The Max-Min High-Order Dynamic Bayesian Network Learning for Identifying Gene Regulatory Networks from Time-Series Microarray Data

Yifeng Li and Alioune Ngom
School of Computer Science, University of Windsor, Windsor, Ontario, Canada
\{li11112c, angom\}@uwindsor.ca

Abstract—We propose a new high-order dynamic Bayesian network (HO-DBN) learning approach, called Max-Min High-Order DBN (MMHO-DBN), for discrete time-series data. MMHO-DBN explicitly models the time lags between parents and target in an efficient manner. It extends the Max-Min Hill-Climbing Bayesian network (MMHC-BN) technique which was originally devised for learning a BN's structure from static data. Both Max-Min approaches are hybrid local learning methods which fuse concepts from both constraint-based Bayesian techniques and search-and-score Bayesian methods. The MMHO-DBN first uses constraint-based ideas to limit the space of potential structure and then applies search-and-score ideas to search for an optimal HO-DBN structure. We evaluated the ability of our MMHO-DBN approach to identify genetic regulatory networks (GRN’s) from gene expression time-series data. Preliminary results on artificial and real gene expression time-series are encouraging and show that it is able to learn (long) time-delayed relationships between genes, and faster than current HO-DBN learning methods.

Keywords—Gene Regulatory Networks, High-Order Relationships, Dynamic Bayesian Network, Max-Min Heuristic

I. INTRODUCTION

Accurate and fast reconstruction of genetic regulatory networks (GRNs) is an important task that has recently become possible due to large-scale high-throughput experiments such as microarray experiments [1]. Gene expression levels obtained over sufficiently large number of time-points can be used to identify GRNs. It is well known that the expressions of a given genes can affect how certain genes are expressed, either down-regulated or up-regulated. GRN represents such causal relationships interactions among genes, encoding all the temporal dependencies between genes in an organism [2]. Regulatory events within an organism are asynchronous, that is different genes can regulate other genes at different time-scales and with different delays.

Accurate and efficient reconstruction of GRNs from expression time-series data is a computationally hard task, in particular due to fact that expression levels are measured for a large number of genes numbering in the thousands, and over few number of time-points numbering in the tens. GRN identification methods based on ordinary differential equations (ODEs) techniques [2] will be prohibitively slow on such amount of data. GRN inference techniques such as Boolean network [2] methods are not causal and are not very robust to noise and uncertainty in the data. GRN reconstruction approaches based on probabilistic graphical model (PGM), 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; missing data can also be taken care. PGMs are also more efficient for processing large number of genes [3].

Bayesian networks (BNs) are PGMs which compactly represents a joint probability distribution among a set of variables [4]. They are directed acyclic graphs (DAGs) which can appropriately model GRNs, that is they model genes as nodes and causal dependencies between genes as edges [5]. Due to their acyclicity constraint, BNs are unable to model self-regulations, feedback loops, and time-delayed interactions, which are the characteristic of GRNs. Dynamic BNs (DBNs) are proposed to tackle these limitations by unrolling a BN over time [6]. In DBNs, a transition network between any two consecutive time-points characterizes the GRN; that is only genes at time-point \( t-1 \) are supposed to regulate genes at time-points \( t \). This is a first-order assumption allowing to model temporal causal dependencies among genes. First-order DBNs (FO-DBNs) however, cannot model time-delayed interactions longer than one time step. To this effect, high-order DBNs (HO-DBNs) were introduced by [7] to model longer time-delayed interactions.

In this paper, we contribute 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 the current BN approaches for learning the structure of BNs from static data.

The rest of this paper is organized as follow. Section II presents GRN modeling with HO-DBNs. In Section III, we discuss current methods to learning HO-DBN structures for reconstructing GRNs from microarray time-series data. Then we introduce our MMHO-DBN structure learning method in Section IV. Preliminary results and discussions are presented in Section V. Finally, we conclude and suggest possible direction of research in HO-DBN learning.

II. MODELING TIME-DELAYED REGULATIONS WITH HO-DBNs

Let us consider a gene expression time-series data set \( \mathbf{g}_{T \times N} = (\mathbf{g}_1, \ldots, \mathbf{g}_T) \) summarizing the observations (i.e.,
expression levels) of $N$ genes at $T$ time-points. Row vector $g_t = (g_{t,1}, \ldots, g_{t,N})^T$ contains the expression levels of the $N$ genes measured at time-point $t$ and where $g_{t,j}$ is an observation from the random variable $G_{t,j}$, for $1 \leq t \leq T$ and $1 \leq j \leq N$.

The DBN [8] usually refers to the first-order DBN (FO-DBN). FO-DBN assumes a Markov dependency of order 1 over time; that is, the expression level of a gene at time $t$ depends only on the expression levels of the genes at time $t-1$ and $t$. FO-DBN is defined by a pair of structures $(S_{t-1}, S_t)$ corresponding to networks at time slices $t-1$ and $t$, and a transition network $S_{[t-1:t]}$ of interactions between $S_{t-1}$ and $S_t$; thus, $S_{[t-1:t]}$ has $2N$ nodes. The FO-DBN structure is obtained by unrolling the transition network over time, and the parents of a variable $G_{t,j}$ are from time-slices $t-1$ and $t$ only. FO-DBN cannot model time-delayed interactions more than 1 time unit which occur in GRNs but can be extended to allow higher-order interactions among variables.

High-order DBNs (HO-DBNs) have been proposed to model time-delayed interactions between genes, where the structure and parameters of the HO-DBN are learned by assuming a fixed order $r > 1$ [7], [3], [9], [10] representing the maximum allowed delay among genes. The $r$-order DBN ($r$-DBN) assumes a $r$-order Markov dependency over time. It is defined by a $(r+1)$-tuple of structures $(S_{t-r}, S_{t-(r+1)}, \ldots, S_{t-1}, S_t)$ corresponding to networks at time-slices $t-r, t-(r+1), \ldots, t-1, t$, and a transition network $S_{[t-r:t-1]}$ (or $S_{[t-r,t]}$, for short) representing the causal connectivity structure between each network $S_{t-l}$ and $S_{t}$. The structure of the $r$-DBN is obtained by unrolling $S_{[t-r,t]}$ over time. The transition network $S_{[t-r:t]}$ consists of $(r+1)N$ nodes, and the parents of a variable $G_{t,j}$ are chosen from the set of variable $\bigcup_{i=0}^{t} G_{t-i}$, where $G_i = \{G_{i,1}, G_{i,2}, \ldots, G_{i,N}\}$ is the set of $N$ random variables at time-slice $i$. We assume an $r$-order stationary Markov chain and that the networks $S_{t-l}$, $0 \leq l \leq r$, have no edges. The GRN can be represented as a matrix $C = \{c_{i,j}\}_{N \times N}$, where $1 \leq c_{i,j} \leq r$ denotes the time delay of regulation between gene $i$ and its parent gene $j$.

Let $G = (G_{t,1}, \ldots, G_{t,N})^T$ where each $G_i = (G_{t,1}, \ldots, G_{t,N})^T$ is a $N$-dimensional random variable vector. Under the Bayesian framework, a gene is a random variable and we consider directed acyclic graph $S$ and a $r$-order stationary Markov assumption between nodes. The $r$-order DBN ($r$-DBN) assumes a $r$-order Markov dependency over time:

$$P(G_t|G_{t-1}, \ldots, G_1) = P(G_t|G_{t-r}, \ldots, G_{t-r})$$

The $r$-DBN thus decomposes the joint probability distribution of $G$ given the structure $S$ into a product of conditional probabilities by assuming independence of non-descendant variables as:

$$P(G) = \prod_{t=1}^{T} P(G_t|G_{t-1}, \ldots, G_{t-r})$$

Let $P_{t-r:t} = (P_{t-1:t}, \ldots, P_{t-r,t})^T$ be the $q_{t-r:t}$-dimensional random vector of the parents of the $j$-th gene at time $t-l$, $1 \leq l \leq r$; $P_{t-r,t} = \emptyset$ if $t-l \leq 0$. We define the set of all parents of the $j$-th gene as the $q_{j}$-dimensional vector $P_{[1,r],j} = \bigcup_{t=1}^{T} P_{t-r,t}$, where $q_j = \sum_{t=1}^{T} q_{t-r,t}$. Then the conditional probabilities $P(G_j|G_{t-1}, \ldots, G_{t-r})$ can be decomposed into a product of conditional probabilities of each gene given its parents $P_{[1,r],j}$ as:

$$P(G_j|G_{t-1}, \ldots, G_{t-r}) = \prod_{j=1}^{N} P(G_{t,j}|G_{t-1,j} \cup \cdots \cup G_{t-r,j})$$

The important issue pertaining to modeling GRNs by $r$-DBNs is how to find the conditional probabilities $P(G_j|P_{[1,r],j})$ which best explain the data. To determine the optimal $P(G_j|P_{[1,r],j})$, we parameterize $P(G)$ by a parameter vector $\theta = (\theta_1, \ldots, \theta_N)$ and transfer the determination of the optimal $P(G)$ into the estimation of the best $\theta$. Parameterizing and substituting Eq. (3) into Eq. (2) we obtain the discrete $r$-DBN model:

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

By using $r$-DBN models, we can model higher-order GRN interactions from time-series data, when we know the true relationships among the genes. Such relationships are still unknown, and hence, it is necessary to devise criteria for evaluating the goodness of a structure, and then, devise search algorithms for searching the large space of candidate structures. In this space, the optimal structure $\hat{S}$ is the one which maximizes the posterior probability $P(S|G)$. From Bayes theorem we have:

$$P(S|G) = \frac{P(S)P(G|S)}{P(G)} \approx P(S)P(G|S)$$

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

$$P(G|S) = \int_{\theta} P(G|S, \theta)P(\theta|S)d\theta$$
where $P(\theta|S)$ is the prior probability of the parameter $\theta$ and $P(G|S, \theta) = P(G|\theta_S)$; note that we write $\theta_S = (\theta_1, \ldots, \theta_N)$ since the form of $\theta$ in Eq. (4) is equivalent to the network structure $S$. The maximum a-posteriori (MAP) estimate of the optimal structure $S$ is then given as:

$$\hat{S}_{\text{MAP}} = \arg \max_S P(S) \int \theta S P(G|\theta_S)P(\theta_S|S)d\theta_S \quad (7)$$

The problem which remains now is to 1) determine the conditional probabilities $P(G_t|p_{[1,r],j}; \theta_j)$, 2) determine the prior probabilities $P(S)$ and $P(\theta_S|S)$, 3) compute the high-dimensional integral, and 4) search for the optimal $\hat{S}_{\text{MAP}}$. Points 1) through 3) combine into a single criterion for learning GRNs based on r-DBN, and which is used within a search algorithm in point 4) to evaluate the goodness of candidate GRN structures.

For the discrete model we assume that gene expression values are 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 $p_{[1,r],j}(t)$ (the parents set of the $j$-th gene) can take is $Q_j = d^q$. Each state of $p_{[1,r],j}$ is also associated with a lag vector $L_{[1,r],j} \in \{1, \ldots, r\}^q$ containing the delay of each parent of gene $j$; hence, the number of distinct lag vectors for gene $j$ is $L_j = r^q$. Let $j,q,l,k \in \{1, \ldots, r\}^q \times \{1, \ldots, d\}$ be the number of observations satisfying $G_{t,j} = k$, $p_{[1,r],j} = q$, and $L_{[1,r],j} = l$, and $N_{j,q,l,k} = \sum_{t=1}^{T} \delta(G_{t,j} = k, p_{[1,r],j} = q, L_{[1,r],j} = l)$ be the number of observations in the transition network $S_{[t-r,t]}$ for $1 \leq t \leq T$. Using the property of decomposability [6], we can model $P(G|\theta_S)$ as a product of local structures at each gene $j$:

$$P(G|\theta_S) = \prod_{j=1}^{N} \prod_{q=1}^{Q_j} \prod_{l=1}^{L_j} \prod_{k=1}^{d} \theta_{N_{j,q,l,k}}^{N_{j,q,l,k}} \quad (8)$$

Define $\theta_S = \bigcup_{j=1}^{N} \{\theta_j\}$, $\theta_j = \bigcup_{q=1}^{Q_j} \{\theta_{j,q,l}\}$, and $\theta_{j,q,l} = \bigcup_{l=1}^{L_j} \{\theta_{j,q,l,k}\}$. Assuming that the global and local parameter vectors are independent of each other, then we can decompose the prior distribution on $\theta_S$ as:

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

Substituting Eqs. (8) and (9) into Eq. (7), we obtain

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

Assuming Dirichlet distribution [11] with hyper-parameters $\alpha_S = \bigcup_{j=1}^{N} \{\alpha_j\}$ and $\alpha_j = \bigcup_{q=1}^{Q_j} \{\alpha_{j,q,l}\}$ as the prior distribution on the global and local parameters, then:

$$P(\theta_j|S) = \text{Dir}(\theta_j|\alpha_j) \quad (11)$$

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

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

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

1) Bayesian Dirichlet-equivalent (BDe) metric of [12] when $\alpha_{j,q,l} = \alpha P(G_{t,j} = k, p_{[1,r],j} = q, L_{[1,r],j} = l|S)$, and $\alpha \geq 0$ is the equivalent sample size parameter.

2) BDe uniform (BDeu) metric of [13] when $\alpha_{j,q,l,k} = \frac{N_{j,q,l,k}}{N_{j,q,l}}$.

3) K2 metric of [14] when $\alpha_{j,q,l,k} = 1$.

To complete the information required to derive the MAP estimate, $\hat{S}_{\text{MAP}}$, we must consider the prior probability $P(S)$ of a given structure. Let $S_j$ be the local structure at the $j$-th gene, we can set the prior probability of $S$, that is:

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

Taking the product of Eqs (12) and (13) yields the r-DBN learning criterion. We 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. That is, we have

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

If we assume that every structure in the structure space is equally probable a priori, then we can simplify Eq (7) to
use the maximum likelihood (ML) estimate of the optimal structure $S$ given as

$$
\hat{S}_{ML} = \arg \max_S \int_{\theta_S} P(G|\theta_S)P(\theta_S|S) d\theta_S
$$

(15)

In this case we estimate the parameters $\theta_{j,q,l,k}$ as

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

(16)

and then use the Bayesian Information Criterion (BIC) of [15] to approximates the integral as:

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

(17)

The BIC score does not require priors over parameters and is computationally faster to compute but less accurate than the BDe, BDeu, and K2 scores. The BIC score is good when given a large data set. Finding the optimal structure of a BNs is known to be NP-hard 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.

III. RELATED WORKS

Few methods have been proposed in literature to address the difficulty of modeling time-delayed regulations. The authors of [16] devised a three-steps approach which essentially transforms the HO-DBN problem into FO-DBN problem thus avoiding to learn from an extremely large space of parameters. Their approach, DBN-ZC, is a local search-and-score learning method which proceeds as follows. First, DBN-ZC limits the potential regulators of each target gene $T$ to only those genes with either earlier or simultaneous expression changes in relation to target $T$; thus significantly reducing the computational effort in the subsequent structure learning phase. Second, the time difference between the initial gene expression change of a potential regulator and $T$, respectively, is considered. Third, the time-series profiles of the potential regulators are appropriately aligned to that of $T$ according to their time-lags with $T$, and then a search-and-score based FO-DBN learning is performed to select the regulators with the highest log-marginal likelihood as the final set of regulators of $T$. In [7], a two-steps heuristic framework is devised to learn $r$-DBNs from time-series expression data. First, pairs of variables $G_{t-l,j}, G_{t,k}$ with time lag $l$, for $1 \leq l \leq r$, $1 \leq t \leq T$ and $1 \leq j, k \leq N$, having mutual information above a given threshold are determined; this step essentially initializes the transition network $S_{[t-r,t]}$. Then in the second step, genetic algorithm (GA) is applied given the initial transition network $S_{[t-r,t]}$ to find the structure having the highest Maximum Likelihood or the Maximum Description Length score. Being a population-based optimization method using implicit parallelism, GA is able to search very large spaces given a good representation of the GRN and appropriate genetic operators. Chaturvedi and Rajapakse [10] used prior biological knowledge contained in current protein interaction networks (PINs) as a mean of limiting the search space. The authors modeled time-delayed regulations using a skip-chain model which predict edges between non-consecutive time-points (called skip-edges) based on the prior knowledge in the given PINs. Viterbi approximation of DBNs is used to select the best skip-edges and combined appropriately with the BIC score which selects edges between two consecutive time-points (non-skip edges called linear-edges in the paper). In [9], the variable-order DBN (VO-DBN) approach is introduced to automatically find the delays of regulations between genes. In HO-DBNs, the order $r$ (or the maximum delay of regulation) is fixed a priori before learning the structure. In the VO-DBN, however, the optimal order $r$ and the optimal structure of which are learned using a Markov chain Monte Carlo (MCMC), which uses an appropriate acceptance mechanism allowing to optimize both order and structure.

IV. MAX-MIN HIGH-ORDER DBNS

In this section, we present an extension of the Max-Min Hill-Climbing (MMHC) heuristic which was originally devised for learning the structure of BNs from static data. MMHC-BN was introduced in [4] as a fast, scalable, and reliable BN learning method which overcomes the perceived limitations of the current state-of-the-art BN algorithms and which also exist in current HO-DBN algorithms. MMHC-BN is a hybrid BN method; it first uses constraint-based Bayesian learning [17] to learn the skeleton (i.e. an undirected graph) of a BN, and then performs a search-and-score Bayesian learning on the skeleton in order to orient its edges. It is the skeleton learning phase which gives MMHC-BN its reliability and accuracy, its efficiency, and more importantly, its ability to scale to distributions with thousands of variables. MMHC-BN is also a local learning method which 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 (PC), of each variable during the skeleton learning phase. This discovery was proven accurate and more efficient than that of the hybrid Sparse Candidate (SC) algorithm [5] and that of the constraint-based methods such as PC algorithm [17].

Our Max-Min High-Order DBN (MMHO-DBN) structure learning method is shown in Algorithm 1. The MMHO-DBN approach proceeds with the two MMHC-BN phases, except both phases are appropriately modified to consider a discrete-time stochastic process $G = (G_{1}, \ldots, G_{T})^{T}$ having a joint probability distribution $P$ and where each $G_{t} = (G_{t,1}, \ldots, G_{t,N})^{T}$ is a $N$-dimensional random variable taking discrete values.
Algorithm 1 The MMHO-DBN Algorithm

Input: $g_{T \times N} = (g_1, \ldots, g_T)^T$; time-series data
 $r$: maximum time delay
 $\alpha$: significance level

Output: $\hat{\Pi}$: best DAG on the variables in $g_{T \times N}$

{Phase 1: Restrict candidate parents}
for every variable $G_{t,j} \in G$
do
$\Pi_{[t-\tau,t],j} \leftarrow$ DMMP($G_{t,j}, g_{T \times N}, r, \alpha$)
end for

{Phase 2: Search for the best DAG $\hat{\Pi}$}
for every variable $G_{t,j} \in G$
do
$\mathcal{P}_{[t-\tau,t],j} \leftarrow$ Best_Subset($G_{t,j}, g_{T \times N}, r, \Pi_{[t-\tau,t],j}$)
end for
return the highest scoring DAG $\hat{\Pi}$ found

In Phase 1 of Algorithm 1 we modified the local discovery method of [4], the Max-Min Parent and Children (MMPC) algorithm in order to compute the maximum possible parent set, $\Pi_{[t-\tau,t],j}$, of each target variable $G_{t,j}$. Given a target variable $T$ and statistical non-stochastic data $D$, the original MMPC algorithm returns its maximum possible set of parent and children, $PC(T)$, provided that the faithfulness assumption [17] holds and that the statistical tests performed return reliable result. 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. MMPC is a constraint-based search algorithm which essentially learn the skeleton of a BN; that is, it identifies the existence of edges to and from targets $T$ without identifying the orientation of the edges. The uniqueness of the $PC(T)$ is also true for $r$-DBNs faithful to the distribution $\mathcal{P}$. Here, however, the edges will be oriented due to the temporal dependencies, and thus we need only find the maximum possible set of parents of $T$; the children of $T$ will be determined following the temporal dependencies. In Algorithm 1 above, the Dynamic Max-Min Parent (DMMP) algorithm in Algorithm 2 is our temporal variant of the MMPC algorithm for computing the maximum possible set of parents, $\Pi_{[t-\tau,t],j}$ of a target variable $G_{t,j}$.

Algorithm 2 is the same as the MMPC algorithm of [4] except here the PC set is the set of maximum possible parents, $\Pi_{[t-\tau,t],j} \subseteq G_{[t-\tau,t]}$, of the target variable $G_{t,j}$; $G_{[t-\tau,t]}$ is the set of all variables within the last $r$ previous time-points $t - \ell$ for $1 \leq \ell \leq t$. Starting from an empty $\Pi_{[t-\tau,t],j}$, Phase 1 of DMMP algorithm (2) sequentially adds variables $\Pi_{\lambda,\mu} \in G_{[t-\tau,t]}$ which maximize the minimum association with the given target $G_{t,j}$ relative to the current $\Pi_{[t-\tau,t],j}$; [4] proved that the $\Pi_{[t-\tau,t],j}$ found in Phase 1 DMMP does not contain false negatives but may contain false positives which are then removed subsequently. As in [4], we define the minimum association between a variable $G_{t,i} \in G_{[t-\tau,t]}$ and the target $G_{t,j}$ relative to a subset $Z \subseteq G_{[t-\tau,t]}$ as:

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

Algorithm 2 performs tests of independence $Ind(G_{t,i}; G_{t,j}|Z)$ which returns true if $G_{t,i}$ and $G_{t,j}$ are conditionally independent given $Z$. The function $Assoc(G_{t,i}; G_{t,j}|Z)$ estimates the strength of dependency between $G_{t,i}$ and $G_{t,j}$ given $Z$ such that $Assoc(G_{t,i}; G_{t,j}|Z) \geq 0$ with equality holding if and only if $Ind(G_{t,i}; G_{t,j}|Z)$. For the independence tests $Ind(G_{t,i}; G_{t,j}|Z)$ we calculated the $G^2$ statistic as in [4] under the null hypothesis of the conditional independence holding and using the same number of degrees of freedom as [4]. The $G^2$ returns a p-value and we reject the null hypothesis when $p-value < \alpha$. We set the significance level $\alpha = 0.05$ and define the measure of association as:

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

Since the gene expression time-series data are sparse and the number of counters goes exponentially as the number of parents increases, there probably are some zero cells in
the contingency table when conducting conditional independence test. This may also lead the degree of freedom to be negative, which is a computational disaster when applying MMHC and its high-order extension to sparse gene time-series data. For example, we test \( \text{Ind}(A, B|C) \) where nodes \( A, B \) and \( C \) have two states, respectively. The contingency table have 8 cells, and the degree of freedom is \( 1 \times 1 \times 2 = 2 \). If there are more than 2 cells are zeros, then the degree of freedom becomes negative. We propose a smooth method that is inspired by the computation of BDeu score [12]. We add a constant number, which is called equivalent sample size (ESS), to each cell of the contingency table. In the current study, we uniformly set the ESS to 10, though ESS following other distributions is also possible.

After determining the set \( \Pi_{t-r,t},j \) target variable \( G_{t,j} \) \( (1 \leq j \leq N) \), Phase 2 of MMHO-DBN will then perform a search-and-score strategy in order to find the best subset of parents \( P_{t-r,t},j \subseteq \Pi_{t-r,t},j \) maximizing a score function (e.g., BBe, BDeu, K2, BIC, etc). The search starts with an empty DAG and is constrained to only consider adding an edge \( "\Pi_{t-r,t},j \rightarrow G_{t,j}" \) if \( \Pi_{t-r,t},j \in P_{t-r,t},j \); that is, the search for best subset \( P_{t-r,t},j \) is constrained to the set of possible parents \( \Pi_{t-r,t},j \) only. Our search algorithm is shown in Algorithm 3. In Algorithm 3, we used the Bayesian Dirichlet-equivalent uniform (BDeu) metric as the score function \( \text{Score} \). The parameter \( \gamma \) is the maximum allowed cardinality of \( \Pi_{t-r,t},j \) below which we can perform an exhaustive search, otherwise we perform a heuristic search for best subset. In the algorithm, we are only searching for subsets \( P_{t-r,t},j \) with \( |P_{t-r,t},j| \leq \varphi \) which maximize BDeu score; essentially, parameter \( \varphi \) limits the size of the search space for the sake of computational efficiency. In our current implementation, we performed only an exhaustive search and set the parameter \( \varphi \) to three. We plan to devise heuristic search methods in the near future.

**Algorithm 3 The Best_Subset Algorithm**  
**Input:** \( G_{t,j} \): target node  
\( g_{T \times N} = (g_1, \ldots, g_T)^T \): time-series data  
\( r \): maximum time delay  
\( \Pi_{t-r,t},j \): maximum possible parent set of \( G_{t,j} \)  
\( \varphi \): cardinality limit for \( P_{t-r,t},j \)  
\( \gamma \): cardinality limit for exhaustive search  
**Output:** \( P_{t-r,t},j \): best subset of parents of \( G_{t,j} \)

if \( |\Pi_{t-r,t},j| \leq \gamma \) then  
{Exhaustive search}  
\( P_{t-r,t},j \leftarrow \text{arg max}_{P_{t-r,t},j} \text{Score}(P_{t-r,t},j, G_{t,j}) \)  
else  
{Heuristic search}  
\( P_{t-r,t},j \leftarrow \text{Heuristic}(G_{t,j}, \Pi_{t-r,t},j, \text{Score}, \varphi) \)  
end if  
return \( P_{t-r,t},j \)

In this section, we investigate the performance of our MMHO-DBN approach in terms of accuracy and running time. We compared our MMHO-DBN, with DBmcmc (a first-order DBN) [18] and DBN-ZC [16] methods. We set the maximum-fan-in of each gene to 3 in all methods for fair comparison. We did two parts of experiments.

Our first experiment is to test whether ours can identify regulators of different time-delays. We designed a small network of 8 nodes as shown in Figure 1(a). The time-delays are given along the directed connections. The values of the network are discrete, and include 1 (down) and 2 (up). This network is composed of a pair of coherent nodes, regulators of different time-delays, and regulators of the same time-delay. Using this network, we generated data of 80 time points with equal sampling rate. In our MMHO-DBN, we set the maximum time-delay to 3. The significant level was \( \alpha = 0.05 \). The predicted networks by our MMHO-DBN, DBmcmc, and DBN-ZC are demonstrated in Figures 1(b), 1(c), and 1(d), respectively. Their performances are compared in Table I. The presence of an directed connection is defined as positive, and an absent edge is negative. From Figure 1, we can see that our MMHO-DBN can identify all the existing connections with correct time-delays, whereas DBmcmc can only predict the connections of 1 time-delay. DBN-ZC method fails to identify all existing connections, which convinces us that grouping regulators according to time-delays may not be a wise choice. DBN-ZC only searches among the subsets of the potential regulators with the same time-delays. Moreover, the time-lag estimation through measuring the initial changes does not make more sense on noisy and truncated data. As a high-order DBN, our approach runs very fast. In this experiment, it took only 11 seconds, while DBmcmc took 584 seconds.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Time (second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMHO-DBN</td>
<td>1</td>
<td>0.9138</td>
<td>11</td>
</tr>
<tr>
<td>DBmcmc</td>
<td>0.5</td>
<td>1</td>
<td>584</td>
</tr>
<tr>
<td>DBN-ZC</td>
<td>0</td>
<td>0.8621</td>
<td>0.4</td>
</tr>
</tbody>
</table>

In the second experiment, we ran our MMHO-DBN and the benchmark methods on a yeast metabolic-cycle dataset [19]. We picked up 44 genes that correspond to three periodical biological processes: Ox (oxidative), R/B (reductive, building), and R/C (reductive, charging). The data was sampled at 36 equally distributed time points. We set the maximum time-delays in MMHO-DBN to 1, 3, 5, and 7, respectively. The predicted GRNs are shown in Figure 2. We define the gene that regulates four or more genes as hub gene. For time-delays 1 and 3, genes POX1, FOX2, and MRPL10 are identified as hub genes by our MMHO-DBN.
In time-delay 5, SSB2 and RML2 are find as hub genes in addition to the three hub genes above using smaller time-delays. Using time-delay 7, we can predict CIT3, MRPL10, and RML2 to be hub genes. The GRN reconstructed by DBmcmc is shown in Figure 3. We can observe that CIT3, CAT2, ICL2, and RPSOB are predicted as hub genes using first-order method. The GRN identified by DBN-ZC is so sparse that only hubs POX1, MRPL10, FOX2, and few single regulators are found as can be seen in Figure 4. Comparing the results obtained by these methods, we can see that POX1, FOX2, MRPL10, and CIT3 are commonly recognized as hub genes. Since the actual GRN is never known, there is no gold standard to validate the quality of predicted GRN. Our current result is preliminary. We will find a reasonable validation to further compare their performance. However, the advantage of our method can be felt in our first experiment on simulated data.

The running time of these methods are listed in Table II. DBN-ZC is the fastest method, however as can be seen above the result of this method does not look better. Using time-lag 1, our MMHO-DBN is much faster than DBmcmc. As the maximum time-lag increase, the running time of our MMHO-DBN does not increase dramatically.

<table>
<thead>
<tr>
<th>Method</th>
<th>Max Time-Lag</th>
<th>Time (second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMHO-DBN</td>
<td>1</td>
<td>147</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>323</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>478</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>432</td>
</tr>
<tr>
<td>DBmcmc</td>
<td>1</td>
<td>379</td>
</tr>
<tr>
<td>DBN-ZC</td>
<td>-</td>
<td>71</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

In this paper, we propose a fast high-order dynamic Bayesian network learning method for reconstructing gene regulatory networks. This is a constraint-and-scoring
method. In the algorithm, we also propose to use equivalent sample size to overcome the potential computational problem when testing the conditional independence. The preliminary experiment on simulated data shows that our method can identify regulators of different time-delays. The experimental result on real data proves that our approach is very efficient. We are currently working on a validation approach for comparing different network learning approaches on real gene expression time-series data.

ACKNOWLEDGMENT

This research has been supported by OGS 2011-2013, and NSERC Grants #RGPIN228117-2011.

REFERENCES


