and measurements, but this approach was strongly affected by the linear dependence. Pe’er et al. proposed a new discretisation procedure for each gene in which gene-specific variation is used to estimate the normal distribution mixture by standard $k$-means clustering [8]. A probabilistic approach is used to identify interactions between genes, such as activation and inhibition, so this method provides a description of the gene network with the interaction features, but many of these interactions are undirected. Moreover, the discretisation step is sensitive to the choice of the number of states that a gene may attain. In this paper we propose a new discretisation approach that depends on the expression profile data (both time-course and steady-state values) for each gene, so that different genes with different expression profiles lead to different discretisation rules. Indeed, we also use this approach to reduce the effect of noise.

After discretisation, the problem faced is how to infer the rules, and not just the pattern, by which the various molecular components interact with each other. We call this problem the inference of regulatory relations given a well-specified gene regulatory network. Gat-Viks et al. proposed an approach to learn improved regulatory functions from high-throughput data using a discrepancy score in which discretisation is carried out as a preprocessing step [17], but the discretisation rules must be determined and tuned rather arbitrarily and each variable is discretised using the same rule. Gat-Viks et al. also proposed a more flexible approach to learn the regulatory functions in a gene network [18], which is represented as a factor graph [19] to model cyclic structures. In this approach discretisation is carried out according to an...
expectation maximisation (EM) algorithm that, combined with the factor graph model, provides a very flexible discretisation scheme. However, in practice this flexibility can lead to over fitting and may decrease learnability, so the authors suggested to use the same or a few discretisation schemes for all the variables. Chuang et al. proposed an approach to infer gene relations from time-course expression profiles in which first and second derivatives are used to detect time-lagged correlated gene pairs [20]. The basic assumption is that pairs of correlated genes exhibit either a complementary pattern (that represents a repressor relation) or a similar pattern (that represents an activator relation). In our approach we propose a simple regulatory function inference that is particularly fast and yields rather good accuracy, even if the network is cyclic. In this step we use an observation similar to that of Chuang et al. [20] for expression profile patterns, but here we use discrete data. Finally, we merge the inferred regulatory functions into a factor graph representation as reported by Gat-Viks et al. [18].

**Implementation**

**Biological Model**

We first define a simple model for biological networks [17]. A biological network can be modelled through a direct graph $G(V, E)$, where each node $v \in V$ represents a gene that can be in a discrete state $D = \{0, 1\}$, representing an inactive and an active state, respectively. If a gene $v \in V$ has at least one parent then it is called a *regulated gene* and we define as $R_v$ the set of parents (regulators) of $v$. If a gene $v \in V$ has no parents, then it is a *stimulator* and we define as $V_s$ the set all stimulators of the network. In addition, we represent the expression data using an $n \times m$ matrix $M$, where $n$ is the number of genes in the network and $m$ is the number of experiments performed or samples. For each $1 \leq i \leq n$ and $1 \leq j \leq m$ the value of $M[i, j]$ is the expression level of gene $i$ in sample $j$.

**IRIS Algorithm**

In this section we describe our approach to infer the regulatory relations in gene networks from high-throughput data. IRIS needs an input network topology $N$, an expression profile data matrix $M$. The method consists of two main steps: (i) Discretisation and (ii) Regulation Functions Learning. The details are reported in the following subsections.

**Discretisation**

This steps is aimed at computing a binary matrix from the observed gene activation levels. The discretisation step uses a matrix of local variation of the gene expression, defined as $M'_i[i, j] = M_i[i, j] - M_i[i, j - 1]$, $j = 2, ..., m$. Let $S$ be the matrix of discrete states, i.e. $S[i, j]$ contains the discrete state relative to the value $M_i[i, j]$. The discretisation procedure is iterative. It first tries to fix the lower values to zero and the upper values to one on the basis of two thresholds $\tau^i_{down}$ and $\tau^i_{up}$, they are computed in order that the interval $[\tau^i_{down}, \tau^i_{up}]$, for each row $i$, contains a given percentage, $\alpha$, of the second and third quartiles of the data. The other values are then fixed on the basis of their nearby values.

We extensively tested various values of $\alpha$ for various datasets, and the results showed that it can be arbitrarily chosen in the range 5%-35% without significantly affecting the results. In all the experiments reported below we choose $\alpha$ as the minimum of this range.

We compute the first discretisation step using the rule:

$$S[i, j] = \begin{cases} 0 & \text{if } M[i, j] \leq 0 \text{ AND } M[i, j] \in (-\infty, \tau^i_{down}] \\ 1 & \text{if } M[i, j] \geq 0 \text{ AND } M[i, j] \in [\tau^i_{up}, +\infty) \\ \text{NaN} & \text{Otherwise} \end{cases}$$

with $2 \leq j \leq m$. And for $j = 1$ we use:

$$S[i, 1] = \begin{cases} 0 & \text{if } M[i, 1] \in (-\infty, \tau^i_{down}] \\ 1 & \text{if } M[i, 1] \in [\tau^i_{up}, +\infty) \\ \text{NaN} & \text{Otherwise} \end{cases}$$

Figure 1 shows an example of matrix $S$ for three genes $A$, $B$ and $C$. The values of $S$ computed as above are used to compute neighbouring values. In particular, the uncertain values (NaN) are recovered as follows:

$$S[i, j] = \begin{cases} 0 & \text{if } S[i, j - 1] = 0 \text{ AND } S[i, j + 1] = 0 \text{ AND } M[i, j] \leq \tilde{x}_i \\ 1 & \text{if } S[i, j - 1] = 1 \text{ AND } S[i, j + 1] = 1 \text{ AND } M[i, j] \geq \tilde{x}_i \end{cases}$$

**Example of the discrete state matrix $S$.**
with $2 \leq j < m - 1$ and where $\bar{x}_i$ is the median value for the expression of gene $i$. To apply these rules we use an iterative approach that runs until either all values are assigned to a valid active/inactive state or no recovery action is performed in the last iteration. Figure 2 shows an example of recovery computation.

**Regulation Functions Learning**

In this step we use the matrix $S$ to compute the PTs and TTs. To infer the PTs we use relative frequencies. Consider a gene $v$ and the set of its regulators $R_v$. Then the matrix $S$ contains several state assignments for the genes in $R_v$ and $v$ itself. Let $\Gamma_v$ be the set of all possible state assignments of the variables in $R_v$.

Then we have:

$$freq(\{r_v, v = s\}) = |\{r_v, v = s\}|$$

with $r_v \in \Gamma_v$ \hspace{1cm} (5)

$$freq(\{r_v, v = 1\}) = |\{r_v, v = 1\}|$$

where $|\{r_v, v = s\}|$ are the occurrence numbers of state assignment $\{r_v, v = s\}$ in $S$. Let

$$PT(\{v = s\} | \{r_v\})$$

be the conditional probability that gene $v$ is in state $s$ given the state assignment $r_v \in \Gamma_v$. Then we compute the conditional probabilities for $v$ as follows:

$$PT(v = 0 | \{r_v\}) = \frac{freq(\{r_v, v = 0\})}{freq(\{r_v, v = 0\}) + freq(\{r_v, v = 1\})}$$

$$PT(v = 1 | \{r_v\}) = \frac{freq(\{r_v, v = 1\})}{freq(\{r_v, v = 0\}) + freq(\{r_v, v = 1\})}$$

\hspace{1cm} (6)

Using (6) we compute the TT for each gene as:

$$TT(\{r_v\}) = \begin{cases} 0 & \text{if } PT(\{v = 0\} | \{r_v\}) > PT(\{v = 1\} | \{r_v\}) \\ 1 & \text{otherwise} \end{cases}$$

\hspace{1cm} (7)

$\forall r_v \in \Gamma_v$ and where $TT(\{r_v\})$ represents the state response of regulated gene $v$ to the state assignment $r_v$ of its regulators. Note that if

$$P(\{v = 0\} | \{r_v\}) = P(\{v = 1\} | \{r_v\}) = 0.50$$

we cannot distinguish between the active and inactive state, so we have an undefined response of regulated gene

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline
 & $M[i,j]$ & $M'[i,j]$ & & $M[i,j]$ & $M'[i,j]$ & & $M[i,j]$ & $M'[i,j]$ \\
\hline
1 & $\bar{x}_i$ & $\bar{x}_i$ & & $\bar{x}_i$ & $\bar{x}_i$ & & $\bar{x}_i$ & $\bar{x}_i$ \\
\hline
\end{tabular}
\caption{(a) Data obtained by the discretisation rule defined in (1). (b) Data obtained after one iteration of the rules defined in (3) and (4). (c) Final data for which all uncertain values were recovered.}
\end{table}
\( v \) to the state assignment \( \{ r_v \} \). This situation is indicated as -1 in the TTs.

**Integration with Factor Graph**

A factor graph is a class of probabilistic models that were originally applied to coding/decoding problems. Using a factor graph we can model complex domain knowledge in which feedback loops play a fundamental role. One of the important advantages of factor graphs is their combination with the sum-product algorithm [19], which is a message-passing algorithm for efficiently computing marginal distributions, even in the presence of cycles.

In our approach the factor graph model is used to combine structural information (network topology) with the inferred regulatory functions. A factor graph contains two types of nodes: factor and variable nodes. We have a variable node for each gene and a factor node linking two variable nodes if and only if a relationship between these two nodes exists. Consider the gene regulatory network depicted in Figure 3(a) containing four genes: \( A \) and \( B \) are regulators of \( C \), \( C \) is a regulator of \( D \), and \( D \) is regulator of \( B \). Note that gene \( A \) is a stimulator of the network. Since there are three regulated genes, we have three PTs representing \( P(C|A, B) \), \( P(D|C) \) and \( P(B|D) \). To translate this gene network into an equivalent factor graph network, we perform the following steps:

- Each gene node becomes a variable node of the factor graph;
- For each regulatory function a factor node must be inserted to link the genes involved;
- Each stimulator must be linked to a specific factor node.

Applying these rules, we obtain the factor graph in Figure 3(b). This model can answer questions such as: "What is the probability that gene \( C \) is active given that genes \( A \) and \( B \) are inactive?" and "What is the likelihood that genes \( B \) and \( A \) are inactive given that gene \( C \) is active?". For this purpose, we set the state of the observed genes and use the sum-product algorithm to compute the posterior distribution of hidden genes. Here we follow the propagation of belief in directed graphs for Forney-style factor graphs [21]. If a stimulator is not fixed, then its factor node will be set with a uniform distribution.

Some considerations about the presence of cycles are useful, indeed this is one of the most important problems in the field of Probabilistic Graphical Models (PGM). The inference in PGM consists in the computation of marginal probabilities of complex probability distributions defined over many variables. Many exact and approximate algo-

---

**Figure 3**

Modelling the gene regulatory network as a factor graph. (a) Gene regulatory network modelled as a direct graph. (b) Equivalent factor graph representation of the network in (a).
rithms have been proposed for this task [22] from Monte Carlo methods to variational methods, mean field methods and belief propagation (BP). The sum-product algorithm, adopted in this paper, is a special case of belief propagation. It is well-known that belief propagation yields exact results if the graphical model does not contain cycles. If the graphical model contains loops, the sum-product algorithm can still yield accurate results using little computational effort [21]. However, if the influence of loops is large, the approximate marginals calculated by BP can have large errors and the quality of the BP results may not be satisfactory. Many recent research efforts in statistical machine learning are devoted to the development of efficient approximate inference algorithms for cyclic graphical models (see for example [23]). For the purposes of this paper we could have to deal with cycles in the case of inference, when we use PTs, and computation of steady states, when using TTs. We adopt the belief propagation algorithm for the former and the algorithm of Gat-Viks et al. [17] for the latter. In any case the IRIS method, proposed here, is by no way influenced by the presence of cycles. In particular IRIS takes as input the description of the network and the expression profiles giving in output a map of the regulatory relations between sets of regulators and regulated genes, this means that all the information used by IRIS are based on "local" relationships between a gene and the set of its regulators. The influence of cycles appears in the successive phases for the use of these relationships in inference tasks. However, the inference in cyclic PGMs is still a very important research question in the field of statistical machine learning and its solution is, of course, outside the scopes of this paper.

**Results**

In this section we report IRIS results for both synthetic networks and microarray expression profiles. IRIS needs a well-defined gene network as input. We say that a gene network is well defined if each of its interactions allows us to distinguish between regulator genes and regulated genes. Given a well-defined network, we have genes with zero regulators (called stimulators representing environmental conditions), genes with one regulator, genes with two regulators, and so on. If a gene has at least one regulator then it has a regulatory function that describes its response to a particular stimulus by its regulator(s). In our approach we suppose, without loss of generality, that a gene can be in one of two states: inactive and active, represented as 0 and 1, respectively. This assumption is commonly used in the literature to distinguish the response of a gene to a given experimental condition.

Given a well-defined gene regulatory network, IRIS computes the regulatory functions, providing two different descriptions: a description in which each interaction is described as a conditional probability table, which we refer to as a potential table (PT), and a description where each regulatory relation is a truth table, which, by analogy to neural logic networks, we refer as a truth table (TT). These two different descriptions allow different analyses. Using the PTs we can execute an inference step to compute the a posteriori probability of hidden genes given observed genes, so that, for example, it is possible to understand how to control a gene using particular environmental conditions. Using the TTs we can compute the steady states of a gene regulatory network. In this scenario, we deal with the problem of the cyclic structure of gene networks, so we use an approach based on the factor graph model [19] as an inference engine and the idea of feedback sets [17] to compute the steady states of the networks.

**Results for Synthetic Networks**

The synthetic networks used to test IRIS were generated using SynTReN [24], it creates synthetic transcriptional regulatory networks and produces simulated gene expression data that approximate experimental data. Network topologies are generated by selecting sub-networks from previously described regulatory networks. Several parameters can be used to adjust the complexity of the data set generated. All gene expression values are normalized between 0 (no transcription) and 1 (maximum level of transcription). In addition, the data generated can be altered by a specified level of biological and experimental noise. Using SynTReN, we generated two synthetic networks representing a sub-pathway of the *E. coli* regulatory network (Figure 4) and a sub-pathway of the *S. cerevisiae* regulatory network (Figure 5). For both synthetic networks we generated a data set 150 samples for each biological noise level in the set \( \{0.10,0.15,0.20,0.25,0.30,0.35,0.40,0.45,0.50\} \), using experimental noise levels of 0.18 and 0.25 for *E. coli* and *S. cerevisiae*, respectively.

Most of the rules governing the activation of genes in these networks have already been investigated in several studies. In particular, for the *E. coli* network we use conclusions from references [25-28] to obtain the regulatory relations for glcD, focA and lacZ. For *S. cerevisiae* we use the results of Wilcox et al. [29] to obtain the regulatory rela-
Using IRIS we inferred the regulatory functions for both synthetic gene networks. To evaluate the accuracy of the PTs computed by IRIS, we used the Kullback-Leibler divergence ($D_{KL}$) [30], which is a measure of the difference between two probability distributions $P$ and $Q$, with $D_{KL}(P||Q) = 0$ if and only if $P = Q$. Figure 6 shows mean $D_{KL}(P_{\text{true}}||P_{\text{IRIS}})$ values as a function of the noise level (asterisks). The results in terms of correct TT entries inferred by the algorithm are reported in Table 1 for a maximum biological 0.5. The size of each regulation table, which depends on the number of regulator genes, is also reported. If $|R_i|$ is the number of regulators of the $i$th gene, then the regulation TT will have a size of $2^{|R_i|}$, assuming that a gene can be in two states. The results show that all but one activation rule were correctly inferred by IRIS.

Both networks generated by SynTReN have an acyclic structure, so we can use the EM-MAP algorithm [31] to compare the performance of IRIS. We used an EM-MAP algorithm implementation of the Bayes net toolbox (BNT) [32] to validate the IRIS discretisation scheme. Figure 6 shows results for different discretisation methods such as equal frequency (diamonds), global width (squares) and equal width (circles). For these methods we used discretisation into two bins. The EM-MAP results for IRIS discretisation are reported in Figure 6 (crosses). The mean values over all levels of biological noise are:

1. $D_{KL}(P_{\text{true}}||P_{\text{IRIS}}) = 0.1872$ and $D_{KL}(P_{\text{true}}||P_{\text{EM-MAP}}) = 0.2865$ for $E.\ coll$

2. $D_{KL}(P_{\text{true}}||P_{\text{IRIS}}) = 0.1743$ and $D_{KL}(P_{\text{true}}||P_{\text{EM-MAP}}) = 0.1821$ for $S.\ cerevisiae$.

Table 2 lists the execution times (taken using a Linux PC with Intel Core Duo CPU at 1.8 Ghz) for IRIS and EM-MAP. We report the running times not for an absolute evaluation, but for a relative comparison, indeed the proposed method yields similar or slightly better results than EM-MAP and requires less computational resources. Actually the running time of IRIS depends linearly on the number of genes, and the number of samples, however it depends exponentially from the maximum in-degree of the node network.

Since the IRIS discretisation scheme can be influenced by the order of the values for expression of each gene in the data set, it is interesting to investigate how the performance changes on randomly changing the order of the data. This is useful for a steady-state data set, for which the order of the expression profile has no biological meaning. Actually the difference between the expression levels at successive points of the profile has been often used to characterise the behaviour of a gene in different conditions, with particular reference to time course experiments (see for example [20]), as it can be used as an indicator of increase/decrease of expression profiles correlated in time. IRIS makes use of this difference in the discretisation procedure to fix the values which are not significantly low or high with respect to the mean expression level in the profile. Actually we found useful to compare differences in the expression levels over different conditions even for non time course data. The intuitive reason for the use of the information carried by the difference of expression levels is based on the fact that in this way we can observe if two or more genes have a common or an opposite pattern, which are indicator of activation and inhibition respectively. For example, suppose that we have a network where the gene $A$ is a regulator of another gene $B$ and let $E_1$ and $E_2$ be two different experimental conditions of a steady state dataset. Suppose also that $a_1(b_1)$ and $a_2(b_2)$ are the expression levels of $A(B)$ under $E_1$ and $E_2$ respectively. Now we can distinguish two different situations:

a) $a_1 < a_2$ and $b_1 > b_2$ (or $a_1 > a_2$ and $b_1 < b_2$): here we can state that in the experiment $E_2$ both genes have an expression level greater (lower) then in $E_1$, in other words, the two genes have a similar behaviour.

Figure 5
Synthetic network for $S.\ cerevisiae$. 

|               | SSL1 | CLN2 | CDC8 | NOT3 | FIT2 | CDC6 | CEF1 | LEU2 | CLB6 | DAL80_GZF3 | IPT1 |
|---------------|------|------|------|------|------|------|------|------|------|-----------|------|
| A1            |      |      |      |      |      |      |      |      |      |           |      |
| A2            |      |      |      |      |      |      |      |      |      |           |      |
| A3            |      |      |      |      |      |      |      |      |      |           |      |

1. $D_{KL}(P_{\text{true}}||P_{\text{IRIS}}) = 0.1872$ and $D_{KL}(P_{\text{true}}||P_{\text{EM-MAP}}) = 0.2865$ for $E.\ coll$

2. $D_{KL}(P_{\text{true}}||P_{\text{IRIS}}) = 0.1743$ and $D_{KL}(P_{\text{true}}||P_{\text{EM-MAP}}) = 0.1821$ for $S.\ cerevisiae$. 

Table 2 lists the execution times (taken using a Linux PC with Intel Core Duo CPU at 1.8 Ghz) for IRIS and EM-MAP. We report the running times not for an absolute evaluation, but for a relative comparison, indeed the proposed method yields similar or slightly better results than EM-MAP and requires less computational resources. Actually the running time of IRIS depends linearly on the number of genes, and the number of samples, however it depends exponentially from the maximum in-degree of the node network.
Table 1: Percentage of correct entries in the inferred truth tables for synthetic networks for *E. coli* and *S. cerevisiae*.

**E. Coli**

| Regulated Gene | Regulator Genes  | TT Size | Correct | Incorrect | Undefined |
|----------------|------------------|---------|---------|-----------|-----------|
| *glcD*         | *glcC arcA*      | 4       | 4       | 0         | 0         |
| *focA*         | *arcA crp*       | 4       | 4       | 0         | 0         |
| *rpoH*         | *crp*            | 2       | 2       | 0         | 0         |
| *tdc*          | *crp*            | 2       | 2       | 0         | 0         |
| *lacZ*         | *crp lacI*       | 4       | 4       | 0         | 0         |
| Total          |                  | 16      | 16      | 0         | 0         |
| Percentage     |                  | 100%    | 0%      | 0%        |           |

**S. cerevisiae**

| Regulated Gene | Regulator Genes  | TT Size | Correct | Incorrect | Undefined |
|----------------|------------------|---------|---------|-----------|-----------|
| *CLN2*         | *SSL1*           | 2       | 2       | 0         | 0         |
| *CDC28*        | *SSL1*           | 2       | 2       | 0         | 0         |
| *NOT3*         | *SSL1*           | 2       | 2       | 0         | 0         |
| *FIT2*         | *SSL1 PDR11*     | 4       | 4       | 0         | 0         |
| *CDC6*         | *PDR11*          | 2       | 2       | 0         | 0         |
| *CEF1*         | *PDR11*          | 2       | 1       | 1         | 0         |
| *LEU2*         | *PDR11*          | 2       | 2       | 0         | 0         |
| *CLB6*         | *PDR11*          | 2       | 2       | 0         | 0         |
| *DAL80_GZF3*   | *PDR11*          | 2       | 2       | 0         | 0         |
| *A2*           | *PDR11 IPT1*     | 4       | 4       | 0         | 0         |
| Total          |                  | 24      | 23      | 1         | 0         |
| Percentage     |                  | 95.83%  | 4.17%   | 0%        |           |

Truth tables (TTs) were computed using the data set with the maximum biological noise level (0.50). "TT Size" reports the number of possible state assignments for the regulator genes. Note that the value for the i-th regulated gene is $2^{|R_i|}$, where $|R|$ is the number of regulators and each gene can be in two states. We distinguish the number of correct/incorrect/undefined inferred states for each regulatory relation and compute these as percentages of the total number of states.
Figure 6
Comparison Kullback-Leibler divergence for (a) E. coli and (b) S. cerevisiae synthetic networks. The x-axis shows the biological noise levels used to generate the data sets. The y-axis represents the $D_{KL}$ values obtained as the mean for all tables of a network for the corresponding biological noise level. The Figure reports the $D_{KL}$ values of EM-MAP obtained using different discretisation approaches: IRIS (crosses), equal frequency (diamonds), global width (squares) and equal width (circles) and the $D_{KL}$ values obtained using IRIS algorithm both in discretisation step and in regulation function learning process (asterisks).

Table 2: Execution time for IRIS and EM-MAP.

| Biological Noise | IRIS  | EM-MAP | Iter | IRIS  | EM-MAP | Iter |
|------------------|-------|--------|------|-------|--------|------|
| 0.10             | 0.929 s | 9.447 s | 6 | 1.833 s | 14.557 s | 5 |
| 0.15             | 0.897 s | 9.435 s | 6 | 1.892 s | 14.399 s | 5 |
| 0.20             | 0.910 s | 8.770 s | 5 | 1.853 s | 14.510 s | 5 |
| 0.25             | 0.905 s | 8.642 s | 5 | 1.874 s | 14.381 s | 5 |
| 0.30             | 0.968 s | 8.787 s | 5 | 1.876 s | 17.446 s | 6 |
| 0.35             | 0.965 s | 8.762 s | 5 | 1.790 s | 17.432 s | 6 |
| 0.40             | 1.041 s | 8.880 s | 5 | 1.825 s | 17.469 s | 6 |
| 0.45             | 0.953 s | 8.674 s | 5 | 1.839 s | 14.600 s | 5 |
| 0.50             | 1.005 s | 8.741 s | 5 | 1.860 s | 14.735 s | 5 |

Each value was obtained as the mean for 10 runs. For EM-MAP we also report the number of iterations to reach convergence.
b) $a_1 > a_2$ and $b_1 < b_2$ (or $a_1 < a_2$ and $b_1 > b_2$): here we can state that in $E_1$ the gene $A$ has an expression level greater (lower) then in $E_2$, whereas, the gene $B$ has an expression level in $E_1$ lower (greater) than in $E_2$, in other words, the two genes have an opposite behaviour which can be observed from an opposite sign of the expression derivative for the genes in the experimental condition $E_2$.

If two genes show the behaviour of the case a) than we can suppose that these two genes have a relationship of activation, whereas, in the second case they have a relationship of inhibition. The fact that for steady state data we have uncorrelated values, suggests to choose any set of points to extract second order information. In order to maintain, a coherence with the case of time course data, without loss of generality, we use the previous point in the expression profile matrix. In order to demonstrate that the choice of the previous point does not significantly affect the results, we evaluated the results in terms of Kullback-Leibler divergence with the known solution on a steady state dataset performing random permutations on the expression profile of each gene. These results reported in Figure 7 show that for different permutations the Kullback-Leibler divergence is nearly constant, in other words, the choice of the previous point has the same impact on the results that could have the choice of any other point for the computation of the difference. Finally, the use of multiple points instead of just the previous one did not produce significant improvements, at the expenses of an increased computational time.

Figure 7
Kullback-Leibler divergence for IRIS using randomised data sets. The S. cerevisiae synthetic network and the data set with biological noise of 0.5 were used in this test. The x-axis represents the percentage of columns swapped randomly. The $D_{KL}$ values reported are the means for 100 runs.

Figure 8
Network for the S. cerevisiae cell cycle.

Results for Microarray Expression Profiles
We also applied the IRIS algorithm to two real data sets comprising microarray expression profiles for the yeast mitotic cell cycle and human B-cells.

Yeast Mitotic Cell Cycle
Figure 8 shows the network topology for the yeast mitotic cell cycle extracted from the study by Noman and Iba [33], where we consider only known interactions reported in the literature. In this network the transcriptional factors $SWI4$, $SWI6$ and $MBP1$ are directly linked to the cyclins $CLN1$, $CLN2$, $CLN3$, $CLB5$ and $CLB6$, which bind to the cyclin-dependent kinase protein $CDC28$, whereas $SIC1$ is an inhibitor of the cyclin $CDC28$ complex. Using the literature [34,35] we obtain a description of the regulatory relations for this network (for more details see Additional
file 1: regulation_true_descriptions.pdf). These relationships can be considered the truth tables for this pathway. To infer the regulatory functions we use the microarray data from Spellman et al. [36].

Table 3 compares literature truth tables with those inferred by IRIS. IRIS properly computes the TTs for the genes CDC28, MBP1, CLN1, CDC6 and CLN2, with an overall accuracy of 75%, an error rate of 14% and 11% undefined activations. One of the main advantages of the PT representation of the regulatory functions is the possibility of performing Bayesian inference with a belief propagation algorithm to compute the marginal distributions of hidden genes given observed genes as reported in Table 4. It is interesting that the marginal distributions in the final column of the table reflect the biological findings reported in the literature shown in the first column of the table.

Finally, the inferred TTs can be used to compute the steady states of the network using an appropriate algorithm [17]. In this case three steady states are identified (Table 5). Among them the first and the third one have been described in literature [35].

### Table 3: Percentage of correct entries in inferred truth tables for the *S. cerevisiae* mitotic cell-cycle network.

| Regulated Gene | Regulator Genes | TT Size | Correct | Incorrect | Undefined |
|----------------|-----------------|---------|---------|-----------|-----------|
| CDC28          | CLN3            | 2       | 2       | 0         | 0         |
| MBP1           | SWI6            | 2       | 2       | 0         | 0         |
| CLN1           | CLN3            | 2       | 1       | 0         | 1         |
| CDC6           | CLB5            | 2       | 2       | 0         | 0         |
| CLN2           | CLN1 SWI6       | 4       | 4       | 0         | 0         |
| CLB5           | SWI6 MBP1 CLB6  | 8       | 7       | 0         | 1         |
| SWI6           | CLN2 CLN3       | 4       | 2       | 2         | 0         |
| CLB6           | SIC1 MBP1 CLB5  | 8       | 6       | 1         | 1         |
| SWI4           | CLN1 CLN2 CLN3  | 8       | 6       | 1         | 1         |
| SIC1           | CLN1 CLN2       | 4       | 1       | 2         | 1         |
| **Total**      |                 | **44**  | **33**  | **6**     | **5**     |
| **Percentage** |                 | **75%** | **13.64%** | **11.36%** |

This table follows the same schema as for Table 1.

### Human B-Cells
Recent studies have demonstrated that the organisation of a gene regulatory network often follows a scale-free nature [37]. A scale-free network is characterised by an inverse relationship between the number of nodes and their connectivity. Another feature of gene networks is the presence of highly connected genes (called hubs). These networks typically contain short feedback loops. To test the suitability of IRIS for scale-free gene regulatory networks, we inferred the regulatory relations from human B-cell data, for which we considered the MYC gene as a major hub. MYC codes for a protein that binds to the DNA sequence of other genes. When MYC is mutated or over expressed, the protein does not bind correctly and often causes cancer. Both the gene expression profiles and network topology were extracted from the results of Basso et al. [3]. The network topology represents the MYC gene and 55 genes directly connected to it. To infer the regulatory rules of this network, we used a subset of 100 expression profiles (in [3] 336 samples are used). Here we use the MYC target gene database (MYC-DB) [38].

Because MYC-subnetwork has an acyclic structure (differently to *S. cerevisiae* cell cycle) we can use EM-MAP algorithm to infer the PTs for this network. In order to
evaluated the method in terms of scalability, Figure 9(a) reports the Kullback-Leibler divergences obtained by IRIS (red asterisk line) and EM-MAP (blue crossed line) for different values of $m/n$ ($m$ is the number of samples and $n$ is the number of genes). As can be seen from the Figure, as the ratio $m/n$ increases the accuracy gets better. This is also evident from Figure 9(b) where the percentage of correct (blue bars), undefined (green bars) and incorrect (red bars) evaluations of IRIS algorithm are reported.

Table 6 shows the results obtained by IRIS on the complete data set composed by 100 samples. Among the genes in MYC-DB, we obtained 93% of correct evaluations. Genes with no specified regulation in MYC-DB such as EEF1E1, TRAP1 and PAICS resulted up-regulated in IRIS. In particular for PAICS this up-regulation was confirmed in [39]. In addition there are 31 genes in the Basso et al. network with no MYC-DB entries, IRIS identifies 24 regulatory relationships that deserve further biological investigation.

### Table 4: Inference results. Column "Biological Findings" lists a short description of the features of interest and references.

| Biological Findings | Observed Genes | Hidden Genes | Inference Results |
|---------------------|----------------|--------------|------------------|
| Strong relationship between cyclins CLB5 and CLB6 [40] and between CLN1 and CLN2 [41] | CLB5 CLB6 | $P(\text{CLB}6 = 0 | \text{CLB}5 = 0) = 0.9851$ | $P(\text{CLB}6 = 1 | \text{CLB}5 = 1) = 1.0000$ |
| Inhibitory activity of SIC1 on cyclins CLB5 and CLB6 [42] | SIC1 CLB5 CLB6 | $P(\text{CLB}5 = 0 | \text{SIC1} = 1) = 0.7367$ | $P(\text{CLB}5 = 0 | \text{SIC1} = 1) = 0.7367$ |
| Inactivation of MBP1 and SWI6 causes CLB5 and CLB6 levels to fall [35] | MBP1 SWI6 CLB5 CLB6 | $P(\text{CLB}5 = 0 | \text{MBP1} = 0, \text{SWI6} = 0) = 0.7742$ | $P(\text{CLB}6 = 0 | \text{MBP1} = 0, \text{SWI6} = 0) = 0.8483$ |
| While CLN1 and CLN2 are active, SIC1 is degraded [43] | CLN1 CLN2 SIC1 | $P(\text{SIC1} = 0 | \text{CLN1} = 1, \text{CLN2} = 1) = 0.6476$ |
| Inactivation activity of MBP1 and SWI6 on CLN1 and CLN2 [44] | MBP1 SWI6 CLN1 CLN2 | $P(\text{CLN}1 = 0 | \text{MBP1} = 1, \text{SWI6} = 1) = 0.6111$ | $P(\text{CLN}2 = 0 | \text{MBP2} = 1, \text{SWI6} = 1) = 0.6148$ |

To infer the conditional probability reported in the "Inference Results" column, the sum-product algorithm was used, and the state of the "Observed Genes" is fixed to derive the potential behaviour of "Hidden Genes".

To infer the conditional probability reported in the "Inference Results" column, the sum-product algorithm was used, and the state of the "Observed Genes" is fixed to derive the potential behaviour of "Hidden Genes".

### Table 5: Steady states for the yeast mitotic cell-cycle network obtained using IRIS

| CLN3 | MBP1 | SWI6 | SWI4 | CLN1 | CLN2 | CLB5 | CLB6 | SIC1 |
|------|------|------|------|------|------|------|------|------|
| 0    | 0    | 1    | 0    | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 1    | 1    | 0    |
| 1    | 0    | 0    | 1    | 1    | 1    | 0    | 0    | 0    |
We also tested IRIS on two real data sets to infer interaction rules for the yeast mitotic cell cycle and the human MYC sub-network. IRIS exhibited good accuracy for these networks compared to literature-derived rules. IRIS relies on knowledge of the network topology, which can be extracted from on-line databases (e.g. KEGG) or can be obtained from network reverse engineering algorithms. In other words regulatory network parameter estimation and model selection are treated and performed as two different tasks. This approach could be useful in studying gene regulatory networks with hundreds of genes as a set of smaller sub-networks, as reported for MYC expression profiles.

IRIS is useful for extracting the main rules within a gene network with a well-defined topology. This information can then be used in subsequent analysis steps, such as probabilistic inference or as a preliminary step for building models of complex biological systems [14].

**Availability and Requirements**

- **Project name**: Inference of Regulatory Interaction Schema (IRIS)
- **Project home page**: http://bioinformatics.biogem.it:8081/BioPlone/downloadfolder/iris-downage
- **Operating system(s)**: Platform independent
- **Programming language**: MATLAB
- **Other requirements**: Graphviz is required if the user wants to create a file containing the gene regulatory network topology
- **License**: GNU GPL
- **Any restrictions to use by non-academics**: None

**Authors’ contributions**

SM performed all the experiments and wrote the code, PZ helped in devising the experiments and analysing the results, and MC proposed the biological problem and devised the method. All authors contributed to the design of the whole work and to the writing of the manuscript. All authors read and approved the final manuscript.

**Additional material**

**Additional file 1**

This is a pdf file that lists the true descriptions of the regulatory functions of all the gene regulatory networks used in this paper. Click here for file

[http://www.biomedcentral.com/content/supplementary/1471-2105-10-444-S1.PDF](http://www.biomedcentral.com/content/supplementary/1471-2105-10-444-S1.PDF)

**Acknowledgements**

We thank the anonymous referees for their comments. PZ was supported by a scholarship of IGRS Biogem. and SM was supported by a scholarship of University of Sannio.
Table 6: IRIS results for the MYC subnetwork including 55 genes directly connected to MYC.

| Gene             | IRIS Inferred Regulation | MYC-DB Regulation |
|------------------|--------------------------|--------------------|
| HSPC111          | Upregulation             | Upregulation       |
| PPAT             | Upregulation             | Upregulation       |
| POLD2            | Upregulation             | Upregulation       |
| NOLSA            | Upregulation             | Upregulation       |
| ZRPI             | Downregulation           | Upregulation       |
| NME1             | Undefined                | Upregulation       |
| EBNA1BP2         | Upregulation             | Upregulation       |
| APEX1            | Upregulation             | Upregulation       |
| NDUFB5           | Undefined                | Not Specified      |
| PSPH             | Upregulation             | Upregulation       |
| EEF1E1           | Upregulation             | Not Specified      |
| CTPS             | Upregulation             | Upregulation       |
| C1QBp            | Upregulation             | Upregulation       |
| SRM              | Undefined                | Upregulation       |
| CCT3             | Upregulation             | Upregulation       |
| NOLCI            | Upregulation             | Upregulation       |
| JTV1             | Upregulation             | Upregulation       |
| TRAPI            | Upregulation             | Not Specified      |
| BOP1             | Undefined                | Upregulation       |
| IARS             | Upregulation             | Upregulation       |
| EIF359           | Upregulation             | Upregulation       |
| PAICS            | Upregulation             | Not Specified      |
| RRSI             | Upregulation             | Upregulation       |
| RCL              | Undefined                | Upregulation       |
| POLR1C           | Upregulation             | Not Present        |
| DKFZP564M182     | Downregulation           | Not Present        |
| HPRT1            | Upregulation             | Not Present        |
| C4orf9           | Upregulation             | Not Present        |
Table 6: IRIS results for the MYC subnetwork including 55 genes directly connected to MYC. (Continued)

| Gene          | Regulation | Note                  |
|---------------|------------|-----------------------|
| MRPL9         | Upregulation | Not Present           |
| LOC283537     | Downregulation | Not Present          |
| STAT3         | Downregulation | Not Present          |
| TUFM          | Undefined   | Not Present           |
| SUPV3L1       | Upregulation | Not Present           |
| MRPL3         | Upregulation | Not Present           |
| LIMK2         | Upregulation | Not Present           |
| ATP6VOD1      | Downregulation | Not Present         |
| MX1           | Upregulation | Not Present           |
| TOMM40        | Upregulation | Not Present           |
| CYC1          | Upregulation | Not Present           |
| NOLA2         | Upregulation | Not Present           |
| MRPL12        | Upregulation | Not Present           |
| TIP-1         | Undefined   | Not Present           |
| BYSL          | Upregulation | Not Present           |
| PFAS          | Upregulation | Not Present           |
| ZT86          | Undefined   | Not Present           |
| TRA@          | Undefined   | Not Present           |
| PRMT3         | Upregulation | Not Present           |
| MGC27165      | Undefined   | Not Present           |
| ATIC          | Upregulation | Not Present           |
| HSD17B8       | Upregulation | Not Present           |
| SSRP1         | Undefined   | Not Present           |
| TEGT          | Downregulation | Not Present     |
| TCP1          | Upregulation | Not Present           |
| Cdna_flj30991 | Undefined   | Not Present           |
| IDH3A         | Upregulation | Not Present           |

Note: Not Specified indicates a gene for which an entry exists in MYC-DB but for which no information on regulation is available; Not Present indicates a gene for which no entry in MYC-DB exists; Undefined indicates a situation for which IRIS cannot distinguish between up- and down-regulation.
References

1. Friedman N, Linial M, Nachman I, Pe'er D: Using Bayesian Networks to Analyze Expression Data. Journal of Computational Biology 2000, 7(3-4):601-620.

2. Bansal M, Della Gatta G, Di Bernardo D: Inference of Gene Regulatory Networks and Compound Mode of Action from Time Course Expression Gene Profiles. Bioinformatics 2006, 22(7):815-822.

3. Basso K, Margolin AA, Stolovitzky G, Klein U, Dalla Fava R, Califano A: Reverse Engineering of Regulatory Networks in Human B Cells. Nature Genetics 2005, 37(4):382-390.

4. Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Dalla Fava R, Califano A: ARACNE: an Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. BMC Bioinformatics 2006, 7(Suppl 1):S7.

5. Yu PP, Wang J, Smith VA, Hartemink AJ, Jarvis ED: Advances to Bayesian Network Inference for Generating Causal Networks from Observational Biological Data. Bioinformatics 2004, 20(18):3594-3603.

6. Imoto S, Goto T, Miyano S: Estimation of Genetic Networks and Functional Structure Between Genes by Using Bayesian Networks and Nonparametric Regression. Pac Symp Biocomput 2002:175-186.

7. Lin S, Imoto S, Miyano S: Dynamic Bayesian Network and Nonparametric Regression for Nonlinear Modeling of Gene Networks from Time Series Gene Expression Data. BioSystems 2004, 75:57-65.

8. Pe'er D, Regev A, Eldan G, Friedman N: Inferring Subnetworks from Perturbed Expression Profiles. Bioinformatics 2001, 17(Suppl 1):S215-S224.

9. Bittner ML, Simon R, Chen Y, Jiang Y,蕾zko RS, Weng Q, Bell D, Zhang Y, Gao Y, Meltzer PS: Molecular Classification of Human Carcinomas Using Gene Expression Profiling. Proc Natl Acad Sci USA 2000, 97:126-131.

10. Schummer M, Alizadeh AA, Golub TR: Discriminating between B-Cell CLL Subtypes Using Gene Expression Profiles. Proc Natl Acad Sci USA 2000, 97:126-131.

11. Bafna V, Chen Y, Batzoglou S: A Probabilistic Methodology for Integrating Knowledge and Experiments on Biological Networks. Journal of Computational Biology 2006, 13(2):165-181.

12. Kschischang FR, Brendan JF, Loeliger AH: Factor Graphs and the Sum-Product Algorithm. IEEE Transactions on Information Theory 2001, 47(2):498-519.

13. Chuang CL, Jen CH, Chen CM, Shieh GS: A Pattern Recognition Approach to Infer Time-lagged Genetic Interactions. Bioinformatics 2008, 24(9):1183-1190.

14. GFF Jr: Normal Realizations. IEEE Transactions on Information Theory 2001, 47(2):520-548.

15. Bishop C: Pattern recognition and machine learning. Springer New York; 2006.

16. Mooli J, Kappen H: Loop corrections for approximate inference on factor graphs. The Journal of Machine Learning Research 2007, 8:1113-1143 [http://jmlr.csail.mit.edu/papers/v8/mooij07a.html].

17. Bujcic K, Van den Van Leemput T, Naudts B, Van Remorel P, Ma H, Vercoren A, De Moor B, Marchal K: SynTREN: a Generator of Synthetic Gene Expression Data for Design and Analysis of Structure Learning Algorithms. BMC Bioinformatics 2006, 7:4-73.