The Max-Min High-Order Dynamic Bayesian Network for Learning Gene Regulatory Networks with Time-Delayed Regulations

Yifeng Li, *Member, IEEE,* Haifen Chen, Jie Zheng, and Alioune Ngom, *Member, IEEE*

Abstract—Accurately reconstructing *gene regulatory network* (GRN) from gene expression data is a challenging task in systems biology. Although some progresses have been made, the performance of GRN reconstruction still has much room for improvement. Because many regulatory events are asynchronous, learning gene interactions with multiple time delays is an effective way to improve the accuracy of GRN reconstruction. Here we propose a new approach, called *Max-Min high-order dynamic Bayesian network* (MMHO-DBN) by extending the *Max-Min hill-climbing* Bayesian network technique originally devised for learning a Bayesian network's structure from static data. Our MMHO-DBN can explicitly model the time lags between regulators and targets in an efficient manner. It first uses constraint-based ideas to limit the space of potential structures, and then applies search-and-score ideas to search for an optimal HO-DBN structure. The performance of MMHO-DBN to GRN reconstruction was evaluated using both synthetic and real gene expression time-series data. Results show that MMHO-DBN is more accurate than current time-delayed GRN learning methods, and has an intermediate computing performance. Furthermore, it is able to learn long time-delayed relationships between genes. We applied sensitivity analysis on our model to study the performance variation along different parameter settings. The result provides hints on the setting of parameters of MMHO-DBN.

Index Terms—Max-Min hill-climbing, higher-order dynamic Bayesian network, gene regulatory network, time-delayed regulation.

1 Introduction

Learning gene interactions is a fundamental problem in systems biology. It is well known that the expression of a given gene can affect how certain genes are expressed, either down-regulated or up-regulated. Such regulatory events within an organism are often asynchronous, that is, different genes can regulate other genes at different time-scales and with different delays. Accurate reconstruction of gene regulations is a challenging task which has recently become possible due to large-scale high-throughput experiments such as microarray experiments [1]. Gene expression levels obtained over sufficiently large numbers of time-points can be used to identify GRNs. A GRN represents causal relationships among genes, encoding all the temporal dependencies between genes in an organism [2].

Accurate and efficient reconstruction of GRNs from time-series gene expression data is a computationally difficult task. One reason is the number of genes often far exceeds the number of time points in gene expression data, where the expression levels are measured for thousands of genes while over just tens of time points. Such a characteristic of data limits the performance of some GRN reconstruction methods such as *ordinary differential equations* (ODEs) based techniques [2], which work prohibitively slowly on such data. Other methods such as Boolean networks [2] could not infer causal relationships and are not very robust to noise in the data. Furthermore, they cannot handle the uncertainty of the data, which is ubiquitous in biological system. On the other hand, *probabilistic graphical models* (PGMs), such as Bayesian networks and Markov random fields [2], have become more popular due to their inherent ability to process uncertain data and their robustness to noise. Importantly, they are able to handle missing data. PGMs are also efficient for processing large numbers of genes [3].

A *Bayesian network* (BN) is a PGM which utilizes a directed acyclic graph (DAG) to compactly represent a set of random variables and the conditional dependencies among them [4]. For GRN reconstruction, *Bayesian networks* model genes as nodes and causal dependencies between genes as edges [5]. BNs cannot model feedback loops because of the acyclicity constraint in their structures. They are unable to learn time-delayed regulations either for similar reason. Both feedback loops and time-delayed regulations are important characteristics of GRNs. To tackle these limitations, *Dynamic Bayesian networks* (DBNs) [6] have been proposed. By unrolling a BN over time, DBN obtains a transition network between any two consecutive time-points which characterizes a GRN. In this transition network, interactions only go from genes at time-point $t-1$ to genes at time-point t. This is the assumption of *first-order DBN* (FO-DBN). By modeling causal dependencies of genes on this temporal transition network, DBN is able to handle feedback loops and time-delayed regulations. A hybrid of ODE and DBN was proposed in [7] to take advantage of both of them.

- *Y. Li and H. Chen have equal contributions, and thus they should be treated as co-first authors.*
- *Y. Li is with the Centre for Molecular Medicine and Therapeutics, Department of Medical Genetics, University of British Columbia, Vancouver, BC, Canada, V5Z 3S6. Email: yifeng@cmmt.ubc.ca*
- *H. Chen and J. Zheng are with the Medical Informatics Research Lab, School of Computer Engineering, Nanyang Technological University, 639798, Singapore. Email:{hchen009,zhengjie}@e.ntu.edu.sg*
- *A. Ngom is with the School of Computer Science, University of Windsor, Windsor, ON, Canada, N9B 3P4. Email: angom@uwindsor.ca*

Manuscript received February 01, 2015; revised XXX.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2015.2474409, IEEE/ACM Transactions on Computational Biology and Bioinformatics

However, FO-DBNs cannot model time-delayed interactions with multiple time lags. To cope with such a problem, a *high-order DBN* (HO-DBN) was introduced in [8] to model gene regulations with multiple time delays. Please see Section 2 for the formal definition of HO-DBN and for the notations and concepts used in this paper. For state-of-the-art of HO-DBN, please refer to Section 3.

In this paper, we propose a new HO-DBN structure learning algorithm, called Max-Min HO-DBN (MMHO-DBN), based on an appropriate extension of the original MMHC-BN algorithm of [4] which was devised to alleviate the limitations of current BN approaches for learning structures of BNs from static data. Details about MMHC-BN algorithm and our MMHO-DBN method would be introduced in Section 4. Both Max-Min approaches are hybrid local learning methods which fuse concepts from both constraint-based Bayesian techniques and search-and-score Bayesian methods. By properly extending the static MMHC-BN algorithm to a dynamic high-order Bayesian model, our method is able to reconstruct gene regulations from high-throughput gene expression data efficiently. Experiments have been carried out on both synthetic and real-life data. Our results show that the proposed approach is more accurate and faster than existing HO-DBN methods.

The preliminary work of MMHO-DBN was presented at the IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, held in Singapore, 2013 [9]. The major changes in this journal version are summarized below:

1) Time complexity of MMHO-DBN is discussed.
2) MMHO-DBN is evaluated on multiple data sets whose actual networks are known.
3) Newer published methods, TDARACNE and TD-LASSO, are compared with MMHO-BN.
4) Sensitivity analysis on the parameters of MMHO-DBN is reported.

The rest of our paper is organized as follow. We first present GRN modeling with HO-DBNs in Section 2. Then, we discuss current HO-DBN methods to learn GRN structures from microarray time-series data in Section 3. After that, we introduce our MMHO-DBN structure learning method in Section 4. Experimental results and sensitivity analysis are presented in Section 5. Finally, we conclude and suggest possible direction of research in HO-DBN learning.

2 Modeling Time-Delayed Regulations with HO-DBNs

The input data for DBN learning are time-series gene expression data. A time-series gene expression data set can be viewed as a matrix: $\boldsymbol{g}_{T\times N} = (\boldsymbol{g}_1, \cdots, \boldsymbol{g}_T)^\intercal$, and $\boldsymbol{g}_t = (g_{t,1}, \ldots, g_{t,N})$, where $g_{t,j}$ represents an observation for random variable $G_{t,j}$, for $1 \le t \le T$, $1 \le j \le N$. This matrix shows the observations (i.e. expression levels) of N genes along T time points.

DBNs are able to model feedback loops and time-delayed interactions among genes, because they model causal dependencies of genes among different time points. DBNs can be categorized into two classes: First-order DBNs (FO-DBNs) and high-order DBNs (HO-DBNs). FO-DBN [10] assumes one-order Markov dependency on temporal gene regulations, which means that the gene expression levels at time t only depend on the gene expression levels at time $t-1$ and t. FO-DBN can be defined by a pair of structures and their transition network: (S_{t-1}, S_t, $S_{[t-1,t]}$), where S_{t-1} and S_t correspond to networks at time point $t-1$ and t respectively. The transition network shows interactions between S_{t-1} and S_t, and thus has $2N$ nodes. By unrolling the transition network over time, FO-DBN obtains its structure which shows how genes regulate each other within one time delay. FO-DBN cannot model gene interactions with time delays larger than one, for which, we need to resort to HO-DBN.

HO-DBN [3], [8], [11], [12] can be represented as r-order DBN (r-DBN) with $r > 1$. The r-DBN assumes an r-order Markov dependency over time, that is, the expression level of a gene at time t depends only on the expression levels of genes at $t-r, t-(r-1), \ldots, t-1, t$. An r-order DBN can be defined by an $(r+1)$-tuple of structures $(S_{t-r}, S_{t-(r-1)}, \ldots, S_{t-1}, S_t)$ and a transition network $S_{[t-r,t-(r-1),\ldots,t-1,t]}$ (or $S_{[t-r,t]}$, for short) among the $(r+1)$ structures. The structures $(S_{t-r}, S_{t-(r-1)}, \ldots, S_{t-1}, S_t)$ correspond to the networks at time $t-r, t-(r-1), \ldots, t-1, t$, and the transition network represents the connectivity from each network S_{t-l} to S_t, $1 \le l \le r$. Here we assume that the networks S_{t-l}, (where $0 \le l \le r$) are without edges and the transition network is an r-order stationary Markov chain. Figure 1 shows a comparison between stationary transition network and non-stationary transition network, and Figure 2 illustrates an example of the transition network of a DBN under the second-order stationary Markov assumption. The structure of a GRN inferred by the r-order DBN can be represented with a matrix $\mathbf{C} = \{c_{i,j}\}_{N\times N}$, where $0 \le c_{i,j} \le r$ denotes the time delay of regulation between gene i and its parent gene j.

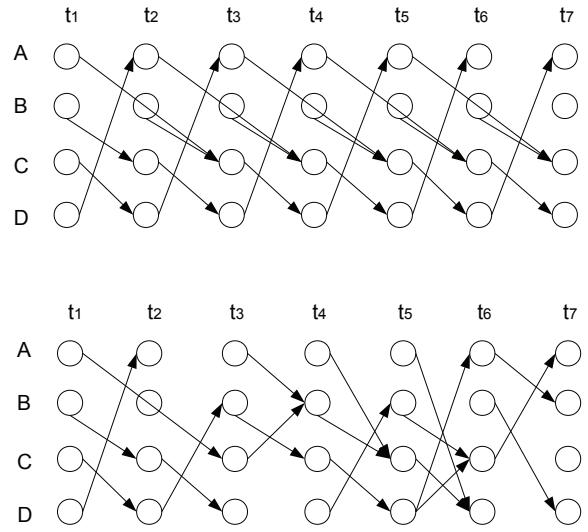


Fig. 1: An example of a second-order stationary transition network (top) and a non-stationary transition network (bottom).

In Bayesian network modeling, a gene is a random variable. Let $\mathbf{G} = (\mathbf{G}_1, \ldots, \mathbf{G}_T)^\intercal$ where each $\mathbf{G}_t = (G_{t,1}, \ldots, G_{t,N})^\intercal$ $(1 \le t \le T)$ is an N-dimensional random

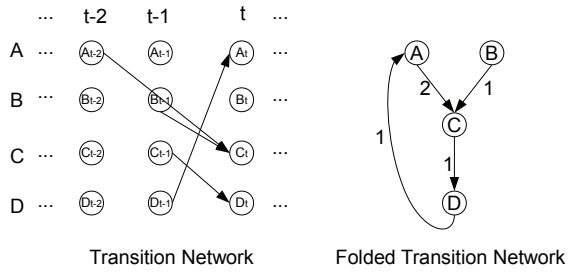


Fig. 2: An example of a DBN under the second-order stationary Markov assumption. Left: transition network. Right: folded transition network.

variable vector representing expression levels of N genes at the t-th time point. The r-order DBN assumes an r-order Markov dependency over time:

$$P(\mathbf{G}_t|\mathbf{G}_{t-1},\ldots,\mathbf{G}_1) = P(\mathbf{G}_t|\mathbf{G}_{t-1},\ldots,\mathbf{G}_{t-r}). \quad (1)$$

By assuming independence of non-descendent variables, the joint probability distribution of G given the structure S can be decomposed into a product of conditional probabilities:

$$P(\mathbf{G}) = \prod_{t=1}^{T} P(\mathbf{G}_t|\mathbf{G}_{t-1},\ldots,\mathbf{G}_{t-r}). \quad (2)$$

We define the parent nodes of the j-th gene at time $t-l$ ($1 \le l \le r$) as a $q_{t-l,j}$-dimension random vector $\mathbf{P}_{t-l,j} = (P_{t-l,j,1},\ldots,P_{t-l,j,q_{t-l,j}})^\intercal$; $\mathbf{P}_{t-l,j} = \emptyset$ if $t-l \le 0$. Then all the parents of the j-th gene can be defined as a q_j-dimension vector $\mathbf{P}_{[1,r],j} = \bigcup_{l=1}^{r} \mathbf{P}_{t-l,j}$, where $q_j = \sum_{l=1}^{r} q_{t-l,j}$. The conditional probabilities in Equation (2) can be further decomposed into a product of conditional probabilities of each gene given its parents $\mathbf{P}_{[1,r],j}$:

$$\begin{aligned} P(\mathbf{G}_t|\mathbf{G}_{t-1},\ldots,\mathbf{G}_{t-r}) &= \prod_{j=1}^{N} P(G_{t,j}|\mathbf{P}_{t-1,j} \cup \cdots \cup \mathbf{P}_{t-r,j}) \\ &= \prod_{j=1}^{N} P(G_{t,j}|\mathbf{P}_{[1,r],j}). \end{aligned} \quad (3)$$

To find the conditional probabilities $P(G_{t,j}|\mathbf{P}_{[1,r],j})$ which best explain the data is the key step in GRN inference using r-DBN. By parameterizing $P(\mathbf{G})$ with a parameter vector $\theta = (\theta_1,\ldots,\theta_N)$, we can transform the problem of finding the optimal $P(\mathbf{G})$ into the estimation of the best θ. After parameterizing and substituting Equation (3) into Equation (2) we obtain the discrete r-DBN model:

$$P(\mathbf{G}|\theta) = \prod_{t=1}^{T}\prod_{j=1}^{N} P(G_{t,j}|\mathbf{P}_{[1,r],j};\theta_j). \quad (4)$$

To model higher-order GRN relationships from time-series data, we need to devise criteria for evaluating the goodness of a structure and devise search algorithms for searching in the large space of candidate structures. In the framework of Bayesian modeling, the optimal structure $\hat{S}$ is obtained by maximizing the posterior probability $P(S|\mathbf{G})$:

$$P(S|\mathbf{G}) = \frac{P(S)P(\mathbf{G}|S)}{P(\mathbf{G})} \propto P(S)P(\mathbf{G}|S), \quad (5)$$

where $P(S)$ is the prior probability of the network structure S and $P(\mathbf{G}) = \sum_S P(S)\int_\theta P(\mathbf{G}|S,\theta)P(\theta|S)d\theta$ is constant, independent of S, and can be removed since it does not relate to structure evaluation. The marginal likelihood of the time-series data given the set of conditional distributions with parameter θ can be expressed as:

$$P(\mathbf{G}|S) = \int_\theta P(\mathbf{G}|S,\theta)P(\theta|S)d\theta, \quad (6)$$

where $P(\theta|S)$ is the prior probability of the parameter θ and $P(\mathbf{G}|S,\theta) = P(\mathbf{G}|\theta_S)$; note that we write $\theta_S = (\theta_1,\ldots,\theta_N)$ since the form of θ in Equation (4) is equivalent to the network structure S. Then the optimal structure $\hat{S}$ from the *maximum a-posteriori* (MAP) estimation is given as:

$$\begin{aligned} \hat{S}_{\text{MAP}} &= \arg\max_S P(S)P(\mathbf{G}|S) \\ &= \arg\max_S P(S)\int_{\theta_S} P(\mathbf{G}|\theta_S)P(\theta_S|S)d\theta_S. \end{aligned} \quad (7)$$

To obtain the MAP estimation of the optimal structure, we need to determine the conditional probabilities $P(\mathbf{G}|\theta_S)$ and the prior probabilities $P(S)$ and $P(\theta_S|S)$. Then by combining these terms into a single criterion, which is used in the search algorithm to evaluate the goodness of candidate GRN structures, we can find the optimal structure $\hat{S}$ from the MAP estimation.

The gene expression values are assumed to be discretized into d levels such that $g_{t,j} \in \{1,\ldots,d\}$ and d denotes the maximum level of expression of any gene. The number of distinct states that $\mathbf{P}_{[1,r],j}$ (the parents set of the j-th gene) can take is $Q_j = d^{q_j}$. Each state of $\mathbf{P}_{[1,r],j}$ is also associated with a lag vector $\mathbf{L}_{[1,r],j} \in \{1,\ldots,r\}^{q_j}$ containing the delay of each parent of gene j; hence, the number of distinct lag vectors for gene j is $L_j = r^{q_j}$. Let $\theta_{j,q,l,k}$ be $P(G_{t,j} = k|\mathbf{P}_{[1,r],j} = q, \mathbf{L}_{[1,r],j} = l)$ (where k, q and l are any distinct states of the target gene j, its parents, and parents' lag vector, respectively), $N_{j,q,l,k} = \sum_{t=1}^{T}\delta(G_{t,j} = k, \mathbf{P}_{[1,r],j} = q, \mathbf{L}_{[1,r],j} = l)$ be the number of observations satisfying $G_{t,j} = k$, $\mathbf{P}_{[1,r],j} = q$, $\mathbf{L}_{[1,r],j} = l$, and $N_{j,q,l,k}^{[t-r,t]}$ be the number of such observations in the transition network $S_{[t-r,t]}$ for $1 \le t \le T$. Using the property of decomposability [6], we can model $P(\mathbf{G}|\theta_S)$ as a multinomial distribution with parameter $\theta_S = (\theta_{j,q,l,k})_{N\times Q_j\times L_j\times d} = (\theta_{1,1,1,1},\ldots,\theta_{N,Q_j,L_j,d})^{\mathrm{T}}$:

$$\begin{aligned} P(\mathbf{G}|\theta_S) &= \prod_{t=1}^{T}\prod_{j=1}^{N}\prod_{q=1}^{Q_j}\prod_{l=1}^{L_j}\prod_{k=1}^{d} \theta_{j,q,l,k}^{N_{j,q,l,k}^{[t-r,t]}} \\ &= \prod_{j=1}^{N}\prod_{q=1}^{Q_j}\prod_{l=1}^{L_j}\prod_{k=1}^{d} \theta_{j,q,l,k}^{N_{j,q,l,k}} \end{aligned} \quad (8)$$

Define $\theta_S = \bigcup_{j=1}^{N}\{\theta_j\}$, $\theta_j = \bigcup_{q=1,l=1}^{Q_j,L_j}\{\theta_{j,q,l}\}$, and $\theta_{j,q,l} = \bigcup_{k=1}^{d}\{\theta_{j,q,l,k}\}$. Assuming that the global and local

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2015.2474409, IEEE/ACM Transactions on Computational Biology and Bioinformatics

parameter vectors are independent of each other, then we can decompose the prior distribution on θ_S as:

$$P(\theta_S|S) = \prod_{j=1}^{N} P(\theta_j|S) = \prod_{j=1}^{N}\prod_{q=1}^{Q_j}\prod_{l=1}^{L_j} P(\theta_{j,q,l}|S) = \prod_{j=1}^{N}\prod_{q=1}^{Q_j}\prod_{l=1}^{L_j}\prod_{k=1}^{d} P(\theta_{j,q,l,k}). \quad (9)$$

Substituting Equations (8) and (9) into the marginal likelihood part of Equation (7), we obtain

$$P(\mathbf{G}|S) = \prod_{j=1}^{N}\prod_{q=1}^{Q_j}\prod_{l=1}^{L_j}\int \prod_{k=1}^{d} \theta_{j,q,l,k}^{N_{j,q,l,k}} P(\theta_{j,q,l,k}) d\theta_{j,q,l,k}. \quad (10)$$

Assuming Dirichlet distribution [13] with hyper-parameters $\alpha_S = \bigcup_{j=1}^{N}\{\alpha_j\}$ and $\alpha_j = \bigcup_{q=1,l=1,k=1}^{Q_j,L_j,d}\{\alpha_{j,q,l,k}\}$ as the prior distribution on the global and local parameters, then:

$$P(\theta_j|S) = Dir(\theta_j|\alpha_j) = \frac{\Gamma(\sum_{q=1}^{Q_j}\sum_{l=1}^{L_j}\sum_{k=1}^{d}\alpha_{j,q,l,k})}{\prod_{q=1}^{Q_j}\prod_{l=1}^{L_j}\prod_{k=1}^{d}\Gamma(\alpha_{j,q,l,k})} \prod_{q=1}^{Q_j}\prod_{l=1}^{L_j}\prod_{k=1}^{d}\theta_{j,q,l,k}^{\alpha_{j,q,l,k}-1}, \quad (11)$$

where $\Gamma(\cdot)$ is a Gamma function [13], which satisfies $\Gamma(x+1) = x\Gamma(x)$ and $\Gamma(1) = 1$. Using the Dirichlet priors of Equation (11), the high-dimensional integral in Equation (10) is solved to obtain a closed-form formula:

$$P(\mathbf{G}|S) = \prod_{j=1}^{N}\prod_{q=1}^{Q_j}\prod_{l=1}^{L_j}\frac{\Gamma(\alpha_{j,q,l})}{\Gamma(\alpha_{j,q,l}+N_{j,q,l})} \prod_{k=1}^{d}\frac{\Gamma(\alpha_{j,q,l,k}+N_{j,q,l,k})}{\Gamma(\alpha_{j,q,l,k})}, \quad (12)$$

where $N_{j,q,l} = \sum_{k=1}^{d} N_{j,q,l,k}$ and $\alpha_{j,q,l} = \sum_{k=1}^{d}\alpha_{j,q,l,k}$. Equation (12) corresponds to the following:

1) *Bayesian Dirichlet-equivalent* (BDe) metric of [14] when $\alpha_{j,q,l,k} = \alpha P(G_{t,j} = k, \mathbf{P}_{[1,r],j} = q, \mathbf{L}_{[1,r],j} = l|S)$, and $\alpha \geq 0$ is the equivalent sample size parameter.
2) *BDe uniform* (BDeu) metric of [15] when $\alpha_{j,q,l,k} = \frac{\alpha}{dL_jQ_j}$.
3) *K2 metric* of [16] when $\alpha_{j,q,l,k} = 1$.

In addition, the prior probability $P(S)$ of a given structure needs to be determined to complete the required information for the derivation of the MAP estimate, $\hat{S}_{\text{MAP}}$. Let S_j be the local structure at the j-th gene, we define the prior probability of S as:

$$P(S) = \prod_{j=1}^{N} P(S_j) = \prod_{j=1}^{N} e^{-q_j}. \quad (13)$$

where q_j is the number of parents of gene j, i.e. the number of edges linking to gene j. Substituting Equations (12) and (13) into Equation (5), we can have:

$$P(S|\mathbf{G}) \propto P(S)P(\mathbf{G}|S) = \prod_{j=1}^{N}\prod_{q=1}^{Q_j}\prod_{l=1}^{L_j} e^{-q_j}\frac{\Gamma(\alpha_{j,q,l})}{\Gamma(\alpha_{j,q,l}+N_{j,q,l})}\prod_{k=1}^{d}\frac{\Gamma(\alpha_{j,q,l,k}+N_{j,q,l,k})}{\Gamma(\alpha_{j,q,l,k})}. \quad (14)$$

Now we have the r-DBN learning criterion (i.e. Equation (14)). Note that this criterion is decomposable since it can be written as a product of local scores, each of which being a function of the j-th gene only.

If all the structures in the structure space are equally probable, the optimal structure $\hat{S}$ can be obtained from the *maximum likelihood* (ML) estimation, which is given by:

$$\hat{S}_{\text{ML}} = \arg\max_{S}\int_{\theta_S} P(\mathbf{G}|\theta_S)P(\theta_S|S)d\theta_S. \quad (15)$$

In the case of ML estimation, the parameters $\theta_{j,q,l,k}$ can be easily estimated as

$$\theta_{j,q,l,k} = \frac{N_{j,q,l,k}}{\sum_{k=1}^{d} N_{j,q,l,k}}, \quad (16)$$

and then the *Bayesian information criterion* (BIC) of [17] can be used to approximate the integral as:

$$BIC = \sum_{j=1}^{N}\sum_{q=1}^{Q_j}\sum_{l=1}^{L_j}\sum_{k=1}^{d} N_{j,q,l,k}\log\frac{N_{j,q,l,k}}{\sum_{k=1}^{d}N_{j,q,l,k}}. \quad (17)$$

The calculation of the BIC score is computationally faster than that of BDe, BDeu, and K2 scores, because it does not require priors over parameters. However, it is less accurate that those scores. The BIC score is more applicable for large data sets.

Finding the optimal structure of a BNs is known to be NP-hard [18] as the number of structures increases super-exponentially with the number of nodes. The learning problem, hence, becomes much harder as the order r increases.

3 Related Works of High-Order GRN Inference

Currently there are very few methods proposed to model time-delayed regulations from time-series expression data. In the following, we survey several approaches that are closely related to our work. For a comprehensive review of GRN modelling algorithms, please refer to [19].

In year 2005, Zou *et al.* [20] introduced an approach named DBN-ZC to transform the HO-DBN problem into FO-DBN problem, which can thus avoid learning structures from extremely large space of parameters. Their method is a local search-and-score learning method which includes three steps as follows. First, for each target gene T_g, DBN-ZC identifies those genes with either earlier or simultaneous expression changes compared to gene T_g, and then limits the potential regulators of gene T_g to those genes, which thus makes the subsequent structure learning much more efficient. Second, DBN-ZC estimates the time delays between two genes using the time difference between the initial gene expression change of a potential regulator and its target T_g. Third, this method aligns the time-series profiles of

the potential regulators and that of their target gene T_g, and then uses a search-and-score based FO-DBN learning algorithm to select the regulators with the highest log-marginal likelihood as the final set of regulators of gene T_g.

In [8], Xing *et al.* proposed a heuristic framework to infer r-DBNs from time-series expression data. Their method includes two steps as follows. First, mutual information is calculated for each pair of variables $G_{t-l,j}, G_{t,k}$ with a time lag l, for $1 \leq l \leq r$, $1 \leq t \leq T$ and $1 \leq j,k \leq N$; then initialize the transition network $S_{[t-r,t]}$ by including those pairs of genes with mutual information larger than a given threshold. Second, given the initial transition network $S_{[t-r,t]}$, this method utilizes genetic algorithm (GA) to find the structure which has the highest *maximum likelihood* or the *minimum description length* score. GA can search very large spaces when it is given a good representation of the GRN and appropriate genetic operators, because it is a population-based optimization method using implicit parallelism.

In [12], Chaturvedi and Rajapakse used prior biological knowledge from protein-protein interaction networks (PPINs) to limit the search space. They proposed a skip-chain model to learn time-delayed regulations. Based on the prior knowledge from given PPINs, their model infers edges between non-consecutive time-points (called *skip-edges*). The best skip-edges are selected using Viterbi approximation of DBNs, which is combined appropriately with BIC score which selects edges between two consecutive time-points (non skip-edges, called *linear-edges* in the paper). In [11], a variable-order DBN (VO-DBN) approach was introduced to automatically find delays of regulations between genes. The VO-DBN method does not fix the order r (or the maximum delay of regulation) as HO-DBNs does. Instead, it applies a *Markov chain Monte Carlo* (MCMC) approach to learn the optimal order r and the optimal structure. An appropriate acceptance mechanism is utilized to optimize both order and structure.

TDARACNE [21] extends the information-theory-based method, ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks), to reconstruct high-order GRNs. The expression of gene A at a time point could depend on the expression of gene B at several time point before. TimeDelay-ARACNE is a three-step algorithm. First, time points of initial expression changes of all genes are estimated. Second, candidate interaction between two genes are measured using mutual information where Markov random field is used as underling probabilistic model. Finally, edges of the produced network are pruned if the corresponding mutual information is below an appropriate threshold.

TDLASSO [22] is a two-step approach that constructs a time-delay GRN in a modified multivariate autoregression (MVAR) framework over continuous gene expression data. First, the order of regulation between two genes is selected using cross-correlation. Then, using the fixed time delays, LASSO (least absolute shrinkage and selection operator) is used to infer connections of the network.

4 MAX-MIN HIGH-ORDER DBNS

We present our HO-DBN method in this section. Our method, named Max-Min High-Order DBN (or MMHO-DBN for short) is an extension of the *Max-Min hill-climbing* (MMHC) heuristic which was originally devised for learning the structure of BNs from static data. Introduced in [4], as a fast, scalable, and reliable BN learning method, MMHC-BN is able to overcome the perceived limitations of the current state-of-the-art BN algorithms and which also exist in current HO-DBN algorithms. As a hybrid BN method, MMHC-BN incudes two steps: first it uses *constraint-based Bayesian learning* [23] to learn the skeleton (i.e. an undirected graph) of a BN; and then a *search-and-score Bayesian learning* is performed on the skeleton in order to orient its edges. By learning the skeleton first, MMHC-BN gains its reliability and accuracy, its efficiency, and more importantly, its ability to scale to distributions with thousands of variables. MMHC-BN does not require the user to estimate the number of parents for each variable as it discovers the maximum number of possible parents and children, of each variable during the skeleton learning phase by using a local learning technique. Compared to the hybrid *sparse candidate* (SC) algorithm [5] and the constraint-based methods such as PC algorithm [23], the local learning method of MMHC-BN is proven to be accurate and more efficient.

Our MMHO-DBN method includes two parts: first it obtains the skeleton of the network by applying the structure learning method on the discrete time-series data of gene expression; then it searches the best subset of parents for each node. These two parts correspond to the two MMHC-BN phases, which are appropriately modified for inferring dynamic Bayesian network from time-series gene expression data. In the modified MMHC-BN phases, the time-series gene expression data are considered as a discrete-time process with a joint probability distribution, where the expression levels of N genes at time t are represented as an N-dimensional vector of random variables taking discrete values.

Algorithm 1 *MMHO-DBN* Algorithm

Input: $\mathbf{g}_{T\times N} = (\mathbf{g}_1, \ldots, \mathbf{g}_T)^\intercal$: time-series data
r: maximum time delay
α: significance level
φ: cardinality limit for $\mathbf{P}_{[t-r,t],j}$
γ: cardinality limit for exhaustive search
Output: $\hat{S}$: the best DAG on the variables in $\mathbf{g}_{T\times N}$

{**Phase 1**: Restrict candidate parents}
for every variable $G_{t,j} \in \mathbf{G}$ **do**
$\mathbf{\Pi}_{[t-r,t],j} \leftarrow DMMP(G_{t,j}, \mathbf{g}_{T\times N}, r, \alpha)$; {/*see Algorithm 2*/}
end for

{**Phase 2**: Search for the best DAG $\hat{S}$}
for every variable $G_{t,j} \in \mathbf{G}$ **do**
$\mathbf{P}_{[t-r,t],j} \leftarrow Best\_Subset(G_{t,j}, \mathbf{g}_{T\times N}, r, \mathbf{\Pi}_{[t-r,t],j}, \varphi, \gamma)$; {/*see Algorithm 3*/}
end for
return the highest scoring DAG $\hat{S}$ found;

The process of MMHO-DBN is described in Algorithm 1. The Phase 1 of this algorithm shows the structure learning

method of MMHO-DBN. This method is modified from the local discovery method of [4], the *Max-Min parent and children* (MMPC) algorithm in order to compute the maximum possible parent set, $\mathbf{\Pi}_{[t-r,t],j}$, of each target variable $G_{t,j}$. Assuming that the *faithfulness assumption* [23] holds and that the statistical tests performed return reliable result, the original MMPC method is able to return the *maximum possible set of parent and children* PC(T) for each target variable T from a statistical non-stochastic data. The faithfulness assumption ensures that the PC(T) set is unique among all BNs faithful to the same distribution; a node may be T's parent in one BN and T's child in another BN, however PC(T) remains the same in both BNs. Using a constraint-based search algorithm, MMPC identifies the undirected connections between target genes and other genes, which leads to the skeleton of a BN. For r-DBNs, although the uniqueness of the PC(T) holds for r-DBNs faithful to the distribution $\mathcal{P}$, the edges will be oriented due to the temporal dependencies. Therefore, we only need to identify the *maximum possible set of parents* of target T and leave the children of T to be determined following the temporal dependencies. The maximum possible set of parents includes all probable parents, but excludes impossible ones, so that the search of actual set of parents are constrained on it, which makes structuring learning more efficient than exploring all combinations of genes. Our temporal variant of the MMPC algorithm is defined as the *dynamic Max-Min parent* (DMMP) algorithm which is used in the Phase 1 of Algorithm 1. The DMMP algorithm, described in Algorithm 2, aims to obtain the maximum possible set of parents, $\mathbf{\Pi}_{[t-r,t],j}$, of a target variable $G_{t,j}$.

The DMMP algorithm (see Algorithm 2) computes for each target variable $G_{t,j}$ the maximum possible set of parents, $\mathbf{\Pi}_{[t-r,t],j} \subseteq \mathbf{G}_{[t-r,t]}$, where $\mathbf{G}_{[t-r,t]}$ is the set of all variables within the last r previous time-points $t-l$ for $1 \leq l \leq r$. Starting from an empty $\mathbf{\Pi}_{[t-r,t],j}$, DMMP algorithm sequentially adds variables with "optimal" time delays, $\Pi_{\lambda,\mu} \in \mathbf{G}_{[t-r,t]}$, which maximize the minimum association with the given target $G_{t,j}$ relative to the current $\mathbf{\Pi}_{[t-r,t],j}$. This is Phase 1 of DMMP algorithm. It has been proven in [4] that the $\mathbf{\Pi}_{[t-r,t],j}$ obtained in this way does not contain false negatives but may contain false positives. These potential false positives would be removed in Phase 2 of DMMP algorithm. Following [4], we define the minimum association between a variable $G_{l,i} \in \mathbf{G}_{[t-r,t]}$ and the target $G_{t,j}$ relative to a subset $\mathbf{Z} \subseteq \mathbf{G}_{[t-r,t]}$ as:

$$MinAssoc(G_{l,i}; G_{t,j}|\mathbf{Z}) = \min_{\mathbf{C}\subseteq\mathbf{Z}} Assoc(G_{l,i}; G_{t,j}|\mathbf{C}). \quad (18)$$

To estimate the association between $G_{l,i}$ and $G_{t,j}$, independence tests $Ind(G_{l,i}; G_{t,j}|\mathbf{C})$ are performed and they will return *true* if $G_{l,i}$ and $G_{t,j}$ are conditionally independent given $\mathbf{C}$. Function $Assoc(G_{l,i}; G_{t,j}|\mathbf{C})$ estimates the strength of dependency between $G_{l,i}$ and $G_{t,j}$ given $\mathbf{C}$ such that $Assoc(G_{l,i}; G_{t,j}|\mathbf{C}) \geq 0$ with equality holding if and only if $Ind(G_{l,i}; G_{t,j}|\mathbf{C})$ is true. Here we use G^2 statistic as in [4] for the test of independence $Ind(G_{l,i}; G_{t,j}|\mathbf{C})$, using the same number of degree of freedom as in [4]. Assuming data are discrete and elements in a contingency table (for conditional independence test) follows a multinominal distribution, G^2 statistic is asymptotic to a χ^2 distribution with appropriate degree of freedom [23]. Gene expression data have to be discretized in order to apply it. The null hypothesis is that the two variables are conditionally independent. And we reject the null hypothesis if the p-value from the G^2 statistic is lower than some significant level α (here we set $\alpha = 0.05$). Therefore, the function of association can be defined as:

$$Assoc(G_{l,i}; G_{t,j}|\mathbf{C}) = \begin{cases} 0 & \text{if } p\text{-value} \geq \alpha \\ \alpha - p\text{-value} & \text{otherwise} \end{cases}. \quad (19)$$

Algorithm 2 *DMMP* Algorithm

Input: $G_{t,j}$: target node
$\mathbf{g}_{T\times N} = (\mathbf{g}_1, \ldots, \mathbf{g}_T)^\intercal$: time-series data
r: maximum time delay
α: significance level
Output: $\mathbf{\Pi}_{[t-r,t],j}$: maximum possible set of parents of $G_{t,j}$

{**Phase 1**: Compute a candidate $\mathbf{\Pi}_{[1,r],j}$}
$\mathbf{\Pi}_{[t-r,t],j} \leftarrow \emptyset$;
repeat
  $\varphi \leftarrow \max_{G_{l,i}\in\mathbf{G}_{[t-r,t]}} MinAssoc(G_{l,i}; G_{t,j}|\mathbf{\Pi}_{[t-r,t],j})$;
  $\Pi_{\lambda,\mu} \leftarrow \arg\max_{G_{l,i}\in\mathbf{G}_{[t-r,t]}} MinAssoc(G_{l,i}; G_{t,j}|\mathbf{\Pi}_{[t-r,t],j})$;
  if $\varphi \neq 0$ **then**
    $\mathbf{\Pi}_{[t-r,t],j} \leftarrow \mathbf{\Pi}_{[t-r,t],j} \cup \{\Pi_{\lambda,\mu}\}$;
  end if
until $\mathbf{\Pi}_{[t-r,t],j}$ does not change

{**Phase 2**: Remove false positives}
for all $\Pi_{\lambda,\mu} \in \mathbf{\Pi}_{[t-r,t],j}$ **do**
  if $\exists \mathbf{\Gamma} \subseteq \mathbf{\Pi}_{[t-r,t],j}$ s.t. $Ind(\Pi_{\lambda,\mu}; G_{t,j}|\mathbf{\Gamma})$ **then**
    $\mathbf{\Pi}_{[t-r,t],j} \leftarrow \mathbf{\Pi}_{[t-r,t],j} \setminus \{\Pi_{\lambda,\mu}\}$;
  end if
end for
return $\mathbf{\Pi}_{[t-r,t],j}$;

There may be a problem about the degree of freedom in calculating G^2 statistic. Because time-series gene expression data are sparse (here "sparse" means the number of genes is much greater than the number of time points) and the number of combinations goes exponentially as the number of parents increases, we may have some zero cells in the contingency table when conducting conditional independence test. The degree of freedom may become negative when the number of zero-cells is large. For example, suppose the expression data are discretized into two states; when testing $Ind(A, B|C)$, the contingency table would have 8 cells, and the degree of freedom is $1 \times 1 \times 2 = 2$; if the number of zeros-cells is larger than 2, the degree of freedom would be negative. The negative degree of freedom would cause a computational disaster. So when extending MMHC algorithm to a high-order version and applying it on sparse gene expression time-series data, we need to employ a smooth method to cope with the problem of negative degree of freedom. Inspired by the calculation of BDeu score [14], we propose a smooth method by adding a constant number to each cell in the contingency table. This constant number is called *equivalent sample size* (ESS) or *pseudocount*. We shall

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2015.2474409, IEEE/ACM Transactions on Computational Biology and Bioinformatics

investigate how the algorithm is affected by ESS in Section 5.

Algorithm 3 *Best_Subset* Algorithm

Input: $G_{t,j}$: target node
$\mathbf{g}_{T\times N} = (\mathbf{g}_1, \ldots, \mathbf{g}_T)^{\mathrm{T}}$: time-series data
r: maximum time delay
$\mathbf{\Pi}_{[t-r,t],j}$: maximum possible parent set of $G_{t,j}$
φ: cardinality limit for $\mathbf{P}_{[t-r,t],j}$
γ: cardinality limit for exhaustive search
Output: $\mathbf{P}_{[t-r,t],j}$: best subset of parents of $G_{t,j}$

```
if |Π_{[t−r,t],j}| ≤ γ then
  {Exhaustive search}
  P_{[t−r,t],j} ← arg max_{P^(φ) ⊆ Π_{[t−r,t],j}} Score(P^(φ), G_{t,j});
else
  {Heuristic search}
  P_{[t−r,t],j} ← Heuristic(G_{t,j}, Π_{[t−r,t],j}, Score, φ);
end if
return P_{[t−r,t],j};
```

After the Phase 1 of MMHO-DBN, we have determined the set of maximum possible parents $\mathbf{\Pi}_{[t-r,t],j}$ for each target variable $G_{t,j}$ ($1 \leq j \leq N$). Next we will perform a search-and-score strategy to find the best subset of parents $\mathbf{P}_{[t-r,r],j} \subseteq \mathbf{\Pi}_{[t-r,t],j}$ maximizing a score function (e.g., BDe, BDeu, K2, BIC, etc). Starting from an empty DAG, edges would be added from $\Pi_{\lambda,\mu}$ to $G_{t,j}$ and $\Pi_{\lambda,\mu} \in \mathbf{\Pi}_{[t-r,t],j}$; that is, the parents of target $G_{t,j}$ are constrained to the set of possible parents $\mathbf{\Pi}_{[t-r,t],j}$ only. The search algorithm of MMHO-DBN is described in Algorithm 3. The *Score* function in Algorithm 3 is the *Bayesian Dirichlet-equivalent uniform* (BDeu) metric. The parameter γ is the maximum allowed cardinality of $\mathbf{\Pi}_{[t-r,t],j}$ below which we can perform an exhaustive search, otherwise we perform a heuristic search for the best subset. In our implementation, we use a greedy algorithm to perform the heuristic search. Another parameter φ is the maximum cardinality of $\mathbf{P}_{[t-r,r],j}$; that is, we only search the subsets with no more than φ members which maximize the BDeu score. By doing this, we can limit the size of search space for the sake of computational efficiency. Here the parameter φ is set to 3.

In order to analyze the complexity of MMHO-DBN, we denote the maximum number of possible parents of all genes as M, the maximum time delay as r, the number of genes as N, the number of time points as T, the number of discrete states as d. Restricted by sample size and reliability of G^2 statistic, M should fulfil $M \ll \min(N, T)$. The time complexities of phase 1 and 2 of the DMMP algorithm are approximately $O(rNd^2(1+d)^M)$ and $O(Md^2(1+d)^{M-1})$, respectively. Algorithm 3 takes $O(1+d)^M$. Therefore, the time complexity of the MMHO-DBN algorithm is $O(rN^2d^2(1+d)^M)$.

5 Computational Experiments

In this section, we evaluate the performance of MMHO-DBN on simulated data and real gene expression time-series data, respectively, whose ground-truth networks are known. We compared MMHO-DBN with DBmcmc (a first-order DBN) [24], DBN-ZC (a high-order DBN) [20], TDARACNE (a high-order method based on information theory) [21], and TDLASSO (a high-order method using sparse regression) [22]. We set the default maximum fan-in of each gene to 3 in MMHO-DBN, DBmcmc, and DBN-ZC for fair comparison.

TDARACNE and TDLASSO worked on continuous values. MMHO-DBN, DBmcmc, and DBN-ZC ran on discrete data. We have tried several methods for data discretization: equal-width binning, equal frequency binning, and Gaussian method. Equal-width and frequency binning are two standard methods for data discretization. The Gaussian method is described in [25]. Results shown in this paper are based on discrete data obtained by equal-width binning. Results on the second synthetic data and all real data discretized by the other methods are quite similar. The expression values were discretized into three states for MMHODBN, DBmcmc, and DBN-ZC. Discretization into two states leads to performance very similar to that of using three states (data not shown), thus we only report the performance of using three states below.

We use sensitivity, precision, and *F*-measure to evaluate the performance of the methods. The presence of an actual directed connection is defined as positive, and an absent actual edge is negative. The numbers of true positives, true negatives, false positives, and false negatives are denoted as TP, TN, FP, and FN, respectively. Sensitivity (also called *recall*) is defined as $\frac{TP}{TP+FN}$. Precision (also called *positive predictive value*) is defined as $\frac{TP}{TP+FP}$. *F*-measure is the harmonic mean of precision and recall: $2 * \frac{presion*sensitivity}{precision+sensitivity}$.

5.1 On Simulated Data

In our first experiment, we test whether MMHO-DBN can identify regulators of different time-delays. We designed a network of 8 nodes as shown in Figure 3. This network is composed of regulators of different time-delays and regulators of the same time-delay. The time delays are given along the directed connections. The parameters, that is the (conditional) probabilities, are also given in the tables of Figure 3.

Using this network, we generated a discrete data set with two states and a continuous data set, respectively, with 200 equally distributed time points. In our MMHO-DBN, we set the maximum time-delay to 2. The significance level was $\alpha = 0.05$ in the conditional independence test. We used BDeu metric to score a structure. The equivalent sample size is set to 1 for both conditional independence test and BDeu score.

On the first simulated data set, the predicted networks by MMHO-DBN, DBmcmc, and DBN-ZC are demonstrated in Figures 4a, 4b, and 4c, respectively. Their performances, including precisions, sensitivities, and *F*-measures, are compared in Table 1. From Figure 4, first of all, we can see that our MMHO-DBN could identify all the existing connections with correct time-delays, whereas DBmcmc could only predict the connections of 1 time-delay correctly. DBN-ZC method failed to identify all existing connections, which convinces us that grouping regulators according to time-delays may not be a wise choice in some cases. DBN-ZC only searches among the subsets of the potential regulators with the *same* time-delays. As a high-order DBN, our

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2015.2474409, IEEE/ACM Transactions on Computational Biology and Bioinformatics

approach runs very fast. It can be seen in Table 1 that, MMHO-DBN took only 34 seconds, while DBmcmc took 1878 seconds.

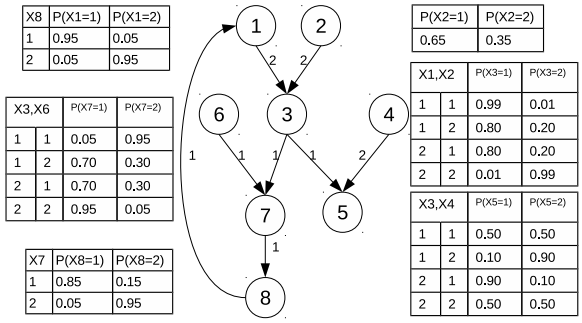
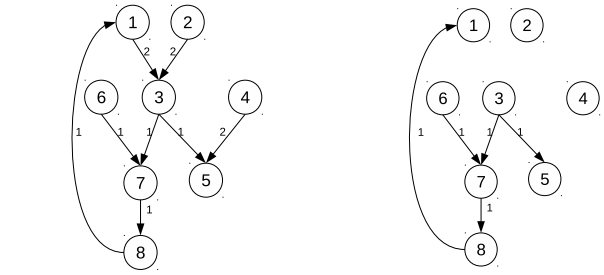
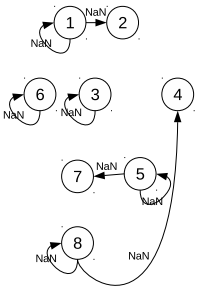


Fig. 3: True network from which simulated data were sampled.



(a) Predicted by MMHO-DBN. (b) Predicted by DBmcmc.



(c) Predicted by DBN-ZC. "NaN" means the time-lag information is unavailable in the implementation provided by the authors of [20].

Fig. 4: Predicted networks using simulated discrete data.

TABLE 1: Comparison on simulated discrete data

Method	Precision	Sensitivity	*F*-measure	Time (s.)
MMHO-DBN	1	1	1	34
DBmcmc	1	0.6250	0.7692	1878
DBN-ZC	0	0	NA	0.355

Since TDARACNE and TDLASSO only work on continuous data, we generated a data set with continuous values using the simulation script of [22]. The data were discretized into 3 states for MMHO-DBN, DBmcmc, and DBN-ZC. The results of these methods are provided in Table 2. We can see that MMHO-DBN achieved the highest sensitivity (1.0000) and *F*-measure (0.8889). TDLASSO had slightly lower precision (0.6667) and *F*-measure (0.8000). TDARACNE obtained a higher precision than TDLASSO, but its sensitivity and *F*-measure were lower. DBmcmc achieved the highest precision, but its sensitivity and *F*-measure are lower than MMHO-DBN and TDLASSO. DBN-ZC was ranked in the last position.

TABLE 2: Comparison on simulated continuous data

Method	Precision	Sensitivity	*F*-measure	Time (s.)
MMHO-DBN	0.8000	1.0000	0.8889	61.86
TDARACNE	0.7143	0.6250	0.6667	35.51
TDLASSO	0.6667	1.0000	0.8000	0.8268
DBmcmc	0.8333	0.6250	0.7143	23
DBN-ZC	0.2222	0.2500	0.2353	0.2207

5.1.1 Sensitivity Analysis

In the MMHO-DBN model, there are a few parameters. The first parameter is the "equivalent sample size", which is used in the calculations of conditional independence test and BDeu score, respectively. A larger equivalent sample size would make the corresponding distribution smoother. The second parameter is the "maximum time-delay" (that is r in Algorithm 1), which specifies the largest time delay of the regulations considered when inferring the regulatory relations among genes. The larger the maximum time-delay is, the higher the order of MMHO-DBN would be, and gene regulations with longer range could be detected. On the other hand, a larger maximum time-delay will result in a smaller sample size. The third parameter is the "maximum number of parents" for each gene (that is φ in Algorithm 3), i.e. the maximum fan-in of each node. A larger maximum number of parents would allow more multiple regulations being discovered, although it may require more running time, and the number of the corresponding parameters would increase exponentially. Generally, we set the equivalent sample sizes to 1, maximum time-delay to 2, and maximum parents to 3. Here, we investigate how the parameters affect the performance of MMHO-DBN with respect to the *F*-measure on the simulated discrete data. This can help us set the parameters properly for future applications.

We first investigated how the parameter of "equivalent sample size" affects the performance of MMHO-DBN. The equivalent sample size is used in the phases of obtaining candidate parents and searching parents for each node. For the convenience of narration, we denote *equivSize1* and *equivSize2* as the equivalent sample sizes in the two phases, respectively. We fixed the maximum time-delay to 2 and the maximum number of parents to 3. MMHO-DBN was run with *equivSize1* and *equivSize2* ranging from 1 to 15. The variation of the performance of MMHO-DBN is shown in the top sub-plot of Figure 5. We can see that, as *equivSize1* increases, the performance of MMHO-DBN does not change. This implies that the performance of the algorithm is insensitive to the value of *equivSize1* on the simulated data. However, a larger *equivSize2* is likely to decrease the performance of MMHO-DBN.

Second, to examine how the parameter of "maximum time-delay" affects the performance of MMHO-DBN, we ran MMHO-DBN with the maximum time-delay from 1 to 100. Meanwhile, we fixed both *equivSize1* and *equivSize2* to 1, and the maximum number of parents to 3. The result is shown at middle of Figure 5. The best performance was obtained when the maximum time-delay was greater than 1. As the maximum time-delay increases, the performance gets worse. Hence, the maximum time-delay corresponding to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2015.2474409, IEEE/ACM Transactions on Computational Biology and Bioinformatics

the best performance is consistent with the actual maximum time-delays (see Figure 3).

Similarly, to study the effect of "maximum number of parents" on the performance of MMHO-DBN, we ran MMHO-DBN with the maximum number of parents changing from 1 to 5. Meanwhile, we fixed both *equivSize1* and *equivSize2* to 1, and the maximum time-delay to 2. The result is shown at the bottom sub-figure of Figure 5. It shows that the algorithm is robust to this parameter on the simulated data.

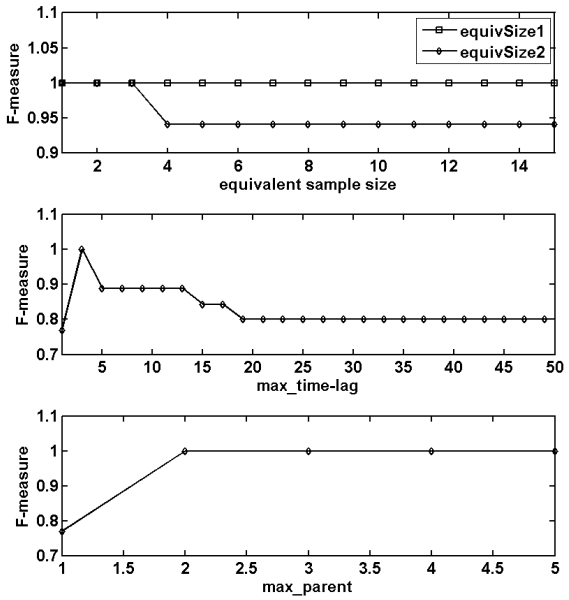


Fig. 5: Effect of parameters, including equivalent sample size (top), maximum time-delay (middle), and maximum number of parents (bottom), on the performance of MMHO-DBN on the simulated discrete data.

5.2 On Real Data

We validated our method on three real-life benchmark networks of yeast. The first benchmark network is a five-gene network for *in vivo* reverse-engineering and modeling assessment (IRMA) [26]. This is one of the first attempts to build a real-life gene regulatory network with accurately known gene interactions and reference data of gene expression. In their work, the authors constructed a network of five genes in *Saccharomyces cerevisiae*. The regulatory interactions among these five genes were carefully designed and this network was negligibly affected by endogenous genes. The network can be triggered by galactose, and it would be "switched on" when the cell culture is shifted from glucose to galactose, and "switched off" if the culture is shifted from galactose to glucose. Two sets of time-series gene expression data were measured after the perturbation: one named "switch-on data set" with 16 equally distributed time points (or *Yeast5on* for short), and another named "switch-off data set" with 21 equally distributed time points (or *Yeast5off* for short). The second benchmark network is a nine-gene network of *S. cerevisiae* related to yeast cell cycle, identified by Simon *et al.* [27]. Each of the nine genes is a cell-cycle transcription activator, and this network shows how the transcription activators functioning at one stage of the yeast cell cycle regulate the transcription activators active at the next stage. The gene expression data of yeast cell cycle were obtained from [28], which is a time-series data set sampled at 24 equally distributed time points. This data set is abbreviated as *Yeast9* in our study. A larger network with 20 genes (named *Yeast20*) was extracted from a yeast network in GeneNetWeaver (GNW, version 3.1.2b) [29]. The yeast network has 4441 nodes and 12873 edges. *Yeast20* was extracted using the "Subnetwork Extraction" in GNW with "Seed" as "From strongly connected component" and other default parameters. Then, we obtained the gene expression data of these 20 genes from the Spellman data set [28]. Note that we only used the "cdc15" samples, which were sampled over 24 equally spaced time points. When using MMHO-DBN on all data above, we set the maximum time-delay upto 8, the maximum number of allowed parents to 3, the equivalent sample size to 1, and the significance level to 0.05.

To show the ability of MMHO-DBN in learning regulations with delayed interactions, we here compare the performance of MMHO-DBN, DBmcmc (a first-order DBN), and DBN-ZC (a high-order DBN) on *Yeast5on*, *Yeast5off*, and *Yeast9*, as shown in Table 3. The actual networks and the predicted networks by different methods are illustrated in Figures 6 and 7. First, MMHO-DBN had a better performance than DBmcmc on all real-life data sets, probably because MMHO-DBN considers simultaneously gene regulations with longer time-delays, while DBmcmc only takes into account genetic interactions with one time delay. Second, MMHO-DBN also performed better than DBN-ZC. We also tested the first-order MMHO-DBN (by setting the parameter maximum time-delay to 1) on the three real data sets and compared with DBmcmc that is a first-order model. The results are shown in Table 4. We can see that even the first-order MMHO-DBN had competitive performance compared to DBmcmc. Furthermore, the results of the high-order MMHO-DBN were better than that of the first-order MMHO-DBN, which convinces us that it is necessary to consider high-order time delays in the reconstruction of GRNs.

The continuous models, TDARACNE and TDLASSO, were also tested on all data sets. From Table 3, it can be seen that MMHO-DBN obtained the best precision, sensitivity, and *F*-measure on all data sets except *Yeast5on* where TDARACNE had a higher sensitivity. The performances of TDARACNE and TDLASSO were generally inferior to MMHO-DBN. As a first-order method, the performance of DBmcmc was not as good as the high-order methods.

We assess computing times using the results on *Yeast20*. Among the three competitive high-order methods (MMHO-DBN, TDARACNE, and TDLASSO), the regression-based method, TDLASSO, was the most efficient algorithm which took only less than one second on *Yeast20*. The information-theory-based algorithm, TDARACNE, was the slowest one on *Yeast20*. Our DBN-based approach, MMHO-DBN, had intermediate computing performance. Possible improvements of the efficiency of MMHO-DBN is discussed in Section 6.

It is difficult to precisely validate predicted time lags of a learned GRN from real data. According to [27], genes SWI4 and MBP1 are both active in phase late G1 of a cell cycle;

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2015.2474409, IEEE/ACM Transactions on Computational Biology and Bioinformatics

TABLE 3: Comparison on real-life data

Data	Method	Precision	Sensitivity	*F*-measure	Time (s.)
Yeast5on	MMHO-DBN	1.0000	0.5000	0.6667	1
	TDARACNE	0.7142	0.6250	0.6667	1.93
	TDLASSO	0.4000	0.2500	0.3077	0.08
	DBmcmc	0.5000	0.2500	0.3333	108
	DBN-ZC	0.6000	0.3750	0.4615	0.03
Yeast5off	MMHO-DBN	0.6667	0.2500	0.3636	2
	TDARACNE	0.5000	0.1250	0.2000	1.2
	TDLASSO	0.2500	0.1250	0.1667	0.26
	DBmcmc	0.1700	0.1200	0.1407	109
	DBN-ZC	0	0	NA	0.1
Yeast9	MMHO-DBN	0.5000	0.1765	0.2609	28
	TDARACNE	0.2308	0.1765	0.2000	24.95
	TDLASSO	0.2500	0.2353	0.2424	0.09
	DBmcmc	0.2100	0.1400	0.1680	140
	DBN-ZC	0.1111	0.0588	0.0769	0.2
Yeast20	MMHO-DBN	0.3158	0.1333	0.1875	29.8305
	TDARACNE	0.1935	0.1333	0.1579	140.84
	TDLASSO	0.1087	0.1111	0.1099	0.5148
	DBmcmc	0.1579	0.0667	0.0938	33
	DBN-ZC	0	0	NA	15.6466

TABLE 4: Comparison of MMHO-DBN (first-order) and DBmcmc on real-life data

Data	Method	Precision	Sensitivity	*F*-measure
Yeast5on	MMHO-DBN	1.0000	0.2500	0.4000
	DBmcmc	0.5000	0.2500	0.3333
Yeast5off	MMHO-DBN	0.4000	0.2500	0.3077
	DBmcmc	0.1700	0.1200	0.1407
Yeast9	MMHO-DBN	0.2857	0.1176	0.1667
	DBmcmc	0.2100	0.1400	0.1680

NDD1 is active in phase G2, and MCM1 is active in phase G2 or M; SWI5 is active in phase M; CLN3 is active in phase G1. Such information can serve as evidence for the delayed regulations between those genes. We have compared the time delays predicted by MMHO-DBN with those in real cases. The results are shown in Table 5. Note that the predicted time delays are measured in the numbers of time lags. As seen, the predicted time delays are consistent with real cases.

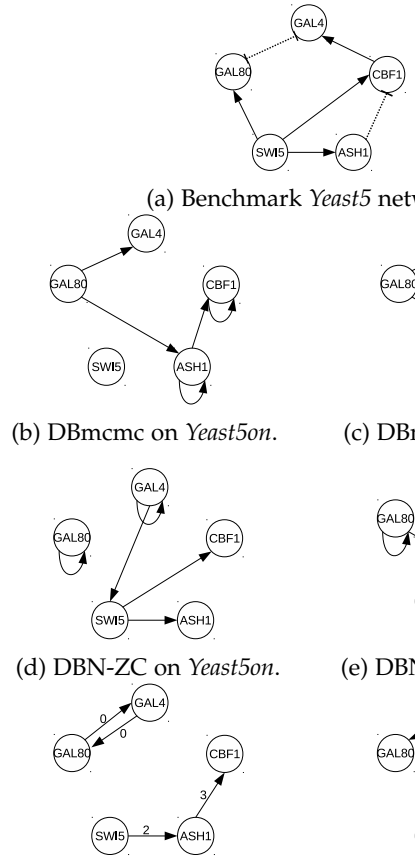
TABLE 5: Validating time-delays on *Yeast9*

TP Regulation	Phases	Predicted Time Lags	Remark
SWI4→MBP1	Late G1→late G1	0	-
SWI4→NDD1	Late G1→G2	6	-
MCM1→CLN3	G2/M→G1	8	-
CLN3→SWI5	G1→M	6	opposite direction

6 Conclusion

In this paper, we proposed a high-order dynamic Bayesian network learning method for reconstructing gene regulatory networks. This is a constraint-and-scoring method. In the algorithm, we also proposed to use equivalent sample size to overcome the potential computational problem when testing the conditional independence. Our experiments on simulated and real-life data show that our method can identify regulators of different time-delays with improved accuracy and has an intermediate computing performance. We also conducted sensitivity analysis on the parameters of our method, which provided suggestions for appropriate parameters setting. The algorithm was implemented in MATLAB (also workable in Octave), and is publicly available at https://sites.google.com/site/pgmtool. The MMHO-DBN algorithm is quite general, thus can be used on any noisy temporal data in other domains.

Several studies need to be done in the near future. First,



(a) Benchmark *Yeast5* network.

(b) DBmcmc on *Yeast5on*.

(c) DBmcmc on *Yeast5off*.

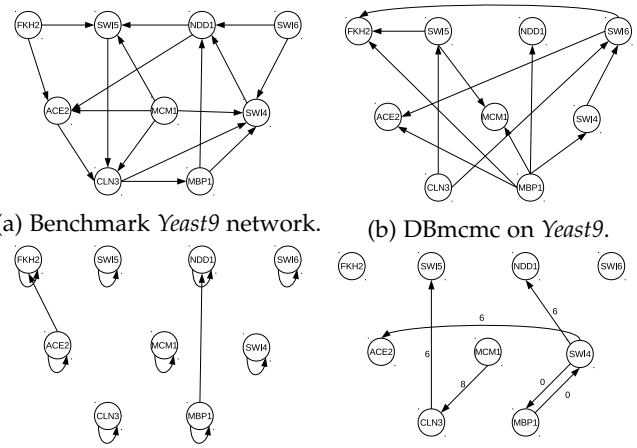
(d) DBN-ZC on *Yeast5on*.

(e) DBN-ZC on *Yeast5off*.

(f) MMHO-DBN on *Yeast5on*.

(g) MMHO-DBN on *Yeast5off*.

Fig. 6: Benchmark *Yeast5* networks and the predicted networks using *Yeast5* data. The dotted T lines in Figure 6a, indicate inhibitions. However, we do not infer the qualitative influences in this paper. Applying qualitative networks is our on-going research.



(a) Benchmark *Yeast9* network.

(b) DBmcmc on *Yeast9*.

(c) DBN-ZC on *Yeast9*.

(d) MMHO-DBN on *Yeast9*.

Fig. 7: Benchmark *Yeast9* network and the predicted networks using *Yeast9* data.

more actual regulatory networks are required to evaluate the performance of our method. Particularly, it will be very interesting to assess our method on large-scale networks, if available. Second, the candidate-parent searching and best-subset searching in our method (Algorithm 1) can be easily adapted to a parallel algorithm, so that computing time can be dramatically reduced on large networks in a cluster platform. Fast methods (e.g. the cross-correlation method used in [22]) that estimate the optimal time lags should be applied to further speed up MMHO-DBN. From the time complexity of MMHO-DBN, we can see that its computing time increases exponentially as the increase of the maximum number of possible parents. Thus, it is necessary to investigate methods that limit this parameter. In MMHO-DBN, the maximum set is defined based on association score, while the exact set of parents is usually defined by a score derived from an MAP estimate. When the former is equivalent to the latter is unknown for us so far. Once found, it would be a big stride in Bayesian network learning. Third, although our algorithm was evaluated on microarray time-series data, it can be applied on RNA-seq time-series data [30] (if available) which is much more precise than microarray data. Furthermore, from Table 3, we can see that the results of all methods (even the best) are still unsatisfactory, which reflects the difficulty of accurately reconstructing GRNs purely from gene expression data. Thus, we plan to combine knowledge from other data, especially from the next-generation sequencing data.

ACKNOWLEDGMENTS

We thank anonymous reviewers for constructive comments. This work was supported by Ontario Graduate Scholarship [2011-2012 to Y.L.]; Natural Sciences and Engineering Research Council of Canada Grant [RGPIN228117-2011 to A.N.]; and Singapore Ministry of Education Academic Research Tier 2 Grant [ARC 09/10 to J.Z.].

REFERENCES

[1] S. Das, D. Caragea, S. M. Welch, and W. H. Hsu, *Handbook of Research on Computational Methodologies in Gene Regulatory Networks*. IGI Global, 2009.

[2] L. Chen, R.-S. Wang, and X.-S. Zhang, *Biomolecular Networks: Methods and Applications in Systems Biology*. John Wiley & Sons, 2009.

[3] R. Daly, K. Edwards, J. O'Neill, S. Aitken, A. Millar, and M. Girolami, "Using higher-order dynamic Bayesian networks to model periodic data from the circadian clock of Arabidopsis thaliana," *LNCS*, vol. 5780, pp. 67–78, 2009.

[4] I. Tsamardinos, L. Brown, and C. Aliferis, "The max-min hill-climbing Bayesian network structure learning algorithm," *Machine Learning*, vol. 65, pp. 31–78, 2006.

[5] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *Journal of Computational Biology*, vol. 7, pp. 601–620, 2000.

[6] N. Friedman, K. Murphy, and S. Russell, "Learning the structure of dynamic probabilistic networks," in *Proc. the 14th Annual Conference on Uncertainty in Artificial Intelligence*, 1998, pp. 139–147.

[7] Z. Li, P. Li, A. Krishnan, and J. Liu, "Large-scale dynamic gene regulatory network inference combining differential equation models with local dynamics Bayesian network analysis," *Bioinformatics*, vol. 21, no. 19, pp. 2686–2691, 2011.

[8] X. Xing and D. Wu, "Modeling multiple time units delayed gene regulatory network using dynamic Bayesian network," in *Proc. ICDMW*, IEEE. Hong Kong: IEEE Press, 2006, pp. 190–195.

[9] Y. Li and A. Ngom, "The max-min high-order dynamic Bayesian network learning for identifying gene regulatory networks from time-series microarray data," in *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, IEEE. Piscataway, NJ: IEEE Press, Apr. 2013, pp. 83–90.

[10] K. Murphy, "Dynamic Bayesian networks: Representation, inference and learning," Ph.D. dissertation, Computer Science Division, Berkley, CA, 2002.

[11] J. Rajapakse and I. Chaturvedi, "Gene regulatory networks with variable-order dynamic Bayesian networks," in *Proc. IJCNN*, IEEE. Barcelona, Spain: IEEE Press, 2010, pp. 1–5.

[12] I. Chaturvedi and J. Rajapakse, "Building gene networks with time-delayed regulations," *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2133 – 2137, 2010.

[13] D. Heckerman, "A tutorial on learning with Bayesian networks," in *Learning in Graphical Models*, ser. Adaptive Computation and Machine Learning series, M. Jordan, Ed. Cambridge, MA: MIT, 1998, ch. 11, pp. 301–354.

[14] D. Heckerman, D. Geiger, and D. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Machine Learning*, vol. 20, pp. 197–243, 1995.

[15] W. Buntine, "Theory refinement on Bayesian networks," in *Proc. UAI*. Morgan Kaufmann, 1991, pp. 52–60.

[16] G. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, pp. 309–347, 1992.

[17] G. Schwarz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, pp. 461–464, 1978.

[18] D. Chickering, "Learning Bayesian networks is NP-complete," in *Learning from Data: AI and Statistics V*, ser. Lecture Notes in Statistics, D. Frisher and H.-J. Lenz, Eds. Springer, 1996, ch. 12, pp. 121–130.

[19] Y. Kim, Y. Han, S. Choi, and D. Hwang, "Inference of dynamic networks using time-course data," *Briefings in Bioinformatics*, vol. 15, no. 2, pp. 212–228, 2014.

[20] M. Zou and S. Conzen, "A new dynamic Bayesian network (DBN) approach for indentifying gene regulatory networks from time course microarray data," *Bioinformatics*, vol. 21, no. 1, pp. 71–79, 2005.

[21] P. Zoppoli, S. Morganella, and M. Ceccarelli, "TimeDelay-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach," *BMC Bioinformatics*, vol. 11, p. 154, 2010.

[22] P. Mundra, J. Zheng, M. Niranjan, R. Welsch, and J. Rajapakse, "Inferring time-delayed gene regulatory networks using cross-correlation and sparse regression," in *Bioinformatics Research and Applications*, vol. LNCS7875. Springer, 2013, pp. 64–75.

[23] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, 2nd ed. Cambridge, MA: MIT, 2001.

[24] D. Husmeier, "Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks," *Bioinformatics*, vol. 19, no. 17, pp. 2271–2282, 2003.

[25] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *Journal of Bioinformatics and Computational Biology*, vol. 3, no. 2, pp. 185–205, 2005.

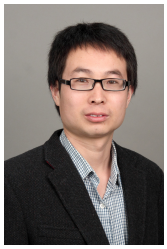
[26] I. Cantone, L. Marucci, F. Iorio, M. Ricci, V. Belcastro, M. Bansal, S. Santini, M. di Bernardo, D. di Bernardo, and M. Cosma, "A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches," *Cell*, vol. 137, pp. 172–181, 2009.

[27] I. Simon, J. Barnett, N. Hannett, C. Harbison, N. Rinaldi, T. Volkert, J. Wyrick, J. Zeitlinger, D. Gifford, and T. Jaakkola, "Serial regulation of transcriptional regulators in the yeast cell cycle," *Cell*, vol. 106, pp. 697–708, 2001.

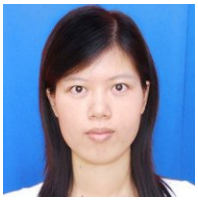
[28] P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization," *Molecular Biology of The Cell*, vol. 9, pp. 3273–3297, 1998.

[29] T. Schaffter, D. Marbach, and D. Floreano, "GeneNetWeaver: In silico benchmark generation and performance profiling of network inference methods," *Bioinformatics*, vol. 27, no. 16, pp. 2263–2270, 2011.

[30] Z. Wang, M. Gerstein, and M. Snyder, "RNA-Seq: A revolutionary tool for transcriptomics," *Nature Reviews Genetics*, vol. 10, pp. 57–63, 2009.



Yifeng Li Dr. Yifeng Li received his Bachelor's and Master's Degrees from Shandong Institute of Light Industry (currently the Qilu University of Technology), China, in 2006, and 2009, respectively. He obtained his Ph.D. from the School of Computer Science, University of Windsor, Canada, in 2013. At present, he is a post-doctoral research fellow at the Centre for Molecular Medicine and Therapeutics, University of British Columbia, Canada. His research interests include sparse machine learning models, large-scale numerical optimization, high dimensional data analysis in bioinformatics, and cis-regulatory elements identification in the human genome. Yifeng is a member of IEEE and ISCB.



Haifen Chen Haifen Chen is currently a PhD student from School of Computer Engineering, Nanyang Technological University (NTU), Singapore. She joined the Bioinformatics Research Center of NTU in 2011. Haifen holds Master's and Bachelor's degrees in Computer Science and Technology, both from Harbin Institute of Technology (HIT), China. Her Master's research focused on biometrics and machine learning. She was awarded the Outstanding Graduate of HIT in 2010. Her research interests lie in applying machine-learning techniques and statistical models to the area of gene regulatory network (GRN) inference, epigenetic regulation of gene expression, modelling of stem cell development and reprogramming.

Jie Zheng Dr. Jie Zheng is a tenure-track Assistant Professor at School of Computer Engineering, Nanyang Technological University (NTU), Singapore, and an adjunct faculty (Senior Research Scientist) with Genome Institute of Singapore (GIS), A*STAR. He received Ph.D. in 2006 from the University of California, Riverside, USA and B.Eng (first class honor) in 2000 from Zhejiang University in China, both in Computer Science. Before joining NTU in Feb. 2011, he did postdoc training at the National Center for Biotechnology Information (NCBI), National Library of Medicine (NLM), National Institutes of Health (NIH), USA. His research interests are Bioinformatics, Computational Systems Biology and Genomics, aiming to develop novel algorithms and in silico models to help answer Biomedical questions (such as the mechanisms of cancer). Although trained in Computer Science, Dr. Zheng maintains active and long-standing collaborations with Life Scientists.

Alioune Ngom Dr. Alioune Ngom received his PhD degree in 1998 at the University of Ottawa, and he is currently a Professor at the School of Computer Science, University of Windsor, Ontario, Canada. Prior to joining Windsor in 2000, he has held an Assistant Professor position at Lakehead University, Ontario, and cofounded Mitomics Inc (formerly, Genesis Genomics Inc.; a biotechnology company specializing in the analysis of the mitochondrial genome and its associated diseases). His main research interests include but are not limited to computational intelligence and machine learning methods, and their applications in the fields of computational biology and bioinformatics. His current research includes gene regulatory network reconstruction, protein complex identification, sparse representation learning, network clustering, and biomarker selection. He is a member of the IEEE.