Systems biology

Inferring gene regulatory networks from time series data using the minimum description length principle

Wentao Zhao[1,*], Erchin Serpedin[1] and Edward R. Dougherty[1,2]

[1]Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843-3128, USA and [2]Translational Genomics Research Institute, 400 North Fifth Street, Suite 1600, Phoenix, AZ 85004, USA

ABSTRACT

Motivation: A central question in reverse engineering of genetic networks consists in determining the dependencies and regulating relationships among genes. This paper addresses the problem of inferring genetic regulatory networks from time-series gene-expression profiles. By adopting a probabilistic modeling framework compatible with the family of models represented by dynamic Bayesian networks and probabilistic Boolean networks, this paper proposes a network inference algorithm to recover not only the direct gene connectivity but also the regulating orientations.

Results: Based on the minimum description length principle, a novel network inference algorithm is proposed that greatly shrinks the search space for graphical solutions and achieves a good trade-off between modeling complexity and data fitting. Simulation results show that the algorithm achieves good performance in the case of synthetic networks. Compared with existing state-of-the-art results in the literature, the proposed algorithm exceptionally excels in efficiency, accuracy, robustness and scalability. Given a time-series dataset for *Drosophila melanogaster*, the paper proposes a genetic regulatory network involved in Drosophila's muscle development.

Availability: Available from the authors upon request.

Contact: wtzhao@ece.tamu.edu

1 INTRODUCTION

The construction of gene regulatory networks to model multivariate gene interactions has become a major issue in systems biology and bioinformatics. Among the models relevant to this paper are Boolean networks, which model regulatory relations in terms of Boolean relationships and combinatorial logic circuits (Kauffman, 1969), probabilistic Boolean networks (PBNs), which are composed of finite numbers of constituent Boolean networks, each of which corresponds to a contextual condition determined by variables outside the model (Shmulevich *et al*., 2002), the immediate extension of PBNs to any finite quantization (also referred to as PBNs), the use of Bayesian networks (Pearl, 1988) to model non-temporal probabilistic dependency relations among genes (Friedman *et al*., 2000) and the use of dynamic Bayesian networks (DBNs) (Dean and Kanazawa, 1989) to model temporal stochastic relations among genes (Murphy, 2002). Bayesian networks constrain the network model to be an acyclic graph, which might not be always the case

since feedback loops have been found to be basic motifs in gene regulations. Several fundamental relationships have been established recently between the class of PBNs and the class of DBNs (Lahdesmaki *et al*., 2006). However, with the exception of some one-to-many mappings between the two classes, a complete understanding of the relationships between the two classes is not yet available. Therefore, there is no approach available to transfer the learning and inference techniques from one class of models to the other. In short, one might say that PBNs and DBNs characterize the same probabilistic understanding, with PBNs being more specific in that they specify functional relationships within their constituent Boolean networks.

To capture gene regulations, this paper assumes a probabilistic network modeling framework compatible with the family of models represented by DBNs and PBNs. As opposed to PBNs, where gene interactions are modeled explicitly in terms of binary or multi-valued logical functions, the proposed probabilistic model represents gene interactions in terms of probability tables. In addition, the proposed probabilistic network can be viewed as the transition network present in DBNs. In summary, all of these models can be considered as sharing similar basic features.

Using time-course microarray data, this paper addresses the fundamental problem of inferring the structure (overall set of temporal interactions) of genetic regulatory networks within the framework of the proposed class of probabilistic network models. The strength of temporal relationships will be evaluated by using a cross-time mutual information metric. The minimum description length (MDL) principle (Rissanen, 1978) is utilized to determine a threshold that helps to differentiate between strong and weak relationships. The MDL principle also helps to achieve a good trade-off between the network model complexity and the accuracy of data fitting. The proposed network inference algorithm is composed of two components: encoding of the model, for instance, the network, and encoding of the time-series data. After combining the network and data coding complexities, a general criterion is obtained for constructing the network so as to contain only direct and oriented interactions. The convergence of the proposed MDL-based network inference algorithm is corroborated by the excellent recovery of the topology of some artificial networks and through the error rate plots obtained through extensive simulations on datasets produced by synthetic networks. When applied on real Drosophila time-series datasets, the proposed network inference algorithm corroborates some of the findings of Arbeitman *et al*. (2002), and offers novel insights into the regulatory mechanisms that lie at the basis of

---

*To whom correspondence should be addressed.

embryonic segmentation and muscle development in *Drosophila melanogaster*.

Historically, Tabus and Astola (2001) were the first to report some preliminary results on the potential of the MDL principle in learning gene-expression networks; however, their work is limited to using the MDL principle in the prediction of gene expressions, while the present paper focuses on the more general task of learning the network structure. The mutual information has been exploited in the Reveal algorithm proposed by Liang *et al.* (1998). In contrast to Reveal, the proposed algorithm removes the critical assumption that all genes have to be observed, utilizes only pairwise mutual information, achieves better performance in the presence of reduced number of samples, improves greatly the computational efficiency and requires reduced computing capabilities even in the presence of large-scale networks. For non-time-course measurements, different information-theoretic approaches have been proposed recently (Margolin *et al.*, 2006; Nemenman, 2004). They are not relying on any optimization technique (e.g. MDL); however, they efficiently learn the structures of genetic networks. These information-theoretic approaches possess several attractive features: low computational complexity, novel ideas for quantifying efficiently the dependencies among a large number of genes (e.g. usage of the data processing inequality) and efficient testing (estimation) of various relationships among information-theoretic quantities (entropy, mutual information).

The rest of the paper is organized as follows. Section 2 describes the probabilistic network inference model, evaluates the temporal regulation relationships using a cross-time mutual information metric, describes how to encode the network and the data, and formulates the MDL network inference algorithm. Section 3 consists of two parts. The first part demonstrates the performance of the proposed inference algorithm in the case of synthetic (artificial) networks, generated in accordance with the assumed probabilistic framework. The advantages of the proposed algorithm are illustrated through comparisons with the Reveal algorithm (Liang *et al.*, 1998). In the second part, the proposed inference algorithm is run on real data measured on *D.melanogaster* (Arbeitman *et al.*, 2002). The algorithm provides a novel insight into the regulatory pathways of muscle development in *D.melanogaster*. Finally, in Section 4 the paper concludes with remarks about the proposed network inference algorithm and simulation results; the future research directions are also outlined.

2 SYSTEMS AND METHODS

2.1 Genetic network formulation

Given a set of genes, an oriented graph $\mathbf{G}(\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ denotes the set of vertices and $\mathbf{E}$ represents the set of oriented edges, is used to map the gene interactions. Each vertex represents a specific gene and at a specific time is associated with a gene-expression value. This paper assumes discrete-valued gene expression levels but no specific limit on the number of quantization levels is enforced. Each edge of the graph denotes a directed regulation (i.e. an oriented edge with a precise temporal regulation implication). If gene x regulates gene y, there exists an oriented edge from vertex x to y ($x \rightarrow y$). Gene x can have several immediate upstream regulators, referred to as predecessors. The notation $\mathbb{P}(x)$ is used to represent the set of predecessors that regulate gene x. For instance, if gene x is regulated cooperatively by genes y and z, then $\mathbb{P}(x) = \{y, z\}$. Similarly, the notation $\mathbb{S}(x)$ is used to

Table 1. Probability table for *or*

$yz{:}x$	0	1
00	0.8	0.2
01	0.2	0.8
10	0.2	0.8
11	0.2	0.8

$x = y + z$ with confidence 0.8.

represent the set of successor genes which are regulated by gene x. If gene x regulates simultaneously only the genes y and z, then $\mathbb{S}(x) = \{y, z\}$.

Associated with a specific gene x is the regulation function $f_x(\mathbb{P}(x))$, which denotes the expression value for gene x determined by the values of the genes in the set of predecessors $\mathbb{P}(x)$. For simplicity, the shorthand notation f_x will be used since $\mathbb{P}(x)$ is uniquely determined in the biological world. For instance, the Boolean relation *if either gene y or gene z is induced, gene x will be induced* can be represented by $f_x = y + z$ [with '+' denoting the logical *or* (summation) operator].

The gene expression is affected by many internal and external factors, e.g. other genes, environmental variables and many other unknown factors. Since it is impossible to account for all factors, all regulation functions are assumed probabilistic to reflect this uncertainty. In addition, the gene-expression values are assumed discrete-valued and the probabilistic regulation functions are represented as look-up tables. Suppose each gene expression is quantized into q levels. If x has n predecessors, i.e. $|\mathbb{P}(x)|=n$, then the look-up table corresponding to regulating function f_x contains q^n rows and q columns; hence, a total of q^{n+1} entries. Each entry corresponds to a conditional probability. For instance, with the probability 0.8, x will be induced if y is induced and z is repressed. By denoting the repression and induction as binary values 0 and 1, respectively, the previous regulation function can be expressed in terms of $p(x = 1 \mid yz = 10) = 0.8$. Hence, the entry at row 3 and column 2 is filled with the value 0.8. Considering the relationship $f_x = y + z$ with probability 0.8 and $f_x = \overline{y + z}$ with probability 0.2, where the overline denotes negation, Table 1 can be used to represent this probabilistic relationship.

All the functions are defined over the temporal domain, i.e. the expression values for the set $\mathbb{P}(x)$ at time t determine the value for gene x at time $t + 1$. For this reason all functions must assume a time-dependent form $x_{t+1} = f_x(\mathbb{P}_t(x))$. Given m time-series samples $x_t, \ldots, x_{t+m}$ starting at time t, the information conveyed by these samples is represented in terms of the joint probability function $p(x_t, \ldots, x_{t+m})$. Estimation of joint probability functions over short time periods $k \ll m$, i.e. $\hat{p}(\mathbf{x}_t)$, $\hat{p}(\mathbf{x}_t, \mathbf{x}_{t+1}), \ldots, \hat{p}(\mathbf{x}_t, \ldots, \mathbf{x}_{t+k})$, can be achieved with satisfactory precision, whereas for longer time intervals it becomes more difficult.

We intend to infer temporal regulations from limited number of time-course samples. To assess the algorithm performance two types of errors are defined. The type I error is referred to as the false alarm. If the inference algorithm creates an oriented edge from gene x to gene y, i.e. $x \rightarrow y[x \in \mathbb{P}_{\text{infer}}(y)]$, however in reality either gene x has no influence on gene y or in fact the orientation should be reversed and the edge should be $x \leftarrow y$, i.e. $x \notin \mathbb{P}_{\text{real}}(y)$, then such events are referred to as false alarms. Notations $\mathbb{P}_{\text{real}}(y)$ and $\mathbb{P}_{\text{infer}}(y)$ stand for the sets of actual (true) and inferred predecessors of gene y, respectively. Let e be the total number of edges in the real network and assume that e_f stands for the number of edges classified by the algorithm as type I error, then the false alarm rate is represented by the ratio e_f/e.

The type II error is called the miss error. A miss occurs if gene x regulates gene y, i.e. $x \in \mathbb{P}_{\text{real}}(y)$, but the network inference algorithm fails to make the right connection, either disconnecting x with y or making a wrong orientation from y to x, i.e. $x \notin \mathbb{P}_{\text{infer}}(y)$. If e_m denotes the number of missed edges, then the miss rate is represented by the ratio e_m/e.

A good inference algorithm assumes small error rates for both types of errors. Unfortunately, in general they cannot be controlled at the same time. The Hamming distance, defined as the summation of misses and false alarms, can be used as a simplified metric to assess the performance. It assumes the same loss coefficients for the two types of errors from a decision theoretic point of view. However, in many practical cases a trade-off between these two types of errors has to be considered.

In this paper the concept of mutual information from information theory is used to evaluate the significance of regulation, and the significance threshold is determined using the MDL principle. These two concepts (mutual information and MDL) lie at the basis of the proposed network inference algorithm.

2.2 Metric for assessing temporal regulation

If gene y regulates gene x at time slot t with a latency 1, x_{t+1} has to depend on y_t. Conversely, if gene x at time slot $t + 1$ is dependent on the gene expression y at a previous time slot t, we can infer that gene y regulates gene x in time scale 1. The cross-time dependency is considered as the metric for assessing the temporal regulation. The gene system is assumed to be event driven, i.e. all the regulations are performed step by step and in each step all regulations happen only once. Therefore, the latency parameter is set by default to a unit step.

Compared with the correlation coefficient, the mutual information is suitable for nonlinear relations and represents a good metric for evaluating the dependency between two random variables (Cover and Thomas, 1991). Explicit time stamps are assumed in the mutual information criterion for measuring the significance of gene y regulating gene x in one step:

$$I(x_{t+1}; y_t) = \sum_{x_{t+1}, y_t} \left[p(x_{t+1}, y_t) \cdot \log \frac{p(x_{t+1}, y_t)}{p(x_{t+1}) \cdot p(y_t)} \right], \quad (1)$$

where $p(x_{t+1}, y_t)$ and $p(x_{t+1})$ are cross-time joint and marginal probabilities, respectively. These probabilities are assumed time invariant. It is well known that the mutual information $I(x; y)$ between two arbitrary random variables x and y is always greater than or equal to zero, and it is zero if and only if x and y are independent. Large mutual information between x_{t+1} and y_t supports the proposition that y regulates x in one step with a high probability. In such a case, the inference algorithm assumes an edge from y to x on the graph. Assuming that the q-level quantization of gene expressions admits the alphabet $\mathbb{A}_q = \{0, 1, \ldots, q - 1\}$, the marginal and joint probabilities from m-sample time series $\{x_1, \ldots, x_m\}$ and $\{y_1, \ldots, y_m\}$ are given by the following equation:

$$\hat{p}(x = j) = \frac{1}{m} \sum_{t=1}^{m} \mathbf{1}_{\{j\}}(x_t), \quad (2)$$

$$\hat{p}(x_{t+1} = i, y_t = j) = \frac{1}{m-1} \sum_{t=1}^{m-1} \mathbf{1}_{\{ij\}}(x_{t+1} y_t), \text{ for } i, j \in \mathbb{A}_q. \quad (3)$$

where $\mathbf{1}_{\{\cdot\}}(\cdot)$ stands for the indicator function and is defined as follows:

$$\mathbf{1}_A(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in A, \\ 0 & \text{if } \mathbf{x} \notin A. \end{cases} \quad (4)$$

The mutual information can also be defined between two groups of genes rather than a pair. Only pairwise mutual information is utilized in the proposed algorithm because of the limitation of sample size and computational complexity. It is unlikely that the number of time points available in expensive microarray measurements will rapidly increase in the near future; therefore, the estimation of multivariate probability is less reliable when higher-order statistics are employed. Besides, high-order computations request much more memory and CPU time, which is a huge burden even for mainframe computers if very large-scale networks have to be inferred.

Assume that all the cross-time mutual information between genes are collected in the entries of the regulation matrix $\mathbb{M}$, i.e. $\mathbb{M}_{y,x} = I(x_{t+1}; y_t)$. A key problem that needs to be resolved is to find a proper threshold δ such that when $\mathbb{M}_{y,x} \geq \delta$ (or $\mathbb{M}_{x,y} \geq \delta$), then one can infer with high probability

that y regulates x (or x regulates y) and there is potentially an oriented edge from y to x (or from x to y) in the network graph. On the contrary, if $\mathbb{M}_{y,x} < \delta$ and $\mathbb{M}_{x,y} < \delta$, there is no relationship between x and y, and hence, x and y are disconnected. Then another follow-up step assumes scanning of all candidate edges and trimming of all suspect connections based on a reliable criterion. Another key issue concerns the construction of unbiased and consistent estimators for mutual information in the presence of reduced number of samples. Recent progress in estimating information theoretic quantities has led to a number of good estimators in this regard (Beirlant *et al.*, 1997; Paninski, 2003, 2004; Treves and Panzeri, 1995).

2.3 Minimum description length principle

Given a network and a dataset, the MDL principle is employed to evaluate simultaneously the goodness of fit of the network and the data. Intuitively, the more complicated the network is, the better the data would be fitted. However, very often models which are over-fitted relative to the actual systems are selected, which give rise to numerous errors. The merit of the MDL principle is that it achieves a good trade-off between model complexity and fitness of the data. The MDL principle aims to minimize a criterion L that consists of two parts: the model coding length L_M and the data coding length L_D.

2.3.1 Network coding length The proposed network model is an oriented graph. Its coding length is positively proportional to the storage size of the graph. The proposed model's data structure involves arrays for predecessors and matrices for probability tables. For a vertex x, it is required to maintain an array that records $\mathbb{P}(x)$, and if d_i bits are used to code an integer, $d_i|\mathbb{P}(x)|$ bits are necessary to encode the array that records $\mathbb{P}(x)$. A matrix should also be maintained for conditional probability. If d_f bits are used to represent a floating point number and each vertex is q-level quantized with the alphabet $\mathbb{A}_q$, then $d_f q^{|\mathbb{P}(x)|}(q - 1)$ bits are required to store the conditional probability table associated with vertex x (the multiplicative factor $q - 1$ being due to the fact that one degree of freedom is lost because each row of the conditional probability table adds up to one). Supposing that any of the n vertices in the network is indexed by x_i, the network coding length (L_M) can be expressed as follows:

$$L_M = \Gamma \sum_{i=1}^{n} \left\{ d_i|\mathbb{P}(x_i)| + d_f q^{|\mathbb{P}(x_i)|}(q - 1) \right\}, \quad (5)$$

where Γ is a free parameter used to quantify the gap between the proposed network coding length and the ideal information theoretic benchmark, as well as to offer an additional control mechanism between model and data encoding complexities. In other words, this free parameter can be used to ensure that the model encoding mechanism is consistent with the data encoding mechanism. Note further that the model encoding scheme is not unique, and there are a number of additional unknown factors (number of genes/regulation functions, selection of quantization levels and floating point arithmetic) that might still affect the model and data coding lengths. Normally, Γ should be a positive value less than one ($0 < \Gamma < 1$). As a flexible design variable, Γ can be interpreted as a simple mechanism to balance the uncertainties present in the MDL metric and to weight the relative influence of model and data encoding complexities. Simulation results illustrate that this free parameter enables also a customized trade-off between the two types of inference errors. Γ could be learned from established genetic networks, and it could also be tuned via simulations. The size of integer d_i is determined by the number of vertices $|\mathbf{V}|$. For example, the human genome contains ~40 000 genes and 16 bits are enough to code each gene's index. Therefore, d_i can be expressed as $d_i \lceil \log_2 |\mathbf{V}| \rceil$, where $\lceil \cdot \rceil$ is the ceil function. The size of floating number d_f is determined by the sample size m. If a large sample size is available, then a relatively precise estimation of the probabilities can be achieved. Consequently, each entry in the truth table presents a higher resolution, and needs more bits to encode it. Practically d_f can be represented by $d_f = \lceil \log_2 m \rceil$.

As can be observed from the analytic dependencies present in (5), the network coding length is biased in favor of outgoing edges; i.e. each vertex is more likely to be associated with a large successor set rather than a large predecessor set. However, this feature is consistent with biological findings and does not represent a weakness of the proposed probabilistic modeling framework. Guelzim *et al.* (2002) summarized that the number of regulating genes per regulated gene decayed exponentially whereas the number of regulated genes per regulating gene decayed in a power law and assumed a broader-support distribution. It is also conjectured that multiple predecessors consume more energy, hence make the coding length larger.

2.3.2 Data encoding length Since the network is probabilistic, each gene can randomly commit any value in the alphabet during the next time slot. The network is associated with a Markov chain, which is used to model the transitions between states. These states are represented in terms of the n-gene expression vector $\mathbf{x}_t = (x_{1,t}, \ldots, x_{n,t})^{\mathrm{T}}$. The transition probability $p(\mathbf{x}_{t+1} \mid \mathbf{x}_t)$ can be derived as follows:

$$p(\mathbf{x}_{t+1} \mid \mathbf{x}_t) = \prod_{i=1}^{n} p(x_{i,t+1} \mid \mathbb{P}_t(x_i)). \tag{6}$$

The probability $p(x_{i,t+1} \mid \mathbb{P}_t(x_i))$ can be obtained from the look-up table associated with the vertex x_i and is assumed to be time invariant. Its estimation can be obtained in a way similar to (2):

$$\hat{p}(x_{i,t+1} = j \mid \mathbb{P}_t(x_i)) = \frac{1}{m-1} \sum_{t=1}^{m-1} \mathbf{1}_{\{j\}}(x_{i,t+1} \mid \mathbb{P}_t(x_i)), \text{ for } j \in \mathbb{A}_q. \tag{7}$$

Each state transition brings new information which is measured by the conditional entropy:

$$H(\mathbf{x}_{t+1} | \mathbf{x}_t) = -\log(p(\mathbf{x}_{t+1} \mid \mathbf{x}_t)). \tag{8}$$

Therefore, given m time-series sample points, $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, the total entropy is

$$L_D = H(\mathbf{x}_1) + \sum_{j=1}^{m-1} H(\mathbf{x}_{j+1} \mid \mathbf{x}_j). \tag{9}$$

The term $H(\mathbf{x}_1)$ in (9) is common for all models and can be omitted. The coding length for the data is given by

$$L_D = \sum_{j=1}^{m-1} H(\mathbf{x}_{j+1} \mid \mathbf{x}_j). \tag{10}$$

Once the coding lengths for the network L_M and the sampling data L_D are obtained, the MDL criterion L is immediately obtained by summing up these two components, $L = L_M + L_D$.

2.3.3 Comparison with other criteria Akaike's information criterion (AIC) and Bayesian information criterion (BIC) are two alternative model selection criteria that are widely used in the literature. They can be expressed as follows:

$$\text{AIC} = -\log \ell(\hat{\theta} \mid \mathbf{x}) + K, \tag{11}$$

$$\text{BIC} = -\log \ell(\hat{\theta} \mid \mathbf{x}) + \frac{1}{2} K \log m, \tag{12}$$

where $\hat{\theta}$ stands for the estimation of parameter vector, $\ell(\cdot)$ represents the likelihood function given the sample $\mathbf{x}$, K abstracts the number of parameters and m denotes the sample size. The log-likelihood in essence equals the data encoding length term in the proposed MDL criterion. Their differences lie in the penalty, which specifies the model complexity. The AIC does not take into account the effect of sample size whereas BIC and the proposed MDL absorb it into the penalty part. Particularly, the proposed MDL criterion explicitly dissembles the complicated graph parameters in terms of (5) and provides the flexibility in trading off the two types of errors. The MDL and BIC criteria will share similar asymptotic features if the parameter K is used to represent the network storage size.

ALGORITHM 1

1: Input time series data set
2: Initialize $n, \mathbb{M} \in \mathfrak{R}^{n \times n}, \forall j \in \{1 \cdots n\}, \mathbb{P}(x_j) \Leftarrow \phi$;
3: $\forall (j,k) \in \{1 \cdots n\}^2, \mathbb{M}_{j,k} \Leftarrow I(x_{j,t}; x_{k,t+1})$;
4: $A \Leftarrow reshape(\mathbb{M}, 1, n^2)$, change the matrix into an array;
5: $A \Leftarrow sort(A)$ in ascending order;
6: **for** $i = 1$ to n^2 **do**
7: $\quad \delta \Leftarrow A_{(n^2-i+1)}$;
8: $\quad \forall (j,k) \in \{1 \cdots n\}^2$, if $\mathbb{M}_{j,k} \geq \delta$, $\mathbb{P}(x_k) \Leftarrow \mathbb{P}(x_k) \cup \{x_j\}$;
9: $\quad \forall j \in \{1 \cdots n\}, \mathbb{T}_j \Leftarrow p(x_{j,t+1} \mid \mathbb{P}(x_j))$ by using (7);
10: $\quad$ compute $L_{M,i}, L_{D,i}$ by using (5) and (10) respectively;
11: $\quad L_i \Leftarrow L_{M,i} + L_{D,i}$;
12: **end for**
13: $h \Leftarrow \arg Min_i L_i$;
14: restore network in h^{th} loop, $L_{pre} = L_h$;
15: **for** $i = 1$ to n **do**
16: $\quad$ **for** $j = 1$ to n **do**
17: $\quad\quad$ **if** $j \in \mathbb{P}(x_i)$ **then**
18: $\quad\quad\quad \mathbb{P}(x_i) \Leftarrow \mathbb{P}(x_i) \backslash \{x_j\}$, exclude x_j from predecessors;
19: $\quad\quad\quad$ update $\mathbb{T}_i \Leftarrow p(x_{i,t+1} \mid \mathbb{P}_t(x_i))$ by using (7);
20: $\quad\quad\quad$ compute L_M, L_D by using (5) and (10) respectively;
21: $\quad\quad\quad L \Leftarrow L_M + L_D$;
22: $\quad\quad\quad$ **if** $L > L_{pre}$ **then**
23: $\quad\quad\quad\quad \mathbb{P}(x_k) \Leftarrow P(x_k) \cup \{x_j\}$;
24: $\quad\quad\quad$ **end if**
25: $\quad\quad$ **end if**
26: $\quad$ **end for**
27: **end for** the inferred network.

2.4 Network inference algorithm

Given m data points $(\mathbf{x}_1, \ldots, \mathbf{x}_m)$, where each point consists of n gene expressions, $\mathbf{x}_k = (x_1, \ldots, x_n)^{\mathrm{T}}$ $(k = 1, \ldots m)$, the first step in the network inference algorithm is to evaluate the cross-time mutual information between any two genes, $I(x_{i,t}; x_{j,t+1})$, and to fill up the corresponding entry $\mathbb{M}_{i,j}$ of matrix $\mathbb{M}$. The next step is determination of the dependency threshold δ with the least MDL metric L, a step which is achieved over n^2 iterations, equal to the maximum number of possible connections among n vertices. Actually, the n^2 complexity can be further reduced to $O(n)$ because of a generally accepted fact in the literature: the genetic regulatory networks are sparse and the number of edges $|\mathbf{E}|$ grows linearly with the number of vertices $|\mathbf{V}|$. Such a statistic can be found for the yeast (Guelzim *et al.*, 2002), and Drosophila (Giot *et al.*, 2003). In the i-th iteration, the dependency threshold δ is assigned to be the i-th largest value in $\mathbb{M}$. The edge $x_i \to x_j$ is treated as a potential connection, and x_i is put into $\mathbb{P}(x_j)$, if $\mathbb{M}_{i,j} \geq \delta$; otherwise, the genes x_i and x_j are treated as not being connected, and the set $\mathbb{P}(x_j)$ is left unchanged. Upon obtaining the predecessor set $\mathbb{P}(\cdot)$ for each vertex, by using (7), the set of conditional probabilities can be estimated to fill up the corresponding probability table $\mathbb{T}_{(\cdot)}$ for each vertex. Now all the network parameters have been set up, and the network and the data can be encoded to obtain $L_i = L_{M,i} + L_{D,i}$. After n^2 [or $O(n)$] iterations, all the MDL metrics $L'_i s$ can be compared and the network with the least L can be selected. This preliminary network might contain false connections. Then in the last step, each edge is scanned and temporally deleted to evaluate whether such a deletion is helpful to reduce the MDL metric. If it does, then the edge is formally removed and the network is updated.

The network inference pseudocode can be formulated in terms of the Algorithm 1, where lines 1 and 2 initialize all the variables, line 3 computes all the pairwise mutual information terms, lines 4 and 5 sort the mutual information terms, lines 6–12 perform a forward step by adding edges, lines 13 and 14 obtain the preliminary network, lines 15–27 perform a backward step by deleting possible false-alarm edges and lines 22–24 restore

the network when the deletion is invalid. Note that all function names conform to Matlab conventions.

3 RESULTS AND DISCUSSION

3.1 Simulation on synthetic networks

Next, the performance of the proposed network inference algorithm is evaluated on synthetic random boolean networks. The Reveal algorithm proposed by Liang *et al.* (1998) is used as a benchmark to illustrate the advantages of the proposed algorithm. Kevin Murphy implemented Reveal in a toolbox, which can be downloaded at http://bnt.sourceforge.net. Random boolean networks are created by the method proposed in the Supplementary Material.

Figure 1 shows the performance for Reveal and the proposed algorithm with different Γ configurations. Figure 1A stands for the performance in terms of the Hamming distance. The proposed algorithm achieves much better performance when the sample size is <60. Avoiding high-order mutual information terms makes the proposed algorithm more accurate for small sample size. When larger sample sizes are observed, the performance of the proposed algorithm is similar to that of Reveal. The Hamming distance is not sensitive to different Γ configurations, and the performance curves for different Γ overlap. Figure 1B demonstrates that the proposed algorithm produces less false alarm errors than Reveal. The miss rate is sacrificed in trading for a smaller false alarm rate when Γ is adjusted to a higher value. The functionality of free parameter Γ is obvious and it serves as a good trade-off mechanism between the false alarms and misses. Currently most biological measurements assume within 20–50 time points, and the proposed algorithm possesses an attractive performance right in this range.

The Reveal algorithm assumes that all variables/genes can be observed. Such an assumption does not hold in the biological world due to a number of factors. In general, the biological systems are not autonomous and are always affected by environmental variables. Many genes, e.g. non-coding genes, remain still undiscovered, and hence no up-to-date microarray could measure all the genes. Finally, in general a subnetwork is constructed in representing a specific biological functionality. This observability effect is examined by simulating the algorithms on artificial subnetworks $\mathbf{G}_{\text{sub}}$, which are constructed by randomly selecting nodes and the associated edges from a larger scale network $\mathbf{G}_{\text{big}}$. Figure 2 explains the performance in terms of Hamming distance for both Reveal and the proposed algorithm. The performance advantage of the proposed algorithm is apparent: it is not that sensitive to the observed proportion, i.e. the ratio of the number of vertices in the subnet $|\mathbf{V}_{\text{sub}}|$ over the number of vertices in the larger network $|\mathbf{V}_{\text{big}}|$.

The proposed algorithm runs efficiently. It only employs pairwise mutual information. For an *n*-gene network, n^2 pairwise mutual information terms have to be estimated. Given *m* samples, each mutual information estimation takes $O(m)$ additions and $O(1)$ multiplications. However, if each gene is regulated by at most three other genes, i.e. $|\mathbb{P}(x)| \leq 3$, Reveal has to estimate $\Omega(n^4)$ mutual information terms, which include pairwise and higher-order ones. This makes a big difference between the two algorithms. In practice, on Pentium IV PC with 512 MB memory and both algorithms implemented in Matlab, for a network with 20 nodes, 30 edges and 100 sample points, the proposed algorithm produces a fairly good result in 50 s whereas Reveal requires >600 s, i.e. >10 times speedup improvement.
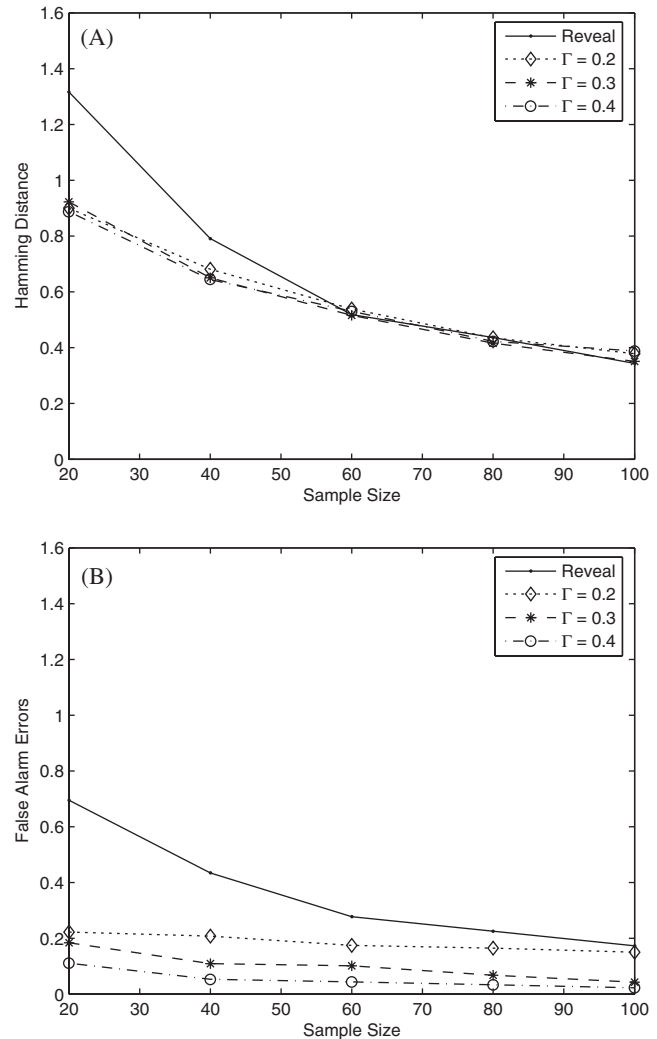

Fig. 1. (A) Hamming distance; (B) false alarm errors. The performance is obtained through averaging over 30 random networks and each network contains 20 vertices and 30 edges. Performance metrics are normalized over 30, the number of edges in synthetic networks.

Reveal can only deal with small networks (with <30 nodes on common PCs) because the space complexity grows as $\Omega(n^4)$, when $\max|\mathbb{P}(x)| \geq 3$. When *n* approaches a large value, Reveal will be out of the capacity of even mainframe computers. However, the proposed algorithm can easily deal with a network with hundreds of nodes and its storage size grows as much as $O(n^2)$. For larger networks, we propose to divide the network into subnets and apply the algorithm on each subnet. This divide and conquer technique relies on the fact that genetic networks are prone to scale free, and the proposed algorithm is not susceptible to the observability effect.

The comparisons between Reveal and the proposed algorithm are summarized in Table 2.

3.2 Simulation on the *Drosophila* dataset

Drosophila and vertebrates share many common molecular pathways, e.g. embryonic segmentation and muscle development. As a simpler species than human, *Drosophila* has fewer muscle types and
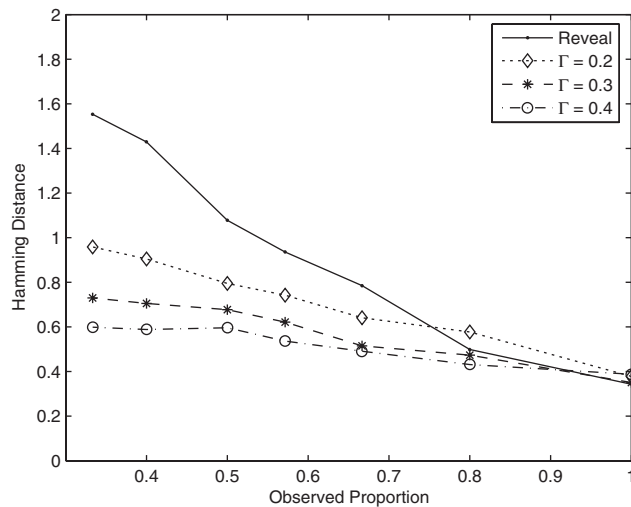
Fig. 2. Observability effects. The performance is obtained by randomly selecting 20 nodes and the associated edges from a larger-scale network. The sample size is kept to 100. The Hamming distance is normalized over the number of edges present in synthetic networks.

Table 2. Performance comparison

Algorithm	Reveal	The proposed algorithm
Small sample performance	Poor	Good
Asymptotic performance	Good	Good
Observability effect	Significant	Minor
Time complexity/efficiency	$\Omega(n^4)$	$O(n^2)$
Largest network processable	Nodes <30	Nodes ≫ 100

The largest network is tested on a PC with 512 MB memory and Pentium IV CPU.

each muscle type is composed of only one fiber type. Besides, *Drosophila* has a shorter life span and assumes a large homogeneous community. These properties make *Drosophila* a good model to study its developmental processes.

Measuring 74 time points, Arbeitman *et al.* (2002) have presented transcriptional profiles for 4028 *Drosophila* genes through the four stages of the life cycle: embryonic, larval, pupal and adulthood. We examine our algorithm using this dataset and propose a novel muscle development network.

In the first step, the original dataset of ratios is quantized into binary values. Let $Y_{(1)}, Y_{(2)}, Y_{(3)}, \ldots, Y_{(n-3)}, Y_{(n-1)}, Y_{(n)}$ be the values of a specific gene expression ordered in ascending order. The smallest two values, $Y_{(1)}$ and $Y_{(2)}$, and the largest two values, $Y_{(n-1)}$ and $Y_{(n)}$, are treated as outliers and discarded. The dynamic range is defined as $Y_{(n-2)} - Y_{(3)}$. The gene expressions are quantized as follows: the upper 50 percentile of the dynamic range R is treated as induced, whereas the lower 50 percentile as repressed. If there is a missing time point, a simple linear interpolation is used, i.e. the value of the missed time point is set to the mean of its two neighbors. When the missing point is a start or end point, it is set as its nearest observed (neighbor's) value.

A set of genes is selected to construct a novel genetic regulatory network for the muscle development process. The selected genes
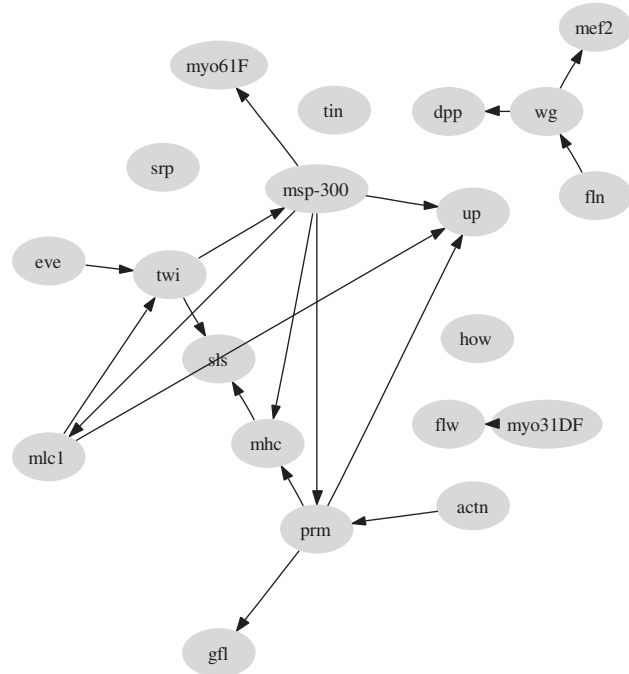

Fig. 3. Muscle development network. Twenty genes are chosen according to their appearance in the literature. The free parameter Γ is 0.19 so that most nodes are connected. The network is split into two domains: muscle motor genes and muscle formation genes.

have been separately reported to relate with muscle development in different works (Giot *et al.*, 2003; Arbeitman *et al.*, 2002; Drosophila Interaction Database http://portal.curagen.com/cgi-bin/interaction/fly Home.pl), but no system level diagram exists yet. The inferred genetic network is shown in Figure 3.

It can be seen in Figure 3 that the gene *muscle specific protein* 300 (*msp*-300), as its name indicates, is a hub gene and regulates *myosin alkali light chain1* (*mlc1*), *myosin heavy chain* (*mhc*), *myosin 61F* (*myo61F*), *paramyosin* (*prm*) and *upheld* (*up*). All these genes except *up* belong to the *myosin* family, which encodes the motor proteins that move along *actin* filaments and are responsible for muscle contraction. These myosin genes play important roles in cellular mechanics and stand nearby in the network.

A loop is found with genes *msp*-300, *twist* (*twi*) and *mlc1*. The boolean relations associated with this loop are $twi \Leftarrow eve \cdot \overline{mlc1}$, $mlc1 \Leftarrow msp\text{-}300$ and $msp\text{-}300 \Leftarrow \overline{twi}$. The network might be intervened by controlling *eve* and *twi*.

The genes *flightin* (*fln*), *wingless* (*wg*), *myocyte enhancing factor 2* (*mef2*) and *decapentaplegic* (*dpp*) form a separate domain from the domain centered around *msp*-300. *Fln* has been shown as a major contributor to muscle development and function. *Wg* functions during metamorphosis to coordinate wing formation and *dpp* acts as a morphogen critical for wing patterning (Shen and Dahmann, 2005). Their cooperation and interactions can be found in Lee and Frasch (2005).

The proposed algorithm provides a systematic view of the *Drosophila's* muscle development. It detaches muscle mechanic genes from formation genes. Further biological experiments are necessary for complete verification of this gene regulatory network.

4 CONCLUSION

An algorithm with good performance in inferring gene regulatory networks from time-series datasets has been designed and implemented. The cross-time mutual information is employed as a metric to discern the oriented connectivity. The MDL principle is used to find the threshold for differentiating between regulation and non-regulation, and to design a network model that achieves a good trade-off between modeling complexity and data fitting accuracy. The proposed network inference algorithm is used for modeling regulatory pathways encountered in embryonic segmentation and muscle development in *D.melanogaster*. The proposed algorithm is practically useful for recovering temporal regulations and can serve as an analysis tool for time-series datasets.

Possible future extensions of this work include (1) combining prior knowledge into the inference algorithm. Such a knowledge might come from a variety of biological sources, e.g. DNA sequencing, gene silencing or other measurements. (2) Implementing the divide and conquer scheme for very large-scale networks. (3) Designing a network inference algorithm that can cope with continuous datasets, without adopting any quantization procedure.

ACKNOWLEDGEMENTS

REFERENCES

Arbeitman,M.N. *et al.* (2002) Gene expression during the life cycle of *Drosophila melanogaster*. *Science*, **297**, 2270–2275.

Beirlant,J. *et al.* (1997) Nonparametric entropy estimation: an overview. *Int. J. Math. Stat. Sci*, **6**, 17–39.

Cover,T.M. and Thomas,J.A. (1991) *Elements of Information Theory*. Wiley-Interscience, New York.

Dean,T. and Kanazawa,K. (1989) A model for reasoning about persistence and causation. *Comput. Intell.*, **5**, 142–150.

Friedman,N. *et al.* (2000) Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, **7**, 601–620.

Giot,L. *et al.* (2003) A protein interaction map of *Drosophila melanogaster*. *Science*, **302**, 1727–1736.

Guelzim,N. *et al.* (2002) Topological and causal structure of the yeast transcriptional regulatory network. *Nat. Genet.*, **31**, 60–63.

Kauffman,S.A. (1969) Metabolic stability and epigenesist in randomly constructed genetic nets. *Theor. Biol.*, **22**, 437–467.

Lahdesmaki,H. *et al.* (2006) Relationships between probabilistic Boolean networks and dynamic Bayesian networks as models of gene regulatory networks. *Signal Process*, **86**, 814–834.

Lee,H.H. and Frasch,M. (2005) Nuclear integration of positive Dpp signals, antagonistic Wg inputs and mesodermal competence factors during *Drosophila* visceral mesoderm induction. *Development*, **132**, 1429–1442.

Liang,S. *et al.* (1998) REVEAL: a general reverse-engineering algorithm for inference of genetic network architectures. *Proc. Pac. Symp. Biocomput.*, **1998**, 18–29.

Margolin,A. *et al.* (2006) ARACNE: an algorithm for reconstruction of genetic networks in a mammalian cellular context. *BMC Bioinformatics*, **7**, S7.

Murphy,K. (2002) Dynamic Bayesian networks: representation, inference and learning. *Technical Report*. University of California, Berkeley, Computer Science Division, CA.

Nemenman,I. (2004) Information theory, multivariate dependence, and genetic network inference. *Technical Report NSF-KITP-04-54*, KITP, UCSB, arXiv: q-bio/0406015.

Paninski,L. (2003) Estimation of entropy and mutual information. *Neural Comput*, **15**, 1191–1253.

Paninski,L. (2004) Estimating entropy on *m* bins given fewer than *m* samples. *IEEE Trans. Inform. Theory*, **50**, 2200–2203.

Pearl,J. (1988) *Probabilistic Reasoning In Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, CA.

Rissanen,J. (1978) Modeling by shortest data description. *Automatica*, **14**, 465–471.

Shen,J. and Dahmann,C. (2005) Extrusion of cells with inappropriate Dpp signaling from *Drosophila* wing disc epithelia. *Science*, **307**, 1789–1790.

Shmulevich,I. *et al.* (2002) From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proc. IEEE*, **90**, 1778–1792.

Tabus,I. and Astola,J. (2001) On the use of MDL principle in gene expression prediction. *EURASIP J. Appl. Si. Pr.*, **2001**, 297–303.

Treves,A. and Panzeri,S. (1995) The upward bias in measures of information derived from limited data samples. *Neural Comput*, **7**, 399–407.