BIOINFORMATICS ORIGINAL PAPER

Vol. 27 no. 5 2011, pages 693–699
doi:10.1093/bioinformatics/btq711

Systems biology

Advance Access publication December 21, 2010

Improvements in the reconstruction of time-varying gene regulatory networks: dynamic programming and regularization by information sharing among genes

Marco Grzegorczyk[1,*] and Dirk Husmeier[2,*]

[1]Department of Statistics, TU Dortmund University, Dortmund, Germany and [2]Biomathematics and Statistics Scotland (BioSS), Edinburgh, UK

Associate Editor: Joaquin Dopazo

ABSTRACT

Method: Dynamic Bayesian networks (DBNs) have been applied widely to reconstruct the structure of regulatory processes from time series data, and they have established themselves as a standard modelling tool in computational systems biology. The conventional approach is based on the assumption of a homogeneous Markov chain, and many recent research efforts have focused on relaxing this restriction. An approach that enjoys particular popularity is based on a combination of a DBN with a multiple changepoint process, and the application of a Bayesian inference scheme via reversible jump Markov chain Monte Carlo (RJMCMC). In the present article, we expand this approach in two ways. First, we show that a dynamic programming scheme allows the changepoints to be sampled from the correct conditional distribution, which results in improved convergence over RJMCMC. Second, we introduce a novel Bayesian clustering and information sharing scheme among nodes, which provides a mechanism for automatic model complexity tuning.

Results: We evaluate the dynamic programming scheme on expression time series for *Arabidopsis thaliana* genes involved in circadian regulation. In a simulation study we demonstrate that the regularization scheme improves the network reconstruction accuracy over that obtained with recently proposed inhomogeneous DBNs. For gene expression profiles from a synthetically designed *Saccharomyces cerevisiae* strain under switching carbon metabolism we show that the combination of both: dynamic programming and regularization yields an inference procedure that outperforms two alternative established network reconstruction methods from the biology literature.

Availability and implementation: A MATLAB implementation of the algorithm and a supplementary paper with algorithmic details and further results for the Arabidopsis data can be downloaded from: http://www.statistik.tu-dortmund.de/bio2010.html

Contact: grzegorczyk@statistik.tu-dortmund.de; dirk@bioss.ac.uk

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on September 9, 2010; revised on October 26, 2010; accepted on December 16, 2010

*To whom correspondence should be addressed.

1 INTRODUCTION

Two paradigm shifts have revolutionized molecular biology in the second half of this decade: systems biology, where the objective is to model the whole complexity of cellular processes in a holistic sense, and synthetic biology, which enables biologists to build new molecular pathways *in vivo*, i.e. in living cells. The combination of both concepts allows the viability of machine learning approaches for network reconstruction to be tested in a rigorous way. As an alternative to mechanistic models (Cao and Ren, 2008; Wang *et al.*, 2010; Wilkinson, 2006; Xiao and Cao, 2008), which provide a powerful approach to the modelling of small systems composed of a few components, and correlation/mutual information based approaches, which do not distinguish between direct and indirect interactions (Butte and Kohane, 2000), dynamic Bayesian networks (DBNs) have emerged as a promising trade-off between over-simplicity and loss of computational tractability (Cantone *et al.*, 2009). The standard assumption underlying DBNs is that of homogeneity: temporal processes and the time-series they generate are assumed to be governed by a homogeneous Markov relation. However, regulatory interactions and signal transduction processes in the cell are usually adaptive and change in response to external stimuli. Following earlier approaches aiming to relax the homogeneity assumption for undirected graphical models (Talih and Hengartner, 2005; Xuan and Murphy, 2007), various recent research efforts have therefore addressed the homogeneity assumption for DBNs. An approach that has become popular recently is based on a combination of a DBN with a multiple changepoint process, and the application of a Bayesian inference scheme via reversible jump Markov chain Monte Carlo (RJMCMC). Robinson and Hartemink (2009) proposed a discrete inhomogeneous DBN, which allows for different structures in different segments of the time series, with a regularization term penalizing differences among the structures. Grzegorczyk and Husmeier (2009) proposed a continuous inhomogeneous DBN, in which the parameters are allowed to vary, while a common network structure provides information sharing among the time series segments. Lèbre (2007) and Lèbre *et al.* (2010) proposed an alternative continuous inhomogeneous DBN, which is more flexible in that it allows the network structure to vary among the segments. The model proposed in Kolar *et al.* (2009) is a close cousin of an inhomogeneous DBN. As opposed to the first three approaches, (hyper-)parameters are not consistently inferred within

the Bayesian context, though, and these methods will therefore not be further considered here.

Instead, we will focus on the Bayesian inference scheme common to the first three approaches. All three methods adopt an RJMCMC scheme for inferring the number and location of changepoints, based on changepoint birth, death and relocation moves. In the present article we show that the number and location of changepoints can be sampled from the proper conditional distribution. This is effected by a modification of the dynamic programming scheme proposed in Fearnhead (2006) in the context of Bayesian mixture models. We discuss the trade-off between computational up-front costs and improvement in mixing and convergence, and we empirically quantify the net gain in computational efficiency in dependence on certain features of the prior distribution.

The above mentioned inhomogeneous DBNs can be divided into two classes according to whether changepoints are common to the whole network (*class 1*), or varying from node to node (*class 2*). The approach of *class 1*, pursued in Grzegorczyk *et al.* (2008), Robinson and Hartemink (2009)[1] and Grzegorczyk *et al.* (2010), is over-restrictive, as it does not allow for individual nodes to be affected by changing processes in different ways. The approach of *class 2*, pursued in Grzegorczyk and Husmeier (2009), Lèbre (2007) and Lèbre *et al.* (2010) is potentially over-flexible, as it does not provide any information sharing among the nodes. When an organism undergoes transitional changes, e.g. morphogenic transitions during embryogenesis, one would expect the majority of genes to be affected by these transitions in identical ways. However, there is no mechanism in the fully flexible model that incorporates this prior notion of commonality. In the present article, we explore a Bayesian clustering scheme akin to the weight sharing principle in neural computation (Nowlan and Hinton, 1992), by which we assign nodes to clusters that are characterized by common changepoints. We demonstrate that our scheme subsumes the aforementioned approaches as limiting cases, and that it automatically identifies the right trade-off between them in a data-driven manner.

2 METHOD

2.1 Review of time-dependent DBNs

DBNs are flexible models for representing probabilistic relationships among interacting variables (nodes) $X_1,\ldots,X_N$ via a directed graph $\mathcal{G}$. The parent node set of node X_n in $\mathcal{G}$, $\pi_n=\pi_n(\mathcal{G})$, is the set of all nodes from which an edge points to node X_n in $\mathcal{G}$. Consider a dataset $\mathcal{D}$, where $\mathcal{D}_{n,t}$ and $\mathcal{D}_{(\pi_n,t)}$ are the t-th realizations $X_n(t)$ and $\pi_n(t)$ of X_n and π_n, respectively, and $1\leq t\leq m$ represents time. A DBN is based on a (first-order) Markov process, which is determined by the conditional probabilities $P(X_n(t)=\mathcal{D}_{n,t}|\pi_n(t-1)=\mathcal{D}_{(\pi_n,t-1)},\boldsymbol{\theta}_n)$. Common choices are a multinomial distribution, as used in Robinson and Hartemink (2009), or a linear Gaussian distribution, as in Grzegorczyk and Husmeier (2009), with corresponding (node-specific) parameter vectors $\boldsymbol{\theta}_n$. An inhomogeneous generalization of the standard first-order homogeneous DBN was proposed (e.g. see Grzegorczyk and Husmeier, 2009; Lèbre, 2007; Lèbre *et al.*, 2010; Robinson and Hartemink, 2009) and is given by

$$P(\mathcal{D}|\mathcal{G},\mathbf{V},\mathbf{K},\boldsymbol{\theta})=\prod_{n=1}^{N}\prod_{t=2}^{m}\prod_{k=1}^{\mathcal{K}_n}\psi(\mathcal{D}_n^{\pi_n}[t,\boldsymbol{\theta}_n^k])^{\delta_{\mathbf{V}_n(t),k}} \tag{1}$$

$$\psi(\mathcal{D}_n^{\pi_n}[t,\boldsymbol{\theta}_n^k])=P\Big(X_n(t)=\mathcal{D}_{n,t}|\pi_n(t-1)=\mathcal{D}_{(\pi_n,t-1)},\boldsymbol{\theta}_n^k\Big)$$

where $\delta_{\mathbf{V}_n(t),k}$ is the Kronecker delta, $\mathbf{V}$ is a matrix of latent variables $\mathbf{V}_n(t)$, $\mathbf{V}_n(t)=k$ indicates that the realization of node X_n at time t, $X_n(t)$, has been generated by the k-th component of a mixture with $\mathcal{K}_n$ components, and $\mathbf{K}=(\mathcal{K}_1,\ldots,\mathcal{K}_n)$. Let $P(\boldsymbol{\theta}|\mathcal{G},\mathbf{K})=\prod_{n=1}^{N}\prod_{k=1}^{\mathcal{K}_n}P(\boldsymbol{\theta}_n^k|\pi_n)$ denote the (conjugate) prior distribution of the parameters. Under fairly weak conditions satisfied (and discussed) in Lèbre (2007), Robinson and Hartemink (2009), Grzegorczyk and Husmeier (2009) and Lèbre *et al.* (2010), the following integral can be solved analytically:

$$\Psi(\mathcal{D}_n^{\pi_n}[k,\mathbf{V}_n])=\int\left(\prod_{t=2}^{m}\psi(\mathcal{D}_n^{\pi_n}[t,\boldsymbol{\theta}_n^k])^{\delta_{\mathbf{V}_n(t),k}}\right)P(\boldsymbol{\theta}_n^k|\pi_n)d\boldsymbol{\theta}_n^k \tag{2}$$

which yields a closed-form expression for the marginal likelihood:

$$\begin{aligned}P(\mathcal{D}|\mathcal{G},\mathbf{V},\mathbf{K})&=\int P(\mathcal{D}|\mathcal{G},\mathbf{V},\mathbf{K},\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathcal{G},\mathbf{K})d\boldsymbol{\theta} \\ &=\prod_{n=1}^{N}\prod_{k=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n}[k,\mathbf{V}_n]):=\prod_{n=1}^{N}\Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathcal{K}_n,\mathbf{V}_n])\end{aligned} \tag{3}$$

The objective of Bayesian inference is to sample the network structure $\mathcal{G}$, the latent variables $\mathbf{V}=(\mathbf{V}_1,\ldots,\mathbf{V}_N)$, and the node-specific numbers of segments $\mathbf{K}=(\mathcal{K}_1,\ldots,\mathcal{K}_N)$ from the posterior distribution $P(\mathcal{G},\mathbf{V},\mathbf{K}|\mathcal{D})\propto P(\mathcal{G},\mathbf{V},\mathbf{K},\mathcal{D})$, where

$$\begin{aligned}P(\mathcal{G},\mathbf{V},\mathbf{K},\mathcal{D})&=P(\mathcal{G})P(\mathbf{V}|\mathbf{K})P(\mathbf{K})P(\mathcal{D}|\mathcal{G},\mathbf{V},\mathbf{K}) \\ &=\prod_{n=1}^{N}P(\pi_n)P(\mathbf{V}_n|\mathcal{K}_n)P(\mathcal{K}_n)\Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathcal{K}_n,\mathbf{V}_n])\end{aligned} \tag{4}$$

In Grzegorczyk and Husmeier (2009) and Lèbre (2007), a truncated Poisson prior is chosen for $P(\mathcal{K}_n)$, and a multiple changepoint process prior for $P(\mathbf{V}_n|\mathcal{K}_n)$. The approach in Grzegorczyk *et al.* (2010) is similar, except that the allocations of time points to components are not node-specific (i.e. $\mathcal{K}_n$ and $\mathbf{V}_n$ do not depend on n); see above (*class 1* vs. *2*).

2.2 Improved Gibbs sampling based on dynamic programming

To sample from the posterior distribution, $P(\mathcal{G},\mathbf{V},\mathbf{K}|\mathcal{D})$, all previous studies (Grzegorczyk and Husmeier, 2009; Grzegorczyk *et al.*, 2010; Lèbre *et al.*, 2010; Robinson and Hartemink, 2009) follow the same procedure: to sample the network structure $\mathcal{G}$, they follow Madigan and York (1995) and apply Metropolis–Hastings (MH) structure Markov chain Monte Carlo (MCMC), based on single-edge operations; to sample the latent variables $(\mathbf{V},\mathbf{K})$, they follow Green (1995) and apply RJMCMC, based on changepoint birth, death and reallocation moves. In the present study, we propose an improved scheme based on dynamic programming. The idea is to adapt the method proposed by Fearnhead (2006) in the context of Bayesian mixture models to inhomogeneous DBNs of the form defined in Equation (1). Fearnhead (2006) assumes that the changepoints occur at discrete time points, and he considers two priors for the changepoints. The first prior is based on a prior for the number of changepoints, and then a conditional prior on their positions. This corresponds exactly to $P(\mathcal{K}_n)$ and $P(\mathbf{V}_n|\mathcal{K}_n)$, as discussed above. The second prior is obtained from a point process on the positive and negative integers. The point process is specified by the probability mass function $g(t)$ for the time between two successive points, for which a natural choice is the negative binomial distribution

$$g(t|a,p)=\binom{t-a}{a-1}p^a(1-p)^{t-a} \tag{5}$$

whose form is defined by two hyperparameters, a and p. The choice of this prior immediately imposes a prior distribution on the latent variables

[1]Changepoints in Robinson and Hartemink (2009) apply, in the first instance, to the whole network (*class 1*), with changepoints that render parent configurations invariant removed for the respective nodes. While this imbues the model with aspects of a *class 2* approach, it suffers from the fact that changepoints are inextricably associated with changes in the presence/absence status of interactions, rather than changes in the interaction strengths, resulting in a loss of model flexibility.

$\mathbf{V}_n$ without any conditioning on $\mathcal{K}_n$, $P(\mathbf{V}_n|\mathcal{K}_n) \to P(\mathbf{V}_n)$; hence the terms $\mathbf{K}$ and $\mathcal{K}_n$ in Equations (1–4) become obsolete. For the remainder of this section, we use the generic notation $\tilde{\mathbf{V}} = (\tilde{\mathbf{V}}_1, \ldots, \tilde{\mathbf{V}}_N)$ to denote the latent variables induced by the changepoint prior. Depending on the form of the latter, we either have $\tilde{\mathbf{V}} = (\mathbf{V}, \mathbf{K})$ or $\tilde{\mathbf{V}} = \mathbf{V}$. Given a Bayesian mixture model for which the latent variables are of the form of one of the two changepoint processes discussed above, and the parameters can be integrated out in the likelihood, as in Equation (2), Fearnhead (2006) shows that the changepoints can be sampled from the proper posterior distribution *exactly*, with a dynamic programming scheme. The computational complexity is quadratic in the number of observations m. To adapt this scheme to the inference of inhomogeneous DBNs, note from Equation (4) that the Bayesian sampling of $P(\mathcal{G}, \tilde{\mathbf{V}}|\mathcal{D})$ can in principle follow a Gibbs sampling procedure, iteratively sampling the latent variables from $P(\tilde{\mathbf{V}}|\mathcal{G}, \mathcal{D})$, and a new network structure from $P(\mathcal{G}|\tilde{\mathbf{V}}, \mathcal{D})$. The first step can be accomplished with Fearnhead's dynamic programming scheme (Fearnhead, 2006). However, given the comparatively high computational costs, the overall scheme is computationally inefficient if we follow Lèbre (2007), Robinson and Hartemink (2009), Grzegorczyk and Husmeier (2009) and Lèbre *et al.* (2010) and stick to a structure MCMC step for updating $\mathcal{G}$, i.e. if we follow a computationally expensive complete Gibbs step for sampling from $P(\tilde{\mathbf{V}}|\mathcal{G}, \mathcal{D})$ by a computationally cheap MH within Gibbs step for incomplete sampling from $P(\mathcal{G}|\tilde{\mathbf{V}}, \mathcal{D})$. To resolve this issue, we adapt the sampling scheme proposed in Friedman and Koller (2003), Equation (10). Recall that the network structure $\mathcal{G}$ is defined by the complete set of parent sets $\{\pi_n\}_{1 \le n \le N}$. Having sampled $\tilde{\mathbf{V}} = (\tilde{\mathbf{V}}_1, \ldots, \tilde{\mathbf{V}}_N)$ from $P(\tilde{\mathbf{V}}|\mathcal{G}, \mathcal{D})$ in the previous Gibbs step, we now sample $\mathcal{G}$ from $P(\mathcal{G}|\tilde{\mathbf{V}}, \mathcal{D})$ by sampling, for all nodes X_n, $n = 1, \ldots, N$, new parent configurations $\{\pi_n\}$ from

$$P(\pi_n|\mathcal{D}, \tilde{\mathbf{V}}_n) = \Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\tilde{\mathbf{V}}_n]) / \sum_{\tilde{\pi}_n} \Psi^{\dagger}(\mathcal{D}_n^{\tilde{\pi}_n}[\tilde{\mathbf{V}}_n]) \tag{6}$$

where $\Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\tilde{\mathbf{V}}_n])$ has been defined in Equation (3). Equation (6) entails a complete enumeration over all parent configurations, which is computationally expensive. In Grzegorczyk and Husmeier (2009) it was found that this sampling scheme is computationally inefficient when applied to inhomogeneous DBNs. We now demonstrate that this scheme is only inefficient when combined with the RJMCMC scheme for sampling $\tilde{\mathbf{V}}$, but that in combination with the dynamic programming scheme for exact sampling of $\tilde{\mathbf{V}}$ from $P(\tilde{\mathbf{V}}|\mathcal{G}, \mathcal{D})$, an overall gain in computational efficiency can be achieved. We empirically corroborate this conjecture in Section 5.1.[2] For the specific *class 2* model employed in this study (Grzegorczyk and Husmeier, 2009) we provide the technical details of the traditional RJMCMC and the novel Gibbs sampling procedures in the Supplementary Material.

2.3 Information coupling between nodes based on Bayesian clustering

We instantiate the model from Equation (4) by following Fearnhead (2006) and employing the point process prior for the changepoint locations defined in Equation (5), i.e. the terms $\mathbf{K}$ and $\mathcal{K}_n$ in Equations (1–4) become obsolete. We extend the model by introducing a cluster function $\mathcal{C}(.)$ that allocates the nodes $X_1, \ldots, X_n$ to c $(1 \le c \le N)$ non-empty clusters, each characterized by its own changepoint vector $\mathbf{V}_i^{\mathcal{C}}$, $1 \le i \le c$:

$$\begin{aligned} P(\mathcal{G}, \mathbf{V}^{\mathcal{C}}, \mathcal{D}, \mathcal{C}) &= P(\mathcal{C}) P(\mathbf{V}^{\mathcal{C}}|\mathcal{C}) P(\mathcal{G}) P(\mathcal{D}|\mathcal{G}, \mathbf{V}^{\mathcal{C}}, \mathcal{C}) \\ &= P(\mathcal{C}) \left(\prod_{i=1}^{c} P(\mathbf{V}_i^{\mathcal{C}}|\mathcal{C}) \right) \prod_{n=1}^{N} P(\pi_n) \Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathbf{V}_{\mathcal{C}(n)}^{\mathcal{C}}]) \end{aligned} \tag{7}$$

with $\mathbf{V}^{\mathcal{C}} = (\mathbf{V}_1^{\mathcal{C}}, \ldots, \mathbf{V}_c^{\mathcal{C}})$, where c is the number of non-empty node clusters induced by $\mathcal{C}$. We assume for P($\mathcal{C}$) a uniform distribution on all functions $\mathcal{C}$ that give c $(1 \le c \le N)$ clusters. The key idea behind the model of Equation (7) is to encourage information sharing among nodes with respect to changepoint locations. Moreover, nodes that are in the same cluster i $(1 \le i \le c)$ share the same allocation vector $\mathbf{V}_i^{\mathcal{C}}$ and will be 'penalized' only once.[3] Note that the novel model is a generalization that subsumes both *class 1* and *class 2* models as limiting cases. It corresponds to *class 1* for $c = 1$ and to *class 2* for $c = N$. Inference can follow a slightly extended Gibbs sampling procedure, where we iteratively sample the latent variables from $P(\mathbf{V}_i^{\mathcal{C}}|\mathcal{G}, \mathcal{D}, \mathcal{C})$, a new network structure from $P(\mathcal{G}|\mathbf{V}_i^{\mathcal{C}}, \mathcal{D}, \mathcal{C})$, and a new cluster formation from $P(\mathcal{C}|\mathbf{V}_i^{\mathcal{C}}, \mathcal{D}, \mathcal{G})$. The first two steps follow the procedure discussed in Section 2.2. For the third step, sampling from $P(\mathcal{C}|\mathbf{V}_i^{\mathcal{C}}, \mathcal{D}, \mathcal{G})$, we adopt an RJMCMC scheme (Green, 1995) based on cluster birth (b), death (d) and re-clustering (r) moves.[4] In a cluster birth move we randomly select a node cluster i that contains at least 2 nodes, and we randomly choose a node contained in it. The move tries to re-cluster this node from the i-th cluster to a new cluster $c+1$. Denote by $\mathcal{C}^{\star}$ the new cluster formation thus obtained. For the i-th cluster and for the new $(c+1)$-th cluster we propose new changepoint allocation vectors $\mathbf{V}_i^{\mathcal{C}^{\star}}$ and $\mathbf{V}_{c+1}^{\mathcal{C}^{\star}}$ by sampling them from the distributions $P(\mathbf{V}_i^{\mathcal{C}^{\star}}|\mathcal{G}, \mathcal{D}, \mathcal{C}^{\star})$ and $P(\mathbf{V}_{c+1}^{\mathcal{C}^{\star}}|\mathcal{G}, \mathcal{D}, \mathcal{C}^{\star})$, defined in Equation (9), with the dynamic programming (DP) scheme proposed in Fearnhead (2006), as discussed in Section 2.2. In a cluster death move we randomly select one of the clusters that contain only a single node, and we re-allocate this node to one of the other existing clusters, chosen randomly. The first cluster disappears and for cluster j, which absorbs the node, we propose a new changepoint allocation vector $\mathbf{V}_j^{\mathcal{C}^{\star}}$ from $P(\mathbf{V}_j^{\mathcal{C}^{\star}}|\mathcal{G}, \mathcal{D}, \mathcal{C}^{\star})$ with DP, where $\mathcal{C}^{\star}$ denotes the proposed cluster formation. In a re-clustering move we randomly choose two clusters i and j $(i \ne j)$ as follows. First, cluster i is randomly selected among those that contain at least 2 nodes. Next, cluster j is randomly selected among the remaining clusters. We then randomly chose one of the nodes from cluster i and re-allocate the selected node to cluster j. Denote by $\mathcal{C}^{\star}$ the new cluster formation obtained. (Since cluster i contains at least 2 nodes, this does not affect c.) For both clusters i and j we propose new changepoint allocation vectors $\mathbf{V}_i^{\mathcal{C}^{\star}}$ and $\mathbf{V}_j^{\mathcal{C}^{\star}}$ from $P(\mathbf{V}_i^{\mathcal{C}^{\star}}|\mathcal{G}, \mathcal{D}, \mathcal{C}^{\star})$ and $P(\mathbf{V}_j^{\mathcal{C}^{\star}}|\mathcal{G}, \mathcal{D}, \mathcal{C}^{\star})$ with DP.

The acceptance probabilities of these three RJMCMC moves are given by the product of the likelihood ratio (*LR*), the prior ratio (*PR*), the inverse proposal probability ratio or Hastings factor (*HR*) and the Jacobian (J) in the standard way (Green, 1995): $A_{(b,d,r)} = min\{1, R_{(b,d,r)}\}$, where $R_{(b,d,r)} = LR \times PR \times HR \times J$. Since this is a discrete problem, the Jacobian is $J = 1$, and for the chosen uniform prior on $\mathcal{C}$, the prior ratio is $PR = 1$. For a cluster birth move (b), symbolically $(\mathcal{C}, \mathbf{V}^{\mathcal{C}}) \to (\mathcal{C}^{\star}, \mathbf{V}^{\mathcal{C}^{\star}})$, we thus get: $R_{(b)} = LR \times HR$

$$R_{(b)} = \frac{P(\mathcal{G}, \mathbf{V}^{\mathcal{C}^{\star}}, \mathcal{C}^{\star}, \mathcal{D})}{P(\mathcal{G}, \mathbf{V}^{\mathcal{C}}, \mathcal{C}, \mathcal{D})} \times \frac{c^{\dagger} c^{\ddagger} P(\mathbf{V}_i^{\mathcal{C}}|\mathcal{G}, \mathcal{D}, \mathcal{C})}{c^{\star} P(\mathbf{V}_{c+1}^{\mathcal{C}^{\star}}|\mathcal{G}, \mathcal{D}, \mathcal{C}^{\star}) P(\mathbf{V}_i^{\mathcal{C}^{\star}}|\mathcal{G}, \mathcal{D}, \mathcal{C}^{\star})} \tag{8}$$

where $c^{\dagger}$ is the number of clusters induced by $\mathcal{C}$ with at least two nodes, $c^{\ddagger}$ is the number of nodes in the i-th cluster (that was selected), and $c^{\star}$ is the number of clusters induced by $\mathcal{C}^{\star}$ that contain only a single node. In our extended model the DP scheme described in Section 2.2 can be employed to sample the j-th $(1 \le j \le c)$ allocation vector $\mathbf{V}_j^{\mathcal{C}}$, and we have:

$$P(\mathbf{V}_j^{\mathcal{C}}|\mathcal{G}, \mathcal{D}, \mathcal{C}) = \frac{q_j(\mathcal{D}, \mathcal{C}, \mathcal{G}, \mathbf{V}_j^{\mathcal{C}})}{\sum_{\mathbf{V}_j^{\mathcal{C}^{\star}}} q_j(\mathcal{D}, \mathcal{C}^{\star}, \mathcal{G}, \mathbf{V}_j^{\mathcal{C}^{\star}})} \tag{9}$$

where

$$q_j(\mathcal{D}, \mathcal{C}, \mathcal{G}, \mathbf{V}_j^{\mathcal{C}}) = P(\mathbf{V}_j^{\mathcal{C}}|\mathcal{C}) \prod_{n:\mathcal{C}(n)=j} \Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathbf{V}_j^{\mathcal{C}}]) \tag{10}$$

and the sum in Equation (9) is over all valid allocation vectors $\mathbf{V}_j^{\mathcal{C}^{\star}}$ for the variables in the j-th cluster of $\mathcal{C}^{\star}$.

It follows from Equations (7–8) that all factors except for the $(c+1)$-th in the nominator and the i-th ones cancel out in the likelihood ratio:

$$LR = \frac{q_i(\mathcal{D}, \mathcal{C}^{\star}, \mathcal{G}, \mathbf{V}_i^{\mathcal{C}^{\star}}) \cdot q_{c+1}(\mathcal{D}, \mathcal{C}^{\star}, \mathcal{G}, \mathbf{V}_{c+1}^{\mathcal{C}^{\star}})}{q_i(\mathcal{D}, \mathcal{C}, \mathcal{G}, \mathbf{V}_i^{\mathcal{C}})} \tag{11}$$

[2] For the study in Section 5.1, we used the commonly applied fan-in restriction of 3. When relaxing the fan-in restriction, the computational costs related to Equation (6) increase. However, a set of effective heuristic techniques for approximate computation at controlled computational complexity are available, as discussed in Friedman and Koller (2003).

[3] Rather than 'penalizing' nodes with identical allocation vectors independently, like the model in Grzegorczyk and Husmeier (2009).

[4] Each RJMCMC step was repeated 5 times.

Hence, $R_{(b)} = LR \times HR$ in Equation (8) reduces to:

$$R_{(b)} = \frac{c^{\dagger} c^{\ddagger}}{c^{\star}} \frac{Q_i(\mathcal{D}, \mathcal{C}^{\star}, \mathcal{G}) Q_{c+1}(\mathcal{D}, \mathcal{C}^{\star}, \mathcal{G})}{Q_i(\mathcal{D}, \mathcal{C}, \mathcal{G})} \quad (12)$$

where the terms $Q_j(\mathcal{D}, \mathcal{C}, \mathcal{G}) = \sum_{\mathbf{V}_j^{\mathcal{C}}} q_j(\mathcal{D}, \mathcal{C}, \mathcal{G}, \mathbf{V}_j^{\mathcal{C}})$ can be computed effectively with DP. The acceptance probabilities for death and re-clustering moves can be derived analogously as shown in the Supplementary Material.

3 DATA

3.1 Synthetic RAF-pathway data

The RAF protein signalling transduction pathway, shown in Figure 2, plays a pivotal role in the mammalian immune response and has hence been widely studied in the literature [e.g. Sachs *et al.* (2005)]. For our simulation study we followed Grzegorczyk and Husmeier (2009) and generated synthetic network data from a slightly modified version of the pathway, in which an extra feedback loop has been added to node 'PIP3': $PIP3(t+1) = \sqrt{1-\varepsilon^2} PIP3(t) + \varepsilon \phi_{PIP3}(t+1)$. The realizations of the other nodes are linear combinations of the realizations of their parents at the preceding time points plus iid standard Normally distributed noise injections. Example for 'PIP2': $PIP2(t+1) = \beta_{PIP3}(t) PIP3(t) + \beta_{PLCG}(t) PLCG(t) + c_{PIP2} \phi_{PIP2}(t+1)$, where the variables $\phi_{\cdot}(.)$ are iid standard Gaussian distributed, and the coefficient $c_{\cdot}$ can be used to vary the signal-to-noise ratio (SNR). The regression coefficients are sampled from continuous uniform distributions on the interval $[0.5, 2]$ with a random sign. We focus on the medium autocorrelation strength $\varepsilon = 0.25$, the SNRs 3 and 10, and we generate time series of length $m = 21$. Unlike Grzegorczyk and Husmeier (2009) we do not only focus on *class 2* data, but distinguish four different scenarios: (i) homogeneous DBN data with regression coefficients that are constant in time, e.g. $\beta_{PIP3}(t) = const.$; (ii) inhomogeneous *class 1* DBN data where *all* regression coefficients of the domain are re-sampled after $t = 11$; (iii) inhomogeneous *class 2* DBN data with each node having one or two node-specific changepoints, where the corresponding regression coefficients are re-sampled; (iv) inhomogeneous *regularized class 2* data generated from a DBN where the coefficients of five nodes are re-sampled after $t = 11$, and the coefficients of the other 5 nodes are re-sampled twice independently, after $t = 8$ and after $t = 13$. We also consider scenario (v): inhomogeneous *regularized class 2* data *without* any autocorrelation: $\varepsilon = 1$. For SNR=3 and SNR=10 we generated 10 independent data instantiations for each scenario (i)–(v).

3.2 Gene expression time series from *Arabidopsis thaliana*

The Arabidopsis data stem from a study related to circadian regulation in plants. To this end, *A.thaliana* seedlings grown under four different artificially controlled light/dark cycles were transferred to constant light and harvested at 12–13 time points in 2/4-hour intervals. From these seedlings, RNA was extracted and assayed on Affymetrix GeneChip oligonucleotide arrays. As Grzegorczyk and Husmeier (2009) we focus on $N = 9$ genes, LHY, TOC1, CCA1, ELF4, ELF3, GI, PRR9, PRR5 and PRR3, which from previous studies are known to be involved in circadian regulation (Locke *et al.*, 2005). Details about the data and their pre-processing are available from Grzegorczyk and Husmeier (2009).

3.3 Synthetically generated network in *Saccharomyces cerevisiae* (yeast)

While *systems biology* aims to develop a formal understanding of biological processes via the development of quantitative mathematical models, *synthetic biology* aims to use such models to design unique biological circuits (synthetic networks) in the cell able to perform specific tasks. Conversely, data from synthetic biology can be utilized to assess the performance of models from systems biology. We used a synthetically generated network of five genes in *S.cerevisiae* (yeast), devised in Cantone *et al.* (2009) and depicted in Figure 3, which was obtained from synthetically designed yeast cells grown with different carbon sources: galactose ('switch on') or glucose ('switch off'). We took the data from Cantone *et al.* (2009), which were obtained with quantitative RT-PCR in intervals of 20 min up to 5 h for the first, and in intervals of 10 min up to 3 h for the second condition. In our study, we standardized the data via a log and a z-score transformation.

4 SIMULATION DETAILS

The two improvements proposed in Section 2 can be applied to any of the inhomogeneous DBNs recently proposed in the literature (Grzegorczyk and Husmeier, 2009; Lèbre, 2007; Lèbre *et al.*, 2010; Robinson and Hartemink, 2009). In our empirical simulation study we use the model presented in Grzegorczyk *et al.* (2010) as *class 1* representant. The *class 2* model representant is taken from Grzegorczyk and Husmeier (2009). The novel model can be thought of as a regularized consensus of both models: It is effectively a *class 1* model if it infers only one cluster, and it becomes a *class 2* model if it infers N clusters such that each node has its own node-specific changepoints. In our simulation study we also include a standard homogeneous dynamic Bayesian network model based on the standard BGe score (Geiger and Heckerman, 1994). As in earlier studies (Grzegorczyk and Husmeier, 2009; Grzegorczyk *et al.*, 2010) we employ a uniform graph prior subject to a maximum fan-in of 3, and we chose the prior parameter distributions in Equations (1) and (2) maximally uninformative subject to the regularity conditions in Geiger and Heckerman (1994). We demonstrate in Section 5.1 that inference based on Gibbs sampling/dynamic programming substantially improves convergence and mixing. Thus, in the cross-method comparison (see Section 5.2) the inhomogeneous DBN models have been inferred by Gibbs sampling/dynamic programming rather than employing the less effective RJMCMC sampling schemes. As is standard, we discarded a burn-in phase, and tested for convergence (PSRF ≤ 1.1) based on potential scale reduction factors (PSRF), see Gelman and Rubin (1992), resulting in 200 Gibbs steps per dataset and method.

5 RESULTS

5.1 Convergence diagnostics on gene expression time series from *A.thaliana*

The objective of the first study was to assess the improvement in convergence and mixing achieved with the dynamic programming scheme of Section 2.2. To this end, we applied the inhomogeneous DBN of Equation (1) to gene expression time series from the model plant *A.thaliana*, described in Subsection 3.2. We aimed to reconstruct a regulatory network among 9 genes, which

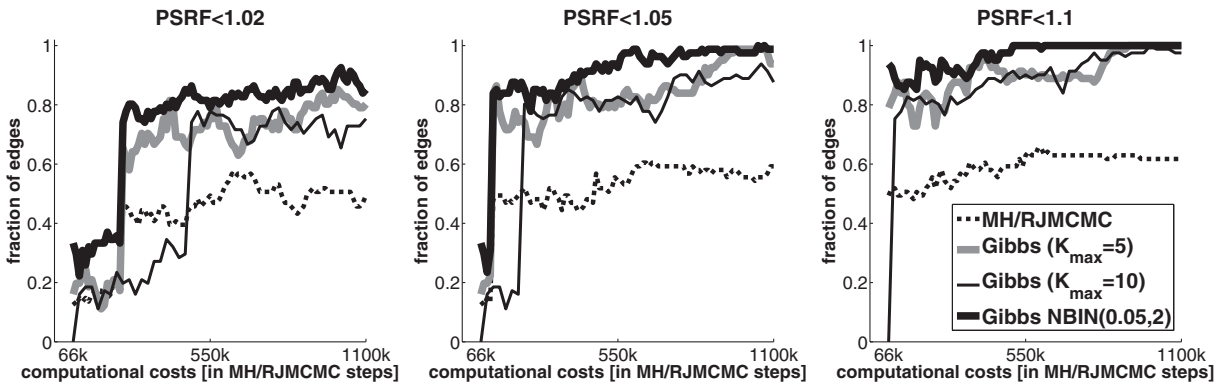

Fig. 1. Convergence diagnostics. The graphs show the proportion of edges for which the PSRF lies below the indicated threshold, satisfying the respective convergence criterion. The horizontal axes represent simulation time, measured in terms of the equivalent number of MH/RJMCMC steps. Four MCMC schemes for the *class 2* model from Grzegorczyk and Husmeier (2009) are compared; see Section 5.1, where the terms in the legend are explained.

from previous studies are known to be involved in circadian regulation (Locke *et al.*, 2005). Our model and simulation setup matched the one described in Grzegorczyk and Husmeier (2009). We compared the standard MCMC scheme applied in previous work, MH/RJMCMC (Grzegorczyk and Husmeier, 2009; Lèbre *et al.*, 2010; Robinson and Hartemink, 2009), which is based on RJMCMC (Green, 1995) and structure MCMC (Madigan and York, 1995), with the Gibbs sampling/dynamic programming scheme discussed in Section 2.2. For the latter, we compared three different subschemes, which differ with respect to the prior distribution on the changepoints. The first subscheme imposes a Poisson prior with truncation threshold $\mathcal{K}_n \leq 10$ on the number of components, $P(\mathcal{K}_n)$, and the same even-numbered order statistics prior as applied in Grzegorczyk and Husmeier (2009) and Green (1995) on the segmentations, $P(\mathbf{V}_n|\mathcal{K}_n)$. The second subscheme is identical, except that the truncation threshold has been lowered to $\mathcal{K}_n \leq 5$. The third subscheme follows Fearnhead (2006) and uses the prior imposed by the point process prior of Equation (5) with hyperparameters $p = 0.05$ and $a = 2$. We refer to these four schemes as MH/RJMCMC, Gibbs($K_{max} = 10$), Gibbs($K_{max} = 5$) and Gibbs-NBIN, respectively. To assess the degree of convergence, we repeated the MCMC simulations from five different initializations and computed the PSRFs for all potential edges, as described in the Supplementary Material. Recall that PSRF = 1 indicates perfect convergence, and PSRF ≤ 1.1 is usually taken as an indication of sufficient convergence. Ideally, we would like to plot the PSRF values against the MCMC iteration number. However, due to different computational costs of the individual steps of the MCMC simulations—a Gibbs step based on dynamic programming is substantially more expensive than an MH/RJMCMC step—we plotted the PSRF scores against the simulation time, measured in terms of conventional MH/RJMCMC steps.[5] The results are shown in Figure 1. The proposed Gibbs sampling scheme based on dynamic programming significantly outperforms the conventional MH/RJMCMC scheme. When comparing the different dynamic programming schemes, Gibbs-NBIN performs slightly better than Gibbs($K_{max} = 10$) and Gibbs($K_{max} = 5$), in agreement with the findings in Fearnhead (2006). For the reconstructed network topology and the inferred changepoint locations, which in the absence of a true gold standard cannot be evaluated properly, we refer to our Supplementary Material.

5.2 Comparative evaluation on simulated data

For our simulation study we employ the synthetically generated RAF-pathway data from Section 3.1 to cross-compare the network reconstruction accuracy of the proposed *regularized class 2* model with three other models: a standard homogeneous Bayesian network model, a *class 1* model with changepoints that are common to all nodes (Grzegorczyk *et al.*, 2010), and a *class 2* model with node-specific changepoints (Grzegorczyk and Husmeier, 2009). In our study we evaluated the network reconstruction accuracy with the area under the precision-recall curve (AUC) (Davis and Goadrich, 2006); see Section 5.3 for more details. This is standard in systems biology, with larger scores indicating a better performance.

Figure 2 summarizes the empirical results of our simulation study. (1) *Homogeneous data*: Except for the highest setting of the hyperparameter p, the three inhomogeneous DBNs never perform worse than the homogeneous model, while on the other hand for inhomogeneous data, the homogeneous model is inappropriate and performs substantially worse. (2) *class 1 data*: The *class 1* model and the proposed *regularized class 2* model perform equally well. Both outperform the *class 2* model, except for high values of p.[6] (3) *class 2 data*: The *class 1* model cannot accommodate the node-specific changepoints and is outperformed by the proposed *regularized class 2* model (the 'NEW' model). Interestingly, the latter also shows more stability than the *class 2* model with respect to a variation of the hyperparameter p, indicating increased robustness as a consequence of the node clustering. (4) *Regularized class 2 data*: The results are comparable to those for the *class 2* data. The *class 1* model is consistently inferior to the *class 2* model, and the *class 2* model is, once again, substantially more susceptible to a variation of p. The mean AUC values are—overall—lower than for the previous case, the *class 2* data. This seems to be a consequence of spurious interactions resulting from chance correlations. Setting

[5] 1100k MH/RJMCMC (5500 Gibbs-NBIN) steps take 45 min with our MATLAB code on a SunFire X4100M2 machine.

[6] Recall that a high value of the hyperparameter p implies a low prior penalty for changepoints.

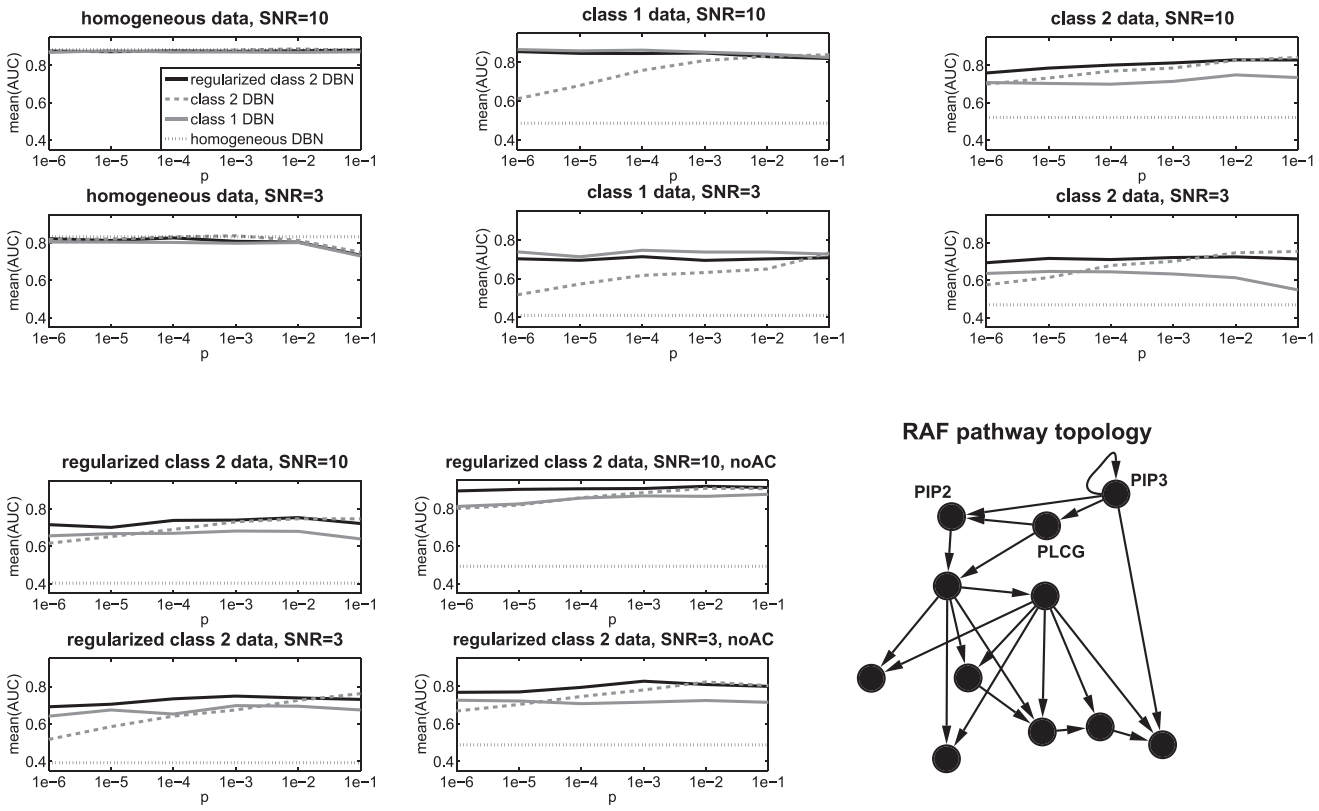

Fig. 2. Network reconstruction accuracy on synthetic data. The figure shows the mean area under the precision-recall curves (AUC) in dependence on the hyperparameter p of the negative binomial point process prior of Equation (5). For the RAF pathway (bottom right panel) we implemented 5 scenarios of inhomogeneity as explained in Section 5.2. For each scenario there is a panel for SNR=3 and SNR=10; 'noAC' stands for 'no autocorrelation'. The following models were applied to the data, each representing a particular class: (i) homogeneous model: the standard DBN model based on the BGe score, (ii) the *class 1* model was taken from Grzegorczyk *et al.* (2010), (iii) the *class 2* model was taken from Grzegorczyk and Husmeier (2009) and (iv) the *regularized class 2* model was generated from the *class 2* model in Grzegorczyk and Husmeier (2009) as explained in Section 3.1. The mean AUC scores were computed from 10 independent data instantiations.

the autocorrelation of node *PIP*3 to zero ($\varepsilon = 1$, no AC), noticeably increases the mean AUC values. In summary, this study shows that the proposed *regularized class 2* model, which implements the method of Section 2.3, is always among the best-scoring models. It shows more robustness than the competing schemes both with respect to a variation of the type of data, and a variation of the prior knowledge (inherent in Equation (5) via p).

5.3 Synthetic biology in *S.cerevisiae*

In the final application we compare the proposed model with other state-of-the art techniques on a topical dataset from synthetic biology. We used a synthetically generated network of five genes in *S.cerevisiae* (yeast), depicted in Figure 3, which was used in Cantone *et al.* (2009) to evaluate two state-of-the-art network reconstruction methods: BANJO, a conventional DBN, trained with simulated annealing; and TSNI, an approach based on ordinary differential equations. Both methods, which are described in more detail in Cantone *et al.* (2009), were applied to gene expression time series obtained from synthetically designed yeast cells grown with different carbon sources: galactose ('switch on') or glucose ('switch off'). BANJO and TSNI were then applied to infer a network [see Cantone *et al.* (2009) for details], from which, by comparison with the known gold standard, the precision (proportion of correctly predicted interactions out of the total number of predicted interactions) and recall (percentage of true interactions that have been correctly identified) scores were determined. In our study, we used the data described in Section 3.3, applied the proposed *regularized class 2* model as described in Sections 2.3, and sampled networks from the posterior distribution with the Gibbs sampling scheme described in Section 2.2. This gives us an ordering of interactions, ranked by their marginal posterior probability, and by plotting precision against recall scores for different thresholds, we obtain the precision-recall (PR) curves (Davis and Goadrich, 2006) shown in Figure 3. Larger areas under the PR curve are indicative of a better reconstruction accuracy; hence in agreement with Cantone *et al.* (2009) we find that the 'switch on' data are more informative than their 'switch off' counterpart. The scores for BANJO and TSNI, which we took from Cantone *et al.* (2009), lie clearly and consistently below the 'switch on' PR curve, for different choices of the changepoint process prior—defined by p in Equation (5). This suggests that the proposed method achieves a genuine and significant improvement over state-of-the-art schemes reported in the recent systems biology literature.

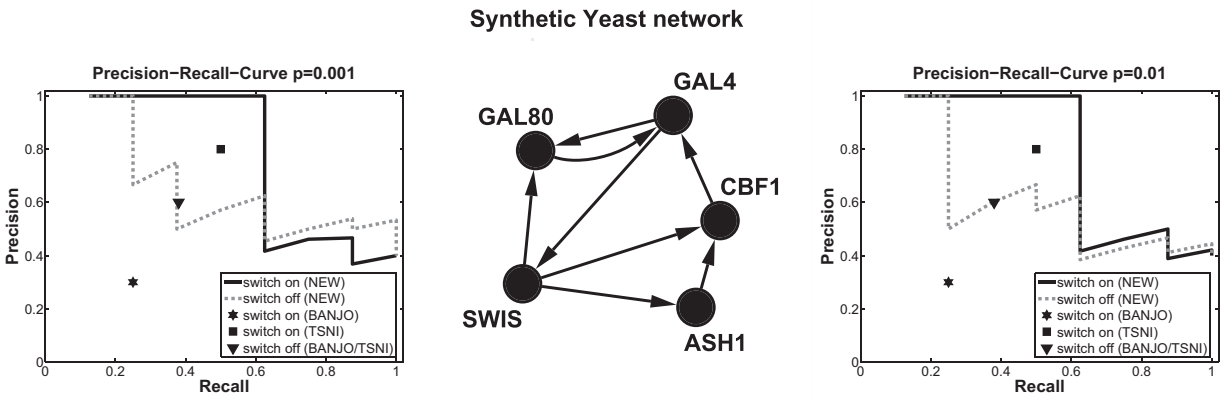

Fig. 3. Network reconstruction accuracy evaluated with synthetic biology. The centre panel shows the true gene regulatory network in *S.cerevisiae*, designed in Cantone *et al.* (2009). The outer panels show the precision-recall curves for the proposed *regularized class 2* model (NEW). Results were obtained for both experimental conditions: the 'switch on' and the 'switch off' time series described in Cantone *et al.* (2009). The symbols at fixed positions (triangle, star and square) mark the precision/recall results obtained in Cantone *et al.* (2009) for two state-of-the-art network reconstruction methods: BANJO (conventional homogeneous DBN) and TSNI (ODE-based approach).

6 CONCLUSION

We have proposed two improvements for time-varying DBNs: a Gibbs sampling (GS) scheme based on dynamic programming (DP) as an alternative to RJMCMC, and information coupling between nodes based on Bayesian clustering. The evaluation on a real gene expression dataset from *A.thaliana* suggests that GS-DP shows faster mixing and convergence than MH/RJMCMC. A comparative evaluation on synthetic data demonstrates that the new model based on information coupling between nodes compares favourably with earlier models that either employ network-wide (*class 1*) or node-specific (*class 2*) changepoints. On gene expression time series from a recent study of synthetic biology in *S.cerevisiae* the proposed model has outperformed two state-of-the-art network reconstruction methods. These findings suggest that the proposed method makes important contributions both to inference and performance of network reconstruction methods, and hence adds a valuable new tool to the kit of computational systems biology. In our future work we will investigate different choices for the prior on node cluster formations, introduced in Section 2.3, exploring methods from Bayesian non-parametrics based on Dirichlet process priors.

ACKNOWLEDGEMENTS

Marco Grzegorczyk is supported by the Graduate School 'Statistische Modellbildung' of the Department of Statistics, TU Dortmund University. Dirk Husmeier is supported by the Scottish Government Rural and Environment Research and Analysis Directorate (RERAD) and under the EU FP7 project 'Timet'.

Conflict of Interest: none declared.

REFERENCES

Butte,A. and Kohane,I. (2000) Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pac. Symp. Biocomput.*, **2000**, 418–429.

Cantone,I. *et al.* (2009) A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. *Cell*, **137**, 172–181.

Cao,J. and Ren,F. (2008) Exponential stability of discrete-time genetic regulatory networks with delays. *IEEE Trans. Neural Netw.*, **19**, 520–523.

Davis,J. and Goadrich,M. (2006) The relationship between precision-recall and ROC curves. *ICML*, **23**, 233–240.

Fearnhead,P. (2006) Exact and efficient Bayesian inference for multiple changepoint problems. *Stat. Comput.*, **16**, 203–213.

Friedman,N. and Koller,D. (2003) Being Bayesian about network structure. *Mach. Learn.*, **50**, 95–126.

Geiger,D. and Heckerman,D. (1994) Learning Gaussian networks. *UAI*, **10**, 235–243.

Gelman,A. and Rubin,D.B. (1992) Inference from iterative simulation using multiple sequences. *Stat. Science*, **7**, 457–472.

Green,P. (1995) Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82**, 711–732.

Grzegorczyk,M. and Husmeier,D. (2009) Non-stationary continuous dynamic Bayesian networks. *NIPS*, **22**, 682–690.

Grzegorczyk,M. *et al.* (2008) Modelling non-stationary gene regulatory processes with a non-homogeneous Bayesian network and the allocation sampler. *Bioinformatics*, **24**, 2071–2078.

Grzegorczyk,M. *et al.* (2010) Modelling non-stationary dynamic gene regulatory processes with the BGM model. *Comput. Stat.* [Epub ahead of print, doi:10.1007/s00180-010-0201-9].

Kolar,M. *et al.* (2009) Sparsistent learning of varying-coefficient models with structural changes. *NIPS*, **22**, 1006–1014.

Lèbre,S. (2007) *Stochastic process analysis for Genomics and Dynamic Bayesian Networks inference.* Ph.D. Thesis, Université d'Evry-Val-d'Essonne, France.

Lèbre,S. *et al.* (2010) Statistical inference of the time-varying structure of gene-regulation networks. *BMC Syst. Biol.*, **4**, Article ID or number: 130.

Locke,J. *et al.* (2005) Extension of a genetic network model by iterative experimentation and mathematical analysis. *Mol. Syst. Biol.*, **1**, Article ID or number: 2005.0013.

Madigan,D. and York,J. (1995) Bayesian graphical models for discrete data. *Int. Stat. Rev.*, **63**, 215–232.

Nowlan,S.J. and Hinton,G.E. (1992) Simplifying neural networks by soft weight-sharing. *Neural Comput.*, **4**, 473–493.

Robinson,J.W. and Hartemink,A.J. (2009) Non-stationary dynamic Bayesian networks. *NIPS*, **21**, 1369–1376.

Sachs,K. *et al.* (2005) Protein-signaling networks derived from multiparameter single-cell data. *Science*, **308**, 523–529.

Talih,M. and Hengartner,N. (2005) Structural learning with time-varying components: Tracking the cross-section of financial time series. *J. R. Stat. Soc. B*, **67**, 321–341.

Wang,Y. *et al.* (2010) Global robust power-rate stability of delayed genetic regulatory networks with noise perturbation. *Cogn. Neurodyn.*, **4**, 81–90.

Wilkinson,D. (2006) *Stochastic modelling for systems biology*. Chapman and Hall/CRC Press, Boca Raton, Florida.

Xiao,M. and Cao,J. (2008) Genetic oscillation deduced from Hopf bifurcation in a genetic regulatory network with delays. *Math. Biosci.*, **215**, 55–63.

Xuan,X. and Murphy,K. (2007) Modeling changing dependency structure in multivariate time series. *ICML*, **24**, 1055–1062.