Non-stationary Bayesian Networks based on Perfect Simulation

Yi Jia, Wenrong Zeng, Jun Huan*
Department of Electrical Engineering & Computer Science, University of Kansas
Lawrence, KS, 66045, USA
jiayi@ittc.ku.edu, wrzeng@ittc.ku.edu, jhuan@ittc.ku.edu

ABSTRACT

Non-stationary Dynamic Bayesian Networks (Non-stationary DBNs) are widely used to model the temporal changes of directed dependency structures from multivariate time series data. However, the existing change-points based non-stationary DBNs methods have several drawbacks including excessive computational cost, and low convergence speed. In this paper we proposed a novel non-stationary DBNs method. Our method is based on the perfect simulation model. We applied this approach for network structure inference from synthetic data and biological microarray gene expression data and compared it with other two state-of-the-art non-stationary DBNs methods. The experimental results demonstrated that our method outperformed two other state-of-the-art methods in both computational cost and structure prediction accuracy. The further sensitivity analysis showed that once converged our model is robust to large parameter ranges, which reduces the uncertainty of the model behavior.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

Algorithms

Keywords

Dynamic Bayesian Networks, Markov Chain Monte Carlo, Perfect Simulation

1. INTRODUCTION

Non-stationary Dynamic Bayesian Network methods are widely used to model the temporal changes of dependency structures from multivariate time series data [32, 10, 34, 17, 22, 5, 16, 11]. Comparing to traditional DBNs modeling, non-stationary DBNs have advantages to capture the structural dynamics of networks in various biological systems, such as Neural assemblies in response to visual stimuli [34], morphogenesis in the organisms' life cycle [32, 16], adaptive mammalian immune response against infection of virus [10], or circadian regulation dynamics of plants caused by dramatic changes of outside environment such as light intensity [10].

Several methods have been developed for constructing non-stationary models. For example, Robinson et al. proposed a discrete non-stationary DBNs method [32] using Reversible Jump Markov Chain Monte Carlo (RJMCMC) [8] to sample underlying changing network structures. Grzegorczy et al. proposed a non-homogeneous continuous Bayesian network method with a Gaussian mixture model based on the allocation sampler technique [27]. Grzegorczy et al. improved the convergence of their method using perfect simulation modeling [6] and reduced the risk of overfitting and inflated inference uncertainty [16] in their later work [11]. Both Robinson's and Grzegorczyk's methods perform change-point detection and we call them change-point based approaches. Song et al. [34] proposed a time-varying DBNs (TV-DBNs) method and used a kernel re-weighted l_1-regularised auto-regressive approach for learning the graph structures at each time step. Lebre et al. [22] proposed a more flexible auto-regressive time varying model called ARTIVA that allows gene-by-gene analysis. Husmeier et al. [16] introduced inter-time segment information sharing schemes to address the over-flexibility issue in the ARTIVA approach. Those three approaches are different from the change-point detection based approaches and fell into the category of structure learning of constantly varying network over time. In this paper, our work focuses on the change-point detection modeling for regulatory network dynamics.

There are several limitations of the existing change-points based techniques. First, the mixture model used by Grzegorczy et al. [10, 11] assumed that the underlying network structures are invariant over time. Such an assumption is too rigid when changes of network structures are expected, for example, morphogenesis or embryogenesis [16]. Second, Grzegorczy's method with the improvement on convergence [11] mixed the structure sampling steps and the perfect simulation steps in the same RJMCMC procedure. The time complexity of each perfect simultion step is quadratic to the number of the observations [6]. This scheme brings extra computational costs on change-point simulation that are proportional to the number of sampling iterations even if genes are decomposed into groups to alleviate the computational concern. Third, the RJMCMC sampling approach in Robinson's work [32] converges slowly. For example, the results in [8] using RJMCMC did not converge as pointed out in the subsequent work in [9]. In addition, our experiments show that the structure prediction accuracy of Robinson's RJMCMC is low.

We posit that the key computational obstacle for efficient mod-

*to whom correspondence should be addressed

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

eling of time series data with non-stationary DBNs methods is the interplay of change-point detection and structure inference for each identified time segment. To improve computational efficiency, we designed an algorithm called ReCursion Non-Stationary Dynamic Bayesian Networks (RCnsDBNs) to separate these two essential steps. Our method adopted Fearnhead's perfect simulation model [6] for change-point detection. The perfect simulation model was originally developed for univariate time series data [6] and we modified the algorithm to model our multi-variate time series data. In particular, we designed an iterative algorithm for the structure inference and change-point detection. Our method first used the point process [29] as the prior for the occurrences of change-points and directly simulated the change-points from the posterior distribution. For each predicted segment, we then used a regular Markov Chain Monte Carlo (MCMC) method, a revised KNUT (Known Transition Number Unknown Transition Time) setting in Robinson's method [32]. Once the algorithm converges, we output the most likely change-points and a sequence of network structures corresponding to the separated segments.

There are several advantages for the novel non-stationary DBNs algorithm. First, by directly simulating the posterior distribution of transition time for graph structures, our method efficiently reduces the model space and improves the computational performance both on time and numbers of sampling iterations for covergence. Second, even if a negative binomial prior is adopted in our point process, the experiments showed that our experimental results are stable within large parameter ranges, which reduces the uncertainty of the model behavior. Third, different from Grzegorczy's method [11], our method only needs to simulate the change-points once for each round of our algorithm. It saves the computational time. Fourth, Our approach outperform Robinson's RJMCMC approach on structural prediction accuracy. Even if our discrete model needs the discretization of the data, compared with Grzegorczyk's continuous approach, our method showed the competitive performance for structure estimation.

2. RELATED WORK

The change-point detection problems have been extensively investigated in time series models. Recent work could be found in: PCA-based singular-spectrum transformation models [26]; non-parametric online-style algorithm via direct density-ratio estimation [20]; two-phase linear regression model [23]; a hybrid algorithm that relies on particle filtering and Markov chain Monte Carlo [3]; the RJMCMC method [8]; The perfect simulation model based on product-partition model [6]; change-point detection by minimizing a penalized contrast function [21]. These models are widely used in various applications, such as climate analysis [23], coal-mining disaster analysis [8, 6], well-log analysis [6], the analysis of abrupt economic agents' behaviors [3], and asset price volatility [21].

Researchers find that change-point modeling is a very promising way of dealing with the non-stationarity property [3]. Hence, the current non-stationary DBNs methods employed different change-point detection techniques to model the underlying change-point processes of network structures. Robinson et al. [32] applied RJMCMC [8] and used a discrete model with the assumed multinomial distributed data with the Dirichlet prior. Using the RJMCMC technique, Lebre et al. [22] proposed a new time varying networks approach based on first-order auto-regression and Yao's two-stage regime-SSM model [30]. Their method focuses on local structural changes and performs node-by-node analysis. Further, in order to address the structural overfitting problem in [22], Dondelinger et al. [5] and Husmeier et al. [16] introduces information sharing between segments into Lebre's approach. Grzegorczy et al. [10, 11] applied the allocation sampler technique and introduced a continuous-valued DBNs method that approximates the non-stationary property with a Gaussian mixture model. Based on their work, Ickstadt et al. [17] further generalized this non-linear BGe mixture model into a broader framework of non-parametric Gaussian Bayesian networks. In this paper, we incorporate the perfect simulation modeling into our dynamic bayesian framework and provide a computationally efficient non-stationary DBNs approach. We chose this change-point detection technique for the following reasons. First, perfect simulation is based on bayesian analysis and can be easily applied into our MCMC algorithm. Second, with an approximation in the recursion, the computational complexity of this method is approximately linear to the number of observations.

3. METHODS

3.1 Perfect Simulation Modeling

Fearnhead used the perfect simulation model to find change-points in the univariate time series data [6]. We adapted his method to the framework of our dynamic bayesian networks, and provided a non-stationary DBNs method to detect the change-points for network structures in multivariate time series data.

We consider an observed time series data $D = \{y_1, \cdots, y_T\}$ spanning T time points, where each observation $y_i \in \mathbb{R}^n : 1 \leq i \leq T$ is a n dimensional vector $(x_1, \cdots, x_n)$. The time series data is subdivided to m segments $D = \{D_1, \cdots, D_m\}$, where m is unknown. We denote the change-points for these segments as $L^T = (l_0, l_1, \ldots, l_{m-1}, l_m)$, where $l_0 = 0$ and $l_m = T$.

We assume the change-points as a point process on positive integers, which is characterized by a probability mass function $g(t)$, where t is the distance of two successive change-points. We choose the negative binomial distribution as the distribution for the distance between two successive change-points and have $g(t) = \binom{t-1}{k-1} p^k(1-p)^{t-k}$ with the parameters $k > 0, p > 0$ and its corresponding accumulative distribution function $G(t) = \sum_{i=1}^{t} g(i)$. For the special case of the first change-point, we have $g_0(t) = \sum_{i=1}^{k} \binom{t-1}{i-1} p^i(1-p)^{t-i}$ and $G_0(t) = \sum_{i=1}^{t} g_0(i)$.

Given the assumption of the independence between segments, we calculate the probability of a sequence of observations after one change-point l: $Q(t) = Pr(y_{t:n}|l = t-1)$ by using recursive function below:

$$Q(t)=\sum_{s=1}^{T-1} P(t,s)Q(s+1)g(s+1-t)+P(t,T)(1-G(T-t)). \quad (1)$$

where $2 \leq t \leq T$, and

$$Q(1)=\sum_{s=1}^{T-1} P(1,s)Q(s+1)g_0(s)+P(1,T)(1-G_0(T-1)). \quad (2)$$

In Equation 1 and 2, $P(t,s)$ is the simplified notation of $P(y_{t:s}|l = t-1) : 1 \leq t \leq T, t \leq s \leq T$, where the observations $y_{t:s}$ are in the same segment between two change-points $t-1$ and s. Similarly, $P(1,s)$ is the simplified notation of $P(y_{1:s}|l = 0) : 1 \leq s \leq T$, where the observations $y_{1:s}$ are in the same segment between two change-points l_0 and s.

Further, based on $Q(t)$, we calculate the probability distribution of the first change-point below:

$$P(l_1)=P(1,l_1)Q(l_1+1)g_0(l_1)/Q(1). \quad (3)$$

where $l_1 : 1 \leq l_1 \leq T-1$ is the first change point.

Given l_i, we calculate the conditional probability $P(l_{i+1}|l_i) : l_i + 1 \leq l_{i+1} \leq T-1$ as the following:

$$P(l_{i+1}|l_i)=P(l_i+1,l_{i+1})Q(l_i+1)g(l_{i+1}-l_i)/Q(l_i+1). \quad (4)$$

And the probability of no more change-point is given as:

$$P(T|l_i)=P(l_i+1,T)(1-G_0(T-l_i-1))/Q(l_i+1). \quad (5)$$

Finally, with the probability distribution of l_1 and conditional distribution of l_{i+1} given l_i, we directly simulate the change-point samples and compute the posterior probability distributions $P(L^T|T)$ and $P(m|T)$. Due to the limitation of the space, we omitted the mathematical derivation and proofs. More technical details are available on [6]. One of the key computations in the simulation procedure is to compute the probability $P(y_{t1:t2}|l = t1-1) : 1 \leq t1 \leq T, t1 \leq t2 \leq T$. By assuming the i.i.d. observations in a single segment and the conjugate priors $\rho(\theta)$ on the parameters θ associated with each segment, Fearnhead's work provides a closed form of solution for $P(y_{t1:t2}|l = t1-1) = \int \prod_{i=t1}^{t2} f(y_i|\theta)\rho(\theta)d\theta$. In the analysis of the well-log data, he assumed the normally distributed observations: $y_i \sim N(\mu_i, \sigma^2)$ with the fixed variance σ^2 and a normal prior for the mean μ_i. In the following we discuss our solutions both under the static and dynamic bayesian network frameworks.

Perfect simulation modeling in bayesian networks. Bayesian networks (BNs) are a special case of probabilistic graphic models. A static BN is defined by an acyclic directed graph G and a complete joint probability distribution of its nodes $P(X) = P(X_1, \ldots X_n)$. The graph $G : G = \{X, E\}$ contains a set of variables $X = \{X_1, \ldots, X_n\}$, and a set of directed edges E, defining the causal relations between variables. With a directed acyclic graph, the joint distribution of random variables $X = \{X_1, \ldots, X_n\}$ are decomposed as $P(X_1, \ldots, X_n) = \prod_i P(X_i|\pi_i)$, where π_i are the parents of the node (variable) X_i.

We assume that the observations inside one segment are independent. In each segment there is one graph $G_h : 1 \leq h \leq m$ that dominates the segment. We denote $y_{t1:t2}$ as $D^*_{t1:t2}$ and calculate $P(D^*_{t1:t2}|l = t1-1)$ as:

$$\begin{aligned} P(D^*_{t1:t2}|l{=}t1{-}1){=}\textstyle\sum_G P(y_{t1:t2}|l{=}t1{-}1,G)P(G) \quad (6)\\ {=}\textstyle\sum_G P(G)\prod_{j=t1}^{t2} P(y_j|G){=}\sum_G P(G)\prod_{j=t1}^{t2}\prod_i P(x_i^j|\pi_{x_i})\\ {=}\textstyle\sum_G P(G)\int P(D_{t1:t2}|G,\Theta_G)\rho(\Theta_G|G)d\Theta_G \end{aligned}$$

Θ_G are the parameters associated with the data $D_{t1:t2}$ corresponding to G. $\rho(\Theta_G|G)$ is the probability density function of Θ_G.

Under the assumption that the data are complete and multinomially distributed with a Dirichlet prior on the parameters Θ_G, we have the BDeu [13] solution to $P(y_{t1:t2}|l = t1-1)$:

$$\begin{aligned} P(D^*_{t1:t2}|l{=}t1{-}1,G){=}\textstyle\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})} \quad (7)\\ \textstyle\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})} \end{aligned}$$

r_i is the number of possible discrete values of x_i. q_i is the number of configurations of parents π_i for the variable x_i. N_{ijk} is the times that x_i had value k. $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. α_{ijk} and α_{ij} are the hyperparameters for Dirichlet distribution. α_{ijk} is assumed to be uniformly distributed inside a segment and is set to $\alpha_{ijk} = \alpha/(r_i q_i)$. The equivalent sample size α is set to 1.

In order to calculate $P(D^*_{t1:t2}|l = t1-1)$, we need to provide a model space $\mathcal{M}$ for G. We use MCMC to simulate $\mathcal{M}$. Given the collected sample size $N_{\mathcal{M}}$, we approximate the calculation of Equation 6 as follows:

$$P(D^*_{t1:t2}|l{=}t1{-}1){=}\textstyle\sum_{i=1}^{N_{\mathcal{M}}}\frac{1}{N_{\mathcal{M}}}P(D^*_{t1:t2}|l{=}t1{-}1,G) \quad (8)$$

Perfect simulation modeling in dynamic bayesian networks. The topology of bayesian networks must be a directed acyclic graph and hence could not be used to model the case where two nodes may be dependent on each other. As an extension of BNs to model time series data, Dynamic Bayesian Networks (DBNs) lift the limitation of directed acyclic graph by incorporating temporal dependence in constructing bayesian networks. It is not straightforward to extend the solution in BNs to modeling DBNs mainly due to that neighboring observations are not independent given the model parameters. Below we develop a heuristic and provide a solution for $P(D^*_{t1:t2}|l = t1-1)$ as follows.

We set the lag value $\tau = 1$ and assume the segments are overlapped. For each segment $D_h : 1 < h \leq m$, the length of overlapped area with the previous segments $D_1, \cdots, D_{h-1}$ is equal to the lag value $\tau = 1$. For D_1, there are no previous segments and we add $\tau = 1$ additional y_1s at the beginning of D_1. Given the transition time points L^T, each segment of observations $D_h = \{y_{l_{h-1}-\tau+1}, \cdots, y_{l_h}\}$, where $y_i = y_1$ when $i \leq 0$. We denote $\{y_{l_{h-1}+1}, \cdots, y_{l_h}\}$ as D^*_h and $\{y_{l_{h-1}-\tau+1}, \cdots, y_{l_{h-1}}\}$ as D^{*c}_h, and have $D_h = \{D^*_h, D^{*c}_h\}$. We take a heuristic to assume that D^{*c}_h is independent of G_h, and that $P(D^{*c}_h)$ is always equal to 1. Similarly we denote $y_{t1:t2}$ as $D^*_{t1:t2}$, $y_{t1-\tau:t1-1}$ as $D^{*c}_{t1:t2}$, and $D_{t1:t2} = \{D^*_{t1:t2}, D^{*c}_{t1:t2}\}$. In each segment, there is one graph $G_h : 1 \leq h \leq m$ that dominates the segment.

With the assumption that

$$P(D^c_h){=}\begin{cases} 1 & \text{if } D^c_h = D^{*c}_h \\ 0 & \text{otherwise} \end{cases}$$

, we have Theorem 1

THEOREM 1.

$$P(D^*_h){=}P(D^*_h|D^{*c}_h){=}P(D_h) \quad (9)$$

PROOF.

$$\begin{aligned} P(D^*_h){=}\textstyle\sum_{D^c_h} P(D^*_h|D^c_h)P(D^c_h)\\ {=}P(D^*_h,D^{*c}_h){=}P(D_h){=}P(D^*_h|D^{*c}_h)P(D^{*c}_h){=}P(D^*_h|D^{*c}_h) \end{aligned}$$

□

With Theorem 1, we have

$$\begin{aligned} P(D^*_{t1:t2}|l{=}t1{-}1){=}\textstyle\sum_G P(D^*_{t1:t2}|l{=}t1{-}1,G)P(G) \quad (10)\\ {=}\textstyle\sum_G P(D^*_{t1:t2}|D^{*c}_{t1:t2},G)P(G)\\ {=}\textstyle\sum_G P(G)\int P(D^*|D^c,G,\Theta_G,T)\rho(\Theta_G|G)d\Theta_G \end{aligned}$$

Θ_G are the parameters associated with the data $D_{t1:t2}$ corresponding to G. $\rho(\Theta_G|G)$ is the probability density function of Θ_G.

The assumption of the multinomially distributed data with the Dirichlet prior leads to the same solution (BDeu metric) of the closed form expression of the marginal likelihood $P(D_{t1:t2}|l = t1-1, G)$ in Equation 7.

Similarly as bayesian networks, we use MCMC to simulate $\mathcal{M}$ for $\{G\}$. Our experimental study shows that our methods in both BNs and DBNs versions output the similar results for the distributions of change-points.

3.2 Structure Learning of Non-stationary Bayesian Networks

Given an observed time series data D, the structure learning problem of DBNs is equal to maximizing the posterior probability of the network structure G.

By the Bayes' rule, the posterior probability is expressed as the following:

$$P(G|D,T){=}\frac{P(D|G,T)P(G|T)}{P(D|T)} \quad (11)$$

Given a non-stationary time series data, we need to find a sequence of network structures $G^T = (G_1, \ldots, G_m)$, m segments, and a

transition vector L^T, the posterior probability in Equation 11 is replaced by Equation 12:

$$P(G^T,L^T,m|D,T)=\frac{P(D|G^T,L^T,m,T)P(G^T,L^T,m|T)}{P(D|T)} \quad (12)$$

$P(D|T)$ is treated as a constant, and then

$$\begin{aligned} &P(G^T,L^T,m|D,T) \quad (13)\\ &\propto P(D|G^T,L^T,m,T)P(G^T,L^T,m|T)\\ &\propto P(D|G^T,L^T,m,T)P(G^T|L^T,m,T)P(L^T|m,T)P(m|T)\end{aligned}$$

In the following discussion, we specify the formula for calculating each component of Equation 13.

We are using the same assumption in [32] that the networks change smoothly over time. We use the exponential priors on the change of network structures. We transform the form of the sequence of graph structures $G^T : G^T = (G_1, \ldots, G_m)$ into $G^T : G^T = (G_1, \triangle G_1, \ldots, \triangle G_{m-1})$, where $\triangle G_h : 1 \leq h \leq m-1$ is the change of edges between G_h and G_{h+1}. We calculate $P(G^T|m,T)$ as follows.

$$\begin{aligned} &P(G^T|L^T,m,T)=P(G_1,\triangle G_1,\ldots,\triangle G_{m-1}) \quad (14)\\ &\propto P(G_1)\prod_{h=1}^{m-1} e^{-\lambda_s s_h} \propto P(G_1)e^{-\lambda_s \sum_{h=1}^{m-1} s_h} \propto P(G_1)e^{-\lambda_s S}\end{aligned}$$

,where $S : S = \sum_{h=1}^{m-1} s_h$, and s_h is the number of edge change between G_{h+1} and G_h. We have no prior knowledge on $P(G_1)$ and see the uniform distribution as the prior.

We assume that the data are complete and multinomially distributed with a Dirichlet prior on the parameters. We calculate $P(D_h|G_h,T)$ of each segment by following Equation 7:

$$\begin{aligned} &P(D_h|G_h,T) \quad (15)\\ &=\int P(D_h|G_h,\Theta_{G_h},T)\rho(\Theta_{G_h}|G_h)d\Theta_{G_h}\\ &=\prod_{i=1}^{n}\prod_{j=1}^{q_{ih}}\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij}(I_h))}\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk}+N_{ijk})(I_h)}{\Gamma(\alpha_{ijk})}\end{aligned}$$

We denote I_h as the segment h, Θ_{G_h} as the parameters corresponding to G_h, r_i as the number of possible values of x_i, and q_{ih} as the number of configurations of parents π_i in I_h. We let α_{ijk} and α_{ij} to be the hyperparameters for Dirichlet distributions applied in I_h. α_{ijk} is uniformly distributed inside I_h and set to $\alpha_{ijk} = \alpha/(r_i q_{ih})$. We set the equivalent sample size α equal to 1. We denote $N_{ijk}(I_h)$ as the times that x_i had value k in I_h and $N_{ij}(I_h) = \sum_{k=1}^{r_i} N_{ijk}(I_h)$.

THEOREM 2. *With Theorem 1 and the Markov property, the marginal likelihood $P(D|G^T,m,T)$ is expressed as below:*

$$P(D|G^T,L^T,m,T)=\prod_{h=1}^{m} P(D_h|G_h,m,T) \quad (16)$$

PROOF.

$$\begin{aligned} &P(D|G^T,L^T,m,T)\\ &=P(D_m^*|D_1^*,\cdots,D_{m-1}^*,G_m,m,T)\cdots P(D_1^*|G_1,m,T)\\ &=\prod_{h=1}^{m} P(D_h^*|D_h^{*c},G_h,m,T)=\prod_{h=1}^{m} P(D_h|G_h,m,T)\end{aligned}$$

□

With Theorem 2 and Equation 15, we get the extended BDeu metric:

$$\begin{aligned} &P(D|G^T,L^T,m,T)=\prod_{h=1}^{m} P(D_h|G_h,m,T) \quad (17)\\ &=\prod_{i=1}^{n}\prod_{h=1}^{m}\prod_{j=1}^{q_{ih}}\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij})+N_{ij}(I_h))}\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk}+N_{ijk}(I_h))}{\Gamma(\alpha_{ijk})}\end{aligned}$$

We use the perfect simulation modeling to calculate the posterior probability distributions of $P(L^T|m,T)$ and $P(m|T)$. We choose the most likely m, fix the number of segments, and have $P(m|T) = 1$. We use the sampling method to collect $\{G^T\}$ and will discuss the details in the subsequent section.

3.3 MCMC Sampling

Considering the fact that the gene expression data are usually sparse, which makes the posterior probability over structures to be diffuse [15], we choose sampling approaches rather than heuristic methods to search structural models, where a group of most likely structures could explain data better than a single one. In addition, the sampling methods also have the advantage to approximate the model space $\mathcal{M}$ for change-points simulation. We select MCMC as our sampling approach to collect G^T samples and compute the posterior probabilities of edges $\{e_{s,i,j}|1 \leq s \leq m, 1 \leq i,j \leq N\}$ in G^T. We use every single G^T sample to calculate the marginal probability $P(D_{t1:t2}|l = t1-1, G^T)$ of successive observations $y_{t1:t2}$ based on Equation 17. With a simulated sample space $\{G^T\}$ by MCMC, we get $P(D_{t1:t2}|l = t1-1)$ based on Equation 8 and further calculate the whole conditional probability distribution of change-points.

We design our algorithm based on the following considerations. First, we choose the heuristic search instead of the MCMC simulation to initialize G with only a single segment at the beginning of the algorithm. With the non-stationary nature, the data consists of multiple segments. And the possible model space and its distribution in each segment are different. In this case, MCMC may not provide a good approximation of $\mathcal{M}$ and is computationally expensive. Hence, we use the heuristic search to initializes a single G to do the perfect simulation and such change does not affect the prediction performance. In general, we take much smaller number of heuristic steps compared with MCMC, and the number of steps is proportional to the size of nodes. The detailed configurations of heuristic steps could be found in Section 4.

Second, we use KNUT move set instead of the KNKT (Known Transition Number Known Transition Time) move set [32] containing six move types, MT1-MT6 . Induced by the limitation of the initialization of a single G at the beginning, the true distributions of $P(m)$ and $P(L^T)$ are doubtful after the first round of perfect simulation. Simply using fixed change-points will distort the simulated model space. By bringing the move to shift the change-points into the move set, we allow MCMC not only to converge for L^T but also to provide a model space approximately at every time point. With this method, we improve the quality of $\mathcal{M}$ and have our algorithm converged. The procedure for our method is shown in the Algorithm as follows.

RCnsDBNs Algorithm

Input: Time series Data D, parameters p, k and λ_s
Output: $P(L^T)$, $P(m)$, and $P(\{e_{s,i,j}\})$
Begin
Use heuristic search and select a single graph G.
Run perfect simulation to sample change-points.
Calculate the distributions $P(m)$ and $P(L^T)$.
Select the most likely m and initialize G^T with G.
while $P(m)$, $P(L^T)$ and $P(\{e_{s,i,j}\})$ not converged **do**
  Run MCMC and collect the samples $\{G^T\}$.
  Simulate the change-point samples.
  Calculate the distributions $P(m)$, $P(L^T)$, and $P(\{e_{s,i,j}\})$.
  Select the most likely m and re-initialize G^T.
end while
End

4. EXPERIMENTAL STUDY AND EVALUATION

We performed all the experiments on Intel Xeon 3.2 Ghz EM64T processors with 4 GB memory. We implemented our method RCnsDBNs in Java.

We compare three approaches: 1) our approach (RCnsDBNs), 2) reversible jump Markov chain Monte Carlo Non-Stationary Dynamic Bayesian Networks (RJnsDBNs) [32], 3) Allocation Sampler Non-Stationary Dynamic Bayesian Networks (ASnsDBNs) [10, 11]. For RJnsDBNs, we use the default setting of unknown numbers and times of transitions (UNUT) in all of the data sets. RJnsDBNs is implemented in Java. ASnsDBNs is implemented in Matlab. In addition, we show the results of our method in BNs version denoted as RCnsBNs (ReCursion Non-Stationary Bayesian Networks). Both two versions of our method find very similar results on the posterior distributions of change-points. We grid-search the parameters for RCnsDBNs and RJnsDBNs for the best performance on change-point and structure estimation. For ASnsDBNs, we choose $K_{max} = 10$ for all experiments that we believe to satisfy the number of different components of the mixture vector in various data sets.

Our experimental study is based on three data sets: (i) Synthetic data set, (ii) Bone Marrow-derived Macrophages gene expression time series data (Macrophages data set), and (iii) Circadian regulation in Arabidopsis Thaliana gene expression time series data (Arabidopsis data set). We evaluate three methods from two aspects: computational performance on convergence and structure prediction accuracy.

Convergence Rate and Computational Time. ASnsDBNs with perfect simulation modeling (ASnsDBNs-PSM) [11] improves ASnsDBNs [10] on convergence. It selects parameters to give best approximation to the outputs of ASnsDBNs. Hence, we choose ASnsDBNs -PSM for computational performance comparison. We follow Grzegorczyk's work in [11] and evaluate ASnsDBNs-PSM and our method with the proportion of edges denoted by η for which potential scale reduction factors (PSRFs) [7] lies below the predefined threshold. PSRFs=1 shows perfect convergence and that PSRFs<1.1 is seen as the sufficient condition for convergence [11, 7]. $0 \leq \eta \leq 1$ and higher η values indicate better convergence.

RJnsDBNs does not output graph samples. We use the variation of edge posterior probabilities (VEPP) to measure the convergence of its output. $VEPP = \frac{1}{m \cdot N \cdot N} \sum_{s=1}^{m} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{|P(e_{s,i,j}^{I+\Delta I}) - P(e_{s,i,j}^{I})|}{P(e_{s,i,j}^{I})}$, where I is the number of iterations continuously sampling in MCMC, and $P(e_{s,i,j}^{I})$ is the posterior probability of an edge $e_{i,j}$ in the graph G_s that dominates the sth segment computed from I iterations. Once MCMC converges, $|P(e_{s,i,j}^{I+\Delta I}) - P(e_{s,i,j}^{I})| \rightarrow 0$ with $I \rightarrow +\infty$. Hence, VEPP values close to 0 indicate that a MCMC chain converges to a stationary distribution. We use a pre-defined threshold σ. When $VEPP < \sigma$, we decide that MCMC converges and calculate the computational time.

Structure Prediction Accuracy. To compare the inferred structure results from different data sets, we follow the evaluation method introduced in [15, 35, 10]. For the synthetic data set, we compare the inferred network structures with the true networks. For each real data set, we first collect gold standard reference networks as the ground truth. For the Macrophages data set, such reference networks are available in [19, 31, 10]. For the Arabidopsis data set, we collect the network information from [24, 33, 4, 28]. In case where we have ground truth network structure (the Bone Marrow data set and Arabidopsis data set), we use the area under receiver operating characteristic curve (AUROC) values to evaluate the performance.

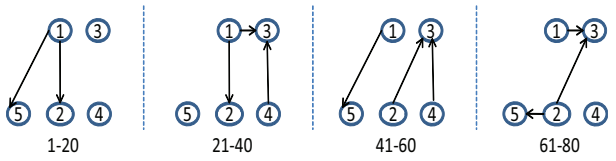

Figure 1: The synthetic networks.

In addition, for each data set, we show the posterior distribution of the number of segments and the locations of change-points. Before we discuss the details of experimental results, we present the characteristics of our data set first below.

4.1 Data Sets

We evaluate our method RCnsDBNs on a synthetic data and two gene expression data sets used in [32, 10]. We preprocess the original gene expression data sets by following Zhao's work [36]. We set the values of a missed time point with the mean of its two neighbors; i.e., $X_{i,t} = (X_{i,t-1} + X_{i,t+1})/2$ if $1 < t < T$. If the missed values are at the beginning or end, simply set the same value as its neighbor; i.e., $X_{i,t} = X_{i,t+1}$ if $t = 1$ or $X_{i,t} = X_{i,t-1}$ if $t = T$. In the following, we show the details of each data set.

Synthetic Data. We created a synthetic time series data with 80 time points and binary valued observations. It was generated by a sequence of 5 node networks with 3-4 edge changes between successive segments. The change-points of the graph structures happened at times 20, 40, 60. We showed the true networks in Figure 1.

Bone Marrow-derived Macrophages Gene Expression Data. We use the Macrophage data sets previously investigated in [10]. The data sets contain three genes, *Irf1*, *Irf2* and *Irf3*, related to Interferon regulatory factors (IRFs), proteins central to the mammalian innate immunity [14, 31]. The Macrophage data sets were sampled from different conditions: (I) Infection with Cytomegalovirus (CMV), (II) Treatment with Interferon Gamma (IFN_γ), and (III) Infection with Cytomegalovirus after pretreatment with IFN_γ (CMV + IFN_γ). Each data set has 25 time points collected with the interval 30 minutes. We follow Grzegorczyk's work [10] and use $Irf2 \leftrightarrow Irf1 \leftarrow Irf3$ as the gold standard. We assume that the network is invariant over time.

Arabidopsis Thaliana Circadian Regulation Gene Expression Data. We use the Arabidopsis Thaliana Circadian data investigated in [10]. The data sets consist of 9 genes, *LHY*, *CCA1*, *TOC1*, *ELF4*, *ELF3*, *GI*, *PRR9*, *PRR5*, and *PRR3*. The group of genes create transcriptional feedback loops and are critical to understand the internal clock-signalling network of plant. The Arabidopsis data are sampled from two light-dark conditions: (I) 10h:10h light/dark cycle and (II) 14h:14h light/dark cycle. Each data set contains 13 time points collected with the interval of 2 hours. We build a gold standard network based on the biological literatures [24, 33, 4, 28, 12, 25]. In this network, CCA1 and LHY proteins directly bind to the promoter of TOC1 to represses the expression of TOC1. The pseudo-response regulators PRR5 and PRR9 are activated by CCA1 and LHY and repress CCA1 and LHY subsequently. G1 improves the expression of TOC1. ELF4 is repressed by CCA1. For a detailed referred graph figure, please refer to our previous work [18].

4.2 Convergence and Computational Performance

We first compared the computational performance between our method RCnsDBNs and ASnsDBNs-PSM. The curves of fraction

of edges with PSRFs<1.04 on two methods for Thaliana T20 data is showed in Figure 2 and the VEPP curves in Figure 3. We calculated the PSRFs and VEPP scores from 10 independent MCMC chains. We found that RCnsDBNs and ASnsDBNs-PSM have the similar convergence rate measured in terms of MCMC sampling iterations. However, for 250,000 iterations, it takes ASnsDBNs-PSM more than **350 hours** while RCnsDBNs only needs less than **1 minute**. Even considering the fact that two algorithms are implemented in different programming languages (RCnsDBNs in java and ASnsDBNs-PSM in Matlab), compared with ASnsDBNs-PSM, our method has much better computational efficiency.

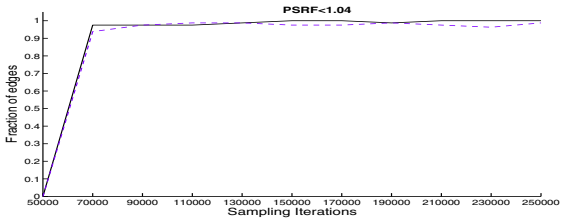

Figure 2: The curves of fraction of edges with PSRFs<1.04 on RCnsDBNs and ASnsDBNs-PSM for Thaliana T20 data. RCnsDBNs: black solid line ; ASnsDBNs-PSM: blue dashed line.

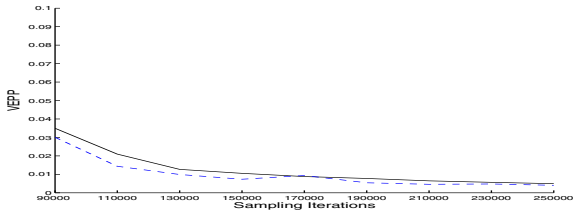

Figure 3: The VEPP curves on RCnsDBNs and ASnsDBNs-PSM for Thaliana T20 data. RCnsDBNs: black solid line ; ASnsDBNs-PSM: blue dashed line.

For the comparison between RJnsDBNs and our approach, we set $\sigma = 0.05$ for the convergence of VEPP values and listed the number of iterations and computational time in Table 1. In multiple data sets, RCnsDBNs converges much faster than RJnsDBNs. Compared with RJnsDBNs , RCnsDBNs got 6 folds computational improvement on CMV data, 6 folds on $CMV + IFN_\gamma$ data, 9 folds on IFN_γ data. On Arobidopsis microarray data, RJnsDBNs took less time than RCnsDBNs. However, it failed to detect any meaningful change-point as RCnsDBNs and ASnsDBNs did on Arobidopsis data. In addition, we showed the VEPP curves of two approaches for CMV data in Figure 4.

4.3 Stability of Results

We use $k = 1$ for all the experiments because the gene expression data usually has limited time points and larger k values eliminate short segments. The value of parameter p is adjusted for the purpose of the convergence of results for different dominant segment numbers m. We grid-search the values of p between $0.00001 \sim 0.5$ for the effective range on the preferred segmentation. For the synthetic data, it has four segments ($m = 4$). For three Macrophages data, we selected $m = 1$ based on the assumption of a single IRFs network structure with varying parameters [10]. For Arobidopsis T20 data, most of the p range leads to $m = 1$. Finally,

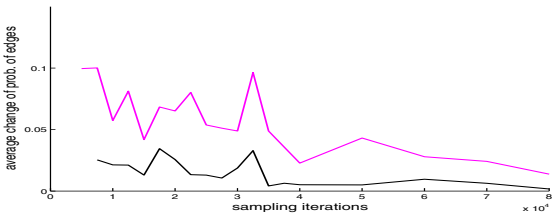

Figure 4: The VEPP curves for CMV data. RCnsDBNs ($p = 0.01, \lambda_s = 2$): black solid line ; RJnsDBNs ($\lambda_m = 0.65, \lambda_s = 2$): magenta dash-dot line.

for Arobidopsis T28 data, we chose $m = 2$ with the consideration of the external light/dark cycle condition.

Table 2: The effective range of parameter p for RCnsDBNs

	effective parameter range of p
Synthetic (m=4)	$0.02 \sim 0.032$
CMV (m=1)	≤ 0.009
CMV+IFN$_\gamma$ (m=1)	≤ 0.0006
IFN$_\gamma$ (m=1)	≤ 0.0001
Arobidopsis T20 (m=1)	≤ 0.5
Arobidopsis T28 (m=2)	$0.18 \sim 0.23$

4.4 Structure Prediction and Change-point Detection

In the following, we will show the results of predicted structures and detected change-points.

The results on synthetic data. We compared two discrete models, RCnsDBNs and RJnsDBNs, on synthetic data. RCnsDBNs totally runs 16 rounds to get converged, and each round uses 5,000 iterations for burn-in and then takes additional 20,000 iterations to collect samples ; RJnsDBNs runs 100,000 iterations for burn-in and then takes additional 400,000 iterations to collect samples. RCnsDBNs initializes G with additional 1000 heuristic search steps.

We showed the predicted posterior distributions on the numbers of segments and change-points in Figure 5. RCnsDBNs correctly identified 4 segments and its predicted change-points are close to the true times at 20, 40, and 60 while RJnsDBNs failed to identify meaningful change-points. The AUROC scores of predicted structures by RCnsDBNs is showed in Table 3. When the equivalence class of bayesian network structures [2] were considered, the AUROC scores of all segments were increased, which were shown in the same table.

The results on Macrophages data. On the CMV Macrophages data, RCnsDBNs totally runs 4 rounds to get converged, and each round uses 500 iterations for burn-in and then takes additional 2,000 iterations to collect samples ; RJnsDBNs runs 10,000 iterations for burn-in and then takes additional 40,000 iterations to collect samples. On the $CMV + IFN_\gamma$ Macrophages data, RCnsDBNs totally runs 5 rounds to get converged, and each round uses 1,000 iterations for burn-in and then takes additional 4,000 iterations to collect samples ; RJnsDBNs runs 26,000 iterations for burn-in and then take additional 104,000 iterations to collect samples. On the IFN_γ Macrophages data, RCnsDBNs totally runs 7 rounds to get converged, and each round uses 500 iterations for burn-in and then takes additional 2,000 iterations to collect samples ; RJnsDBNs runs 16,000 iterations for burn-in and then take additional 64,000 iterations to collect samples. In both data sets, ASnsDBNs runs

Table 1: The comparison of computational performance

N	RJnsDBNs	I	C_T	RCnsDBNs	I	T	I	C_T	speedup
5	$Synthetic\ Data$ $(\lambda_m =, \lambda_s = 2)$	$500,000$	$34.28m$	$Synthetic\ Data$	$500,000$	$17.91m$	$400,000$	$14.36m$	2.39
3	CMV $(\lambda_m = 0.65, \lambda_s = 2)$	$50,000$	$7.316s$	CMV	$50,000$	$4.333s$	$10,000$	$1.127s$	6.49
3	IFN_γ $(\lambda_m = 0.001, \lambda_s = 2)$	$80,000$	15.962	IFN_γ	$80,000$	$6.615s$	$17,500$	$1.646s$	9.70
3	$CMV + IFN_\gamma$ $(\lambda_m = 1, \lambda_s = 2)$	$130,000$	$26.152s$	$CMV + IFN_\gamma$	$130,000$	$16.223s$	$25,000$	$4.029s$	6.49
9	$ArobidopsisT20$ $(\lambda_m = 0.0005, \lambda_s = 2)$	$30,000$	$6.274s$	$ArobidopsisT20$	$50,000$	$5.363s$	$100,000$	$10.314s$	0.61
9	$ArobidopsisT28$ $(\lambda_m = 0.005, \lambda_s = 2)$	$30,000$	$6.048s$	$ArobidopsisT28$	$50,000$	$15.26s$	$100,000$	$28.325s$	0.21

N is the number of genes in the data sets. I is the number of iterations. T is the computational time. C_T is the computational time for convergence. $\sigma = 0.05$ is used to decide the convergence of the results.

Table 3: The AUROC values of RCnsDBNs on synthetic data

	Synthetic Data
RCnsBNs	$G1:1; G2:0.6078;$ $G3:0.4706; G4:0.6078$
RCnsDBNs (equivalence class considered)	$G1:0.9688; G2:0.6719;$ $G3:0.5938; G4:0.6406$

TP, true positive; FP, false positive; TN, true negative; FN, false negative.
Sensitivity = TP/(TP+FN).
Specificity = TN/(TN+FP).
Complementary Specificity = 1- Specificity = FP/(TN+FP).
The ROC curves are plotted with the Sensitivity scores against the corresponding Complementary Specificity scores.

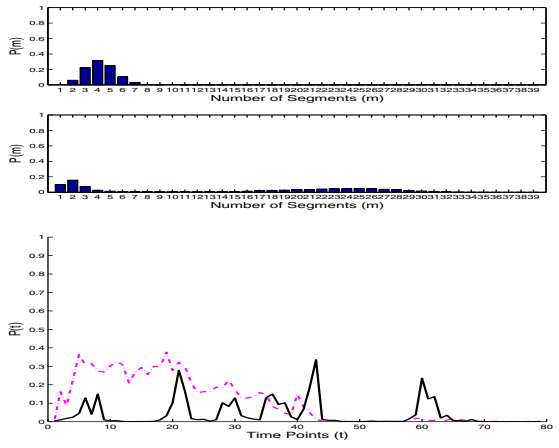

Figure 5: Comparison of two methods on the synthetic data. Up: The posterior probabilities of the numbers of segments P(m) (top: RCnsDBNs ($p = 0.031, \lambda_s = 0.5$); bottom: RJnsDBNs ($\lambda_m = 0.4, \lambda_s = 0.5$)). Low: The posterior probabilities of the change points P(t) (RCnsDBNs: black solid line; RJnsDBNs: magenta dash-dot line).

10,000 iterations for burn-in and then take additional 40,000 iterations to collect samples. RCnsDBNs initializes G with additional 100 heuristic search steps.

In Figure 6, 7, and 8, we show the posterior probabilities of the numbers of segments and change-points on Macrophages data sets.

For the CMV data, we observe that ASnsDBNs clearly identifies a dominant 3-segment in the data set while the posterior probabilities produced by RJnsDBNs are almost flat. There is a consensus among three methods that the most probable change-point occurs around the location 5. The results of three methods are consistent with the biological phenomenon that the simultaneous responses of Macrophages happen under the attack of Cytomegalovirus [10]. In order to assess the network prediction performance, we show the AUROC scores in Table 2. We find that all methods perform well in the CMV data with the AUROC scores equal to 1.

For the $CMV + IFN_\gamma$ data, both RJnsDBNs and ASnsDBNs methods identify 1 segment, which [10] explained as a coexistence state between virus and its host cell [1, 10]. And their posterior probabilities are flat. Different from these two methods, RCnsDBNs found two posterior peaks at 3 and 8. Such finding indicates the coexistence state may not happen at the beginning under both the IFN_γ treatment and invasion of virus. In Table 4, we find that RCnsDBNs and ASnsDBNs show a much better network prediction with the AUROC score equal to 0.6667 while in RJnsDBNs the AUROC score is equal to 0.2222.

For the IFN_γ data, there is a postulated transition with the immune activation under the treatment of IFN_γ. Both RJnsDBNs and ASnsDBNs infer 1 segments. RJnsDBNs and ASnsDBNs identify a same posterior peak at the location around 5. RCnsDBNs finds two posterior peaks of transition time at 9 and 13. On the assessment of the predicted network structures, the AUROC scores are 0.7778 in RCnsDBNs and RJnsDBNs, and 0.6667 in ASnsDBNs.

For each Macrophages data set using RCnsDBNs and RJnsDBNs methods, we find that the posterior probability distributions of any edge do not change much across different segments. This finding is consistent with the assumption that the underlying network structure does not change through the time.

The results on Arabidopsis data. On the Arabidopsis data, RCnsDBNs totally runs 10 rounds to get converged, and each round uses 5,000 iterations for burn-in and then takes additional 20,000 iterations to collect samples ; RJnsDBNs runs 6,000 iterations for burn-in and then take additional 24,000 iterations to collect sam-

Table 4: Comparison of AUROC values on Macrophage data

	CMV	IFN_γ	$CMV + IFN_\gamma$
RJnsDBNs	1	0.7778	0.2222
ASnsDBNs	1	0.6667	0.6667
RCnsBNs	1	0.5556	0.6667
RCnsDBNs	1	0.7778	0.6667

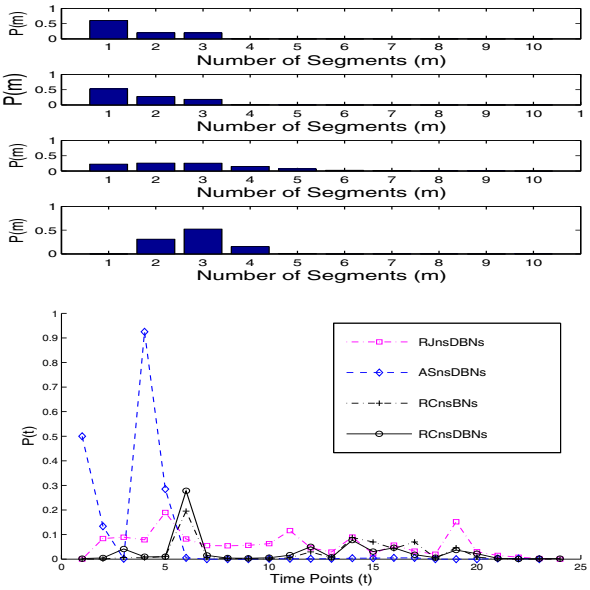

Figure 6: Comparison of four methods on CMV Macrophage data. Up: The posterior probabilities of the numbers of segments P(m) (from the top to the bottom: RCnsBNs ($\lambda_s = 2$), RCnsDBNs ($\lambda_s = 2$), RJnsDBNs ($\lambda_m = 0.65, \lambda_s = 2$), and ASnsDBNs). Low: The posterior probabilities of the change points P(t).

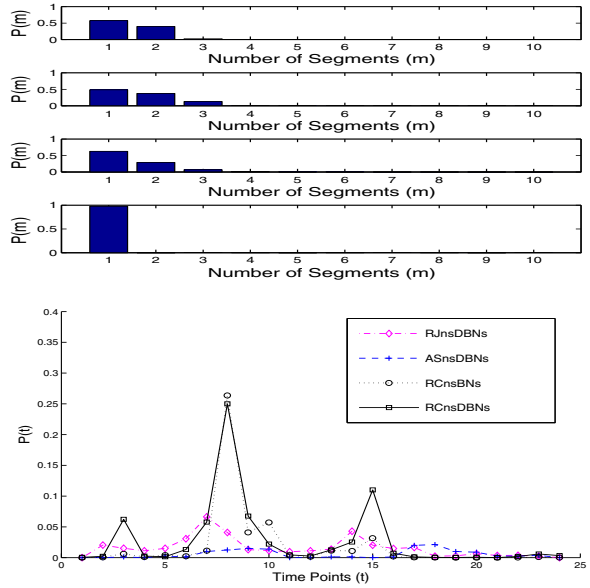

Figure 7: Comparison of four methods on $CMV + IFN_\gamma$ Macrophage data. Up: The posterior probabilities of the numbers of segments P(m) (from the top to the bottom: RCnsBNs ($\lambda_s = 2$), RCnsDBNs ($\lambda_s = 2$), RJnsDBNs ($\lambda_m = 1, \lambda_s = 2$), and ASnsDBNs). Low: The posterior probabilities of the change points P(t).

ples; ASnsDBNs runs 990,000 iterations for burn-in and then take additional 10,000 iterations to collect samples. Considering larger size of variables and thereafter the larger model space compared

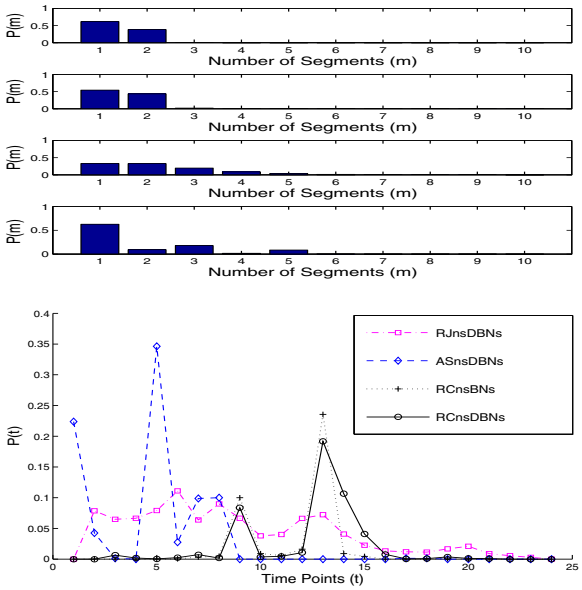

Figure 8: Comparison of four methods on IFN_γ Macrophage data. Up: The posterior probabilities of the numbers of segments P(m) (from the top to the bottom: RCnsBNs ($\lambda_s = 2$), RCnsDBNs ($\lambda_s = 2$), RJnsDBNs ($\lambda_m = 0.001, \lambda_s = 2$), and ASnsDBNs). Low: The posterior probabilities of the change points P(t).

Table 5: Comparison of AUROC values on Arabidopsis data

	Arabidopsis T20	Arabidopsis T28
RJnsDBNs	0.5035	0.3893
ASnsDBNs	0.5929	0.5641
RCnsBNs	$G1: 0.4856$	$G1: 0.4856; G2: 0.5315$
RCnsDBNs	$G1: 0.5183$	$G1: 0.5925; G2: 0.5979$

with other two data sets, RCnsDBNs takes more heuristic search steps and initializes G with additional 10000 heuristic iterations.

In Figure 9 and 10, we show the posterior distributions of the numbers of segments and changepoints on two Arabidopsis data sets. For the Arabidopsis T20 data, the dominant samples in RJnsDBNs and ASnsDBNs are respectively 2 and 3 segments. For the Arabidopsis T28 data, RJnsDBNs infers 1 segment and ASnsDBNs infers 5 segments. In both data sets, we find that the difference between the posterior peaks of changepoints and the time points nearby in RJnsDBNs are not noticeable. Hence, for this data set, we only use a single network in RJnsDBNs to compare with other methods. Using ASnsDBNs, the posterior peaks of change-points on T20 data are 1, 5 and those on T28 are 2, 7, 10. In [10], the results of ASnsDBNs are explained as a phase shift incurred by different dark/light cycles. Our method RCnsDBNs had the same finding by identifying the peaks at 5, 7, and 10 on T20 data and the peaks at 2, 6, and 9 on T28 data. And in addition, RCnsdBNs finds a peak around 10 on T20 data. This time point is exactly the beginning of the new light/dark cycle.

We evaluated the network reconstruction accuracy of three methods by comparing with the reference network showed in Section 4.1. We show the AUROC scores in Table 5. Our method outperforms RJnsDBNs in both datasets and has competitive performance on structure prediction accuracy against ASnsDBNs.

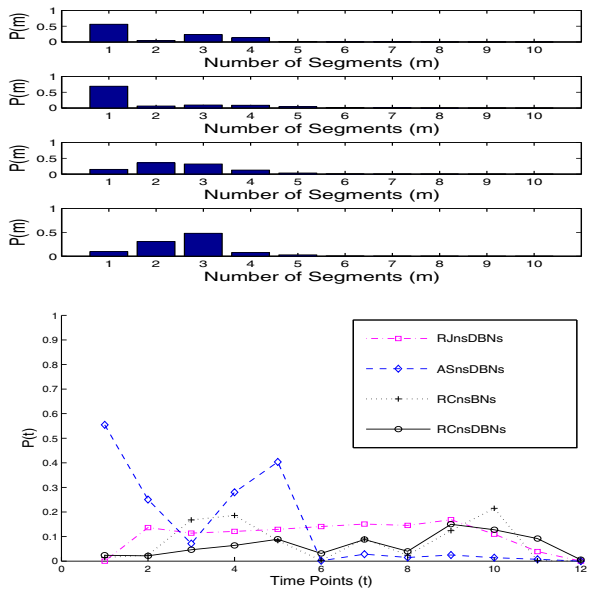

Figure 9: Comparison of four methods on Arabidopsis T20 data. Up: The posterior probabilities of the numbers of segments P(m) (from the top to the bottom: RCnsBNs ($\lambda_s = 2$), RCnsDBNs ($\lambda_s = 2$), RJnsDBNs ($\lambda_m = 0.0005, \lambda_s = 2$), and ASnsDBNs). Low: The posterior probabilities of the change points P(t).

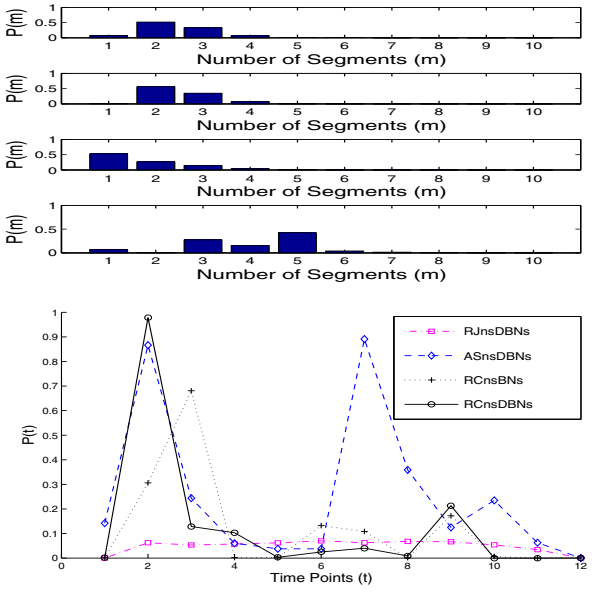

Figure 10: Comparison of four methods on Arabidopsis T28 data. Up: The posterior probabilities of the numbers of segments P(m) (from the top to the bottom: RCnsBNs ($\lambda_s = 2$), RCnsDBNs ($\lambda_s = 2$), RJnsDBNs ($\lambda_m = 0.005, \lambda_s = 2$), and ASnsDBNs). Low: The posterior probabilities of the change points P(t).

5. CONCLUSION

In this paper we introduced a new computationally efficient non-stationary DBNs method based on perfect simulation model. We applied this approach for the network inference on one synthetic data and two non-stationary time series microarray gene expression data. The experimental results demonstrated that our method outperformed two other state-of-the-art methods in both computational cost and structure prediction accuracy. The further sensitivity analysis showed that once converged our model is insensitive to the parameter, which reduces the uncertainty of the model behavior.

6. ACKNOWLEDGEMENTS

We are grateful to Dr. Marco Grzegorczy at TU Dortmund University and Joshua W Robinson at Duke University to provide the data sets and softwares. This work is partially supported by NSF IIS award 0845951.

7. REFERENCES

[1] Chris A. Benedict, Theresa A. Banks, Lionel Senderowicz, Mira Ko, William J. Britt, Ana Angulo, Peter Ghazal, and Carl F. Ware. Lymphotoxins and cytomegalovirus cooperatively induce interferon-b establishing host-virus detente. *Immunity*, 15:617–626, 2001.

[2] David Maxwell Chickering. Learning equivalence classes of bayesian network structures. *Journal of Machine Learning Research*, 2:445–498, 2002.

[3] N. Chopin. Dynamic detection of change points in long time series. *The Annals of the Institute of Statistical Mathematics*, 2006.

[4] Michael F. Covington, Satchidananda Panda, Xing Liang Liu, Carl A. Strayer, D. Ry Wagner, and Steve A. Kay. Elf3 modulates resetting of the circadian clock in arabidopsis. *The Plant Cell*, 13:1305–1315, 2001.

[5] Frank Dondelinger, Sophie Lebre, and Dirk Husmeier. Heterogeneous continuous dynamic bayesian networks with flexible structure and inter-time segment information sharing. In *Proceedings of 2010 International Conference on Machine Learning (ICML10)*, 2010.

[6] Paul Fearnhead. Exact and efficient bayesian inference for multiple changepoint problems. *Statistics and Computing*, 16:203–213, 2006.

[7] Andrew Gelman and Donald B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457ÍC472, 1992.

[8] Peter J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732, 1995.

[9] Peter J. Green. Trans-dimensional markov chain monte carlo. *Highly Structured Stochastic Systems, Oxford University Press*, 2003.

[10] Marco Grzegorczy, Dirk Husmeier, Kieron D. Edwards, Peter Ghazal, , and Andrew J. Millar. Modelling non-stationary gene regulatory processes with a non-homogeneous bayesian network and the allocation sampler. *Bioinformatics*, 24:2071 – 2078, 2008.

[11] Marco Grzegorczyk and Dirk Husmeier. Improvements in the reconstruction of time-varying gene regulatory networks: dynamic programming and regularization by information sharing among genes. *Journal of Bioinformatics*, 27(5):693–699, 2011.

[12] Anthony Hall, Laszlo Kozma-Bognar, RekaToth, Ferenc Nagy, and Andrew J. Millar. Conditional circadian regulation of phytochrome a gene expression. *Plant Physiol.*, 127(4):1808–18, 2001.

[13] David Heckerman, Dan Geiger, and David Maxwell Chickering. Learning bayesian networks: The combination

of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

[14] Kenya Honda, Akinori Takaoka, and Tadatsugu Taniguchi. Type i interferon gene induction by the interferon regulatory factor family of transcription factors. *Immunity*, 25:349–360, 2006.

[15] Dirk Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks. *Bioinformatics*, 19:2271–2282, 2003.

[16] Dirk Husmeier, Frank Dondelinger, and Sophie Lebre. Inter-time segment information sharing for non-homogeneous dynamic bayesian networks. In *Proceedings of 2010 Advances in Neural Information Processing Systems (NIPS)*, 2010.

[17] Katja Ickstadt, Bjorn Bornkamp, Marco Grzegorczyk, Jakob Wieczorek, M. Rahuman Sheriff, Hernan E. Grecco, and Eli Zamir. Nonparametric bayesian networks. *In: Bernardo, Bayarri, Berger, Dawid, Heckerman, Smith, and West (eds): Bayesian Statistics 9, Oxford University Press*, 2010.

[18] Yi Jia and Jun Huan. Constructing non-stationary dynamic bayesian networks with a flexible lag choosing mechanism. *BMC Bioinformatics*, 2010.

[19] JE Darnell Jr, IM Kerr, and GR Stark. Jak-stat pathways and transcriptional activation in response to ifns and other extracellular signaling proteins. *Science*, 264:1415–1421, 1994.

[20] Yoshinobu Kawahara and Masashi Sugiyama. Change-point detection in time-series data by direct densityratio estimation. In *Proceedings of 2009 SIAM International Conference on Data Mining (SDM09)*, 2009.

[21] M. Lavielle and G. TeyssiÍĺree. Adaptive detection of multiple change-points in asset price volatility. *Long Memory in Economics*, page 129ÍC156, 2005.

[22] Sophie Lebre, Jennifer Becq, Frederic Devaux, Michael Stumpf, and Gaelle Lelandais. Statistical inference of the time-varying structure of gene regulation networks. *BMC Systems Biology*, 4(1):130, 2010.

[23] Robert Lund and Jaxk Reeves. Detection of undocumented changepoints: a revision of the two-phase regression model. *J. Climate*, 2002.

[24] Paloma Mas. Circadian clock function in arabidopsis thaliana: time beyond transcription. *Trends Cell Biology*, 18:273–181, 2008.

[25] Takeshi Mizuno and Norihito Nakamichi. Pseudo-response regulators (prrs) or true oscillator components (tocs). *Plant Cell Physiol.*, 46(5):677–685, 2005.

[26] V. Moskvina and A. Zhigljavsky. An algorithm based on singular spectrum analysis for changepoint detection. *Communications in Statistics Part B: Simulation and Computation*, 2003.

[27] Agostino Nobile and Alastair T. Fearnside. Bayesian finite mixtures with an unknown number of components: The allocation sampler. *Statistics and Computing*, 17:147–162, 2007.

[28] Alessia Para, Eva M. Farre, Takato Imaizumi, Jose L. Pruneda-Paz, Franklin G. Harmon, and Steve A. Kay. Prr3 is a vascular regulator of toc1 stability in the arabidopsis circadian clock. *Plant Cell*, 19(11):3462–73, 2007.

[29] A. Pievatolo and Peter J. Green. Bounary detection through dynamic polygons. *Journal of the Royal Statistical Society*, Series B 60:609–626, 1998.

[30] Arvind Rao, Alfred O. Hero III, David J. States, and James Douglas Engel. Inferring time-varying network topologies from gene expression data. *EURASIP Journal on Bioinformatics and Systems Biology*, 2007.

[31] Sobia Raza, Kevin A Robertson, Paul A Lacaze, David Page, Anton J Enright, Peter Ghazal, and Tom C Freeman. A logic-based diagram of signalling pathways central to macrophage activation. *BMC System Biology*, 2:Article 36, 2008.

[32] Joshua W Robinson and Alexander J Hartemink. Non-stationary dynamic bayesian networks. *Procedding of Advances in Neural Information Processing Systems Conference*, 2008.

[33] Patrice A. Salome and C. Robertson McClung. The arabidopsis thaliana clock. *Journal of Biological Rhythms*, 19(5):425–435, 2004.

[34] Le Song, Mladen Kolar, and Eric Xing. Time-varying dynamic bayesian networks. *NIPS*, 2009.

[35] Adriano V. Werhli, Marco Grzegorczyk, and Dirk Husmeier. Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks. *Bioinformatics*, 20(22):2523–2531, 2006.

[36] Wentao Zhao, Erchin Serpedin, and Edward R. Dougherty. Inferring gene regulatory networks from time series data using the minimum description length principle. *Bioinformatics*, 22(17):2129–2135, 2006.