OXFORD JOURNALS
Bioinformatics

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

Journal:	*Bioinformatics*
Manuscript ID:	Draft
Category:	Original Paper
Date Submitted by the Author:	n/a
Complete List of Authors:	Grzegorczyk, Marco; TU Dortmund University, Statistics Husmeier, Dirk; Biomathematics and Statistics Scotland (BioSS)
Keywords:	Bayesian networks, Gene regulatory networks, Machine learning, Systems biology

BIOINFORMATICS
Vol. 00 no. 00 2010
Pages 1–7

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

Marco Grzegorczyk[a] and Dirk Husmeier[b]

[a] Department of Statistics, TU Dortmund University, Dortmund, Germany
[b] Biomathematics and Statistics Scotland (BioSS), Edinburgh, United Kingdom

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 paper, 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: From http://www.statistik.tu-dortmund.de/bio2010.html a supplementary paper with algorithmic details and further results for the Arabidopsis data can be downloaded. The MATLAB programs used for our simulations are available upon request.

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

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. Facing the extremes of mechanistic models, which are restricted to small systems with only few components, and mutual information based approaches, which are too simplistic, 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); 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 Ahmed and Xing (2009) and 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 paper 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 paper, 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 data set $\mathcal{D}$, where $\mathcal{D}_{n,t}$ and $\mathcal{D}_{(\pi_n,t)}$ are the tth 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 $P\Big(X_n(t) = \mathcal{D}_{n,t}|\pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)}, \boldsymbol{\theta}_n\Big)$. Common choices are a multinomial distribution, as used in Robinson and Hartemink (2009), or a linear Gaussian distribution, as for example applied 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 Lèbre (2007); Robinson and Hartemink (2009); Grzegorczyk and Husmeier (2009); Lèbre *et al.* (2010)) and is given by

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

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

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 kth 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); Lèbre *et al.* (2010), the following integral can be solved analytically:

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

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

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

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

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

In Grzegorczyk and Husmeier (2009); 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 (Robinson and Hartemink, 2009; Grzegorczyk and Husmeier, 2009; Lèbre *et al.*, 2010; Grzegorczyk *et al.*, 2010) follow the same procedure: to sample the network structure $\mathcal{G}$, they follow Madigan and York (1995) and apply Metropolis-Hastings (MH) structure MCMC, based on single-edge operations; to sample the latent variables $(\mathbf{V}, \mathbf{K})$, they follow Green (1995) and apply reversible jump Markov chain Monte Carlo (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 Eq. (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

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

whose form is defined by two hyperparameters, a and p. The choice of this prior immediately imposes a prior distribution on the latent variables $\mathbf{V}_n$ without any conditioning on $\mathcal{K}_n$, $P(\mathbf{V}_n|\mathcal{K}_n) \rightarrow P(\mathbf{V}_n)$; hence the terms $\mathbf{K}$ and $\mathcal{K}_n$ in Eqn. (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 Eq. (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

[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.

from Eq. (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); 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), Eq. (10). Recall that the network structure $\mathcal{G}$ is defined by the complete set of parent sets $\{\pi_n\}_{1\leq n\leq N}$. 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

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

where $\Psi^\dagger(\mathcal{D}_n^{\pi_n}[\tilde{\mathbf{V}}_n])$ has been defined in Eq. (3). Eq. (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 Eq. (4) by following Fearnhead (2006) and employing the point process prior for the changepoint locations defined in Eq. (5), i.e. the terms $\mathbf{K}$ and $\mathcal{K}_n$ in Eqn. (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 $(1 \leq c \leq N)$ non-empty clusters, each characterized by its own changepoint vector $\mathbf{V}_i^{\mathcal{C}}$, $1 \leq i \leq c$:

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

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 $(1 \leq c \leq N)$ clusters. The key idea behind the model of Eq. (7) is to encourage information sharing among nodes with respect to changepoint locations. Moreover, nodes that are in the same cluster i $(1 \leq i \leq c)$ 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 ith cluster to a new cluster $c+1$. Denote by $\mathcal{C}^\star$ the new cluster formation thus obtained. For the ith 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 Eq. (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 $(i \neq j)$ 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 $(\mathcal{C},\mathbf{V}^{\mathcal{C}}) \rightarrow (\mathcal{C}^\star,\mathbf{V}^{\mathcal{C}^\star})$, we thus get: $R_{(b)} = LR \times HR$

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

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 ith 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 $(1 \leq j \leq c)$ allocation vector $\mathbf{V}_j^{\mathcal{C}}$, and we have:

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

where

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

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

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

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

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

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

[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 Eq. (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.

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 paper.

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. E.g. for 'PIP2': $PIP2(t+1) = \beta_{PIP3}(t)PIP3(t) + \beta_{PLCG}(t)PLCG(t) + c_{PIP2}\phi_{PIP2}(t+1)$, where the variables $\phi_{.}(.)$ are iid standard Gaussian distributed, and the coefficient $c_{.}$ 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 $m = 21$. Different from Grzegorczyk and Husmeier (2009) we do not 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, *Arabidopsis 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 preprocessing 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 utilised to assess the performance of models from systems biology. We used a synthetically generated network of five genes in *Saccharomyces 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 minutes up to 5 hours for the first, and in intervals of 10 minutes up to 3 hours 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 (Lèbre, 2007; Robinson and Hartemink, 2009; Grzegorczyk and Husmeier, 2009; Lèbre *et al.*, 2010). 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 Eqn. (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$\leq$ 1.1) based on potential scale reduction factors (PSRF), see (Gelman and Rubin, 1992), resulting in 200 Gibbs steps per data set and method.

5 RESULTS

5.1 Convergence diagnostics on gene expression time series from *Arabidopsis 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 Eq. (1) to gene expression time series from the model plant *Arabidopsis thaliana*, described in Subsection 3.2. We aimed to reconstruct a regulatory network among 9 genes, which 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 (Robinson and Hartemink, 2009; Grzegorczyk and

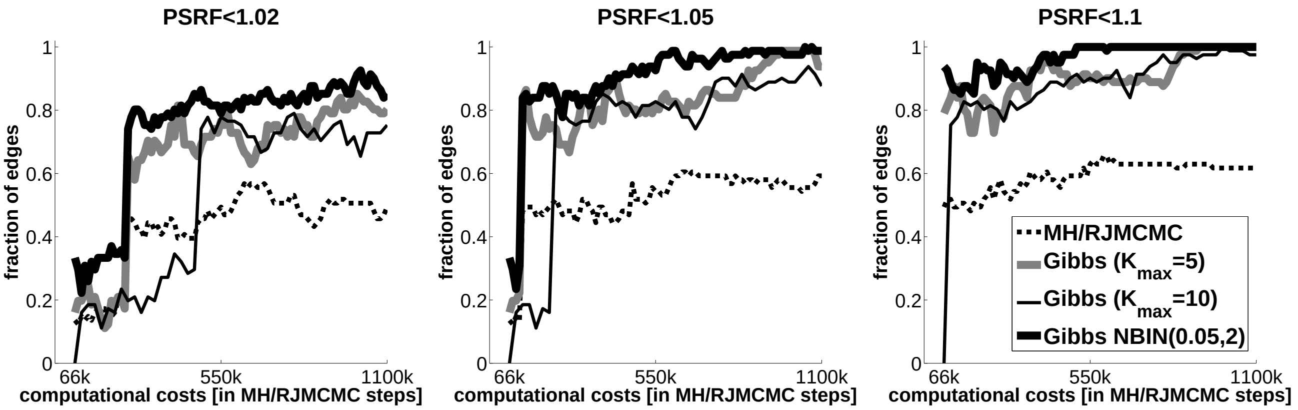


Fig. 1. Convergence diagnostics. The graphs show the proportion of edges for which the PSRF lies below the indicated threshold, satisfying the respective convergence criterion. The horizontal axes represent simulation time, measured in terms of the equivalent number of MH/RJMCMC steps. Four MCMC schemes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

Husmeier, 2009; Lèbre *et al.*, 2010), 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); 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 Eq. (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 paper. 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 paper.

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 the autocorrelation of node $PIP3$ to zero ($\varepsilon = 1$, no AC), noticeably increases the mean AUC values. In summary, this study shows that the proposed *regularized class 2*

[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.

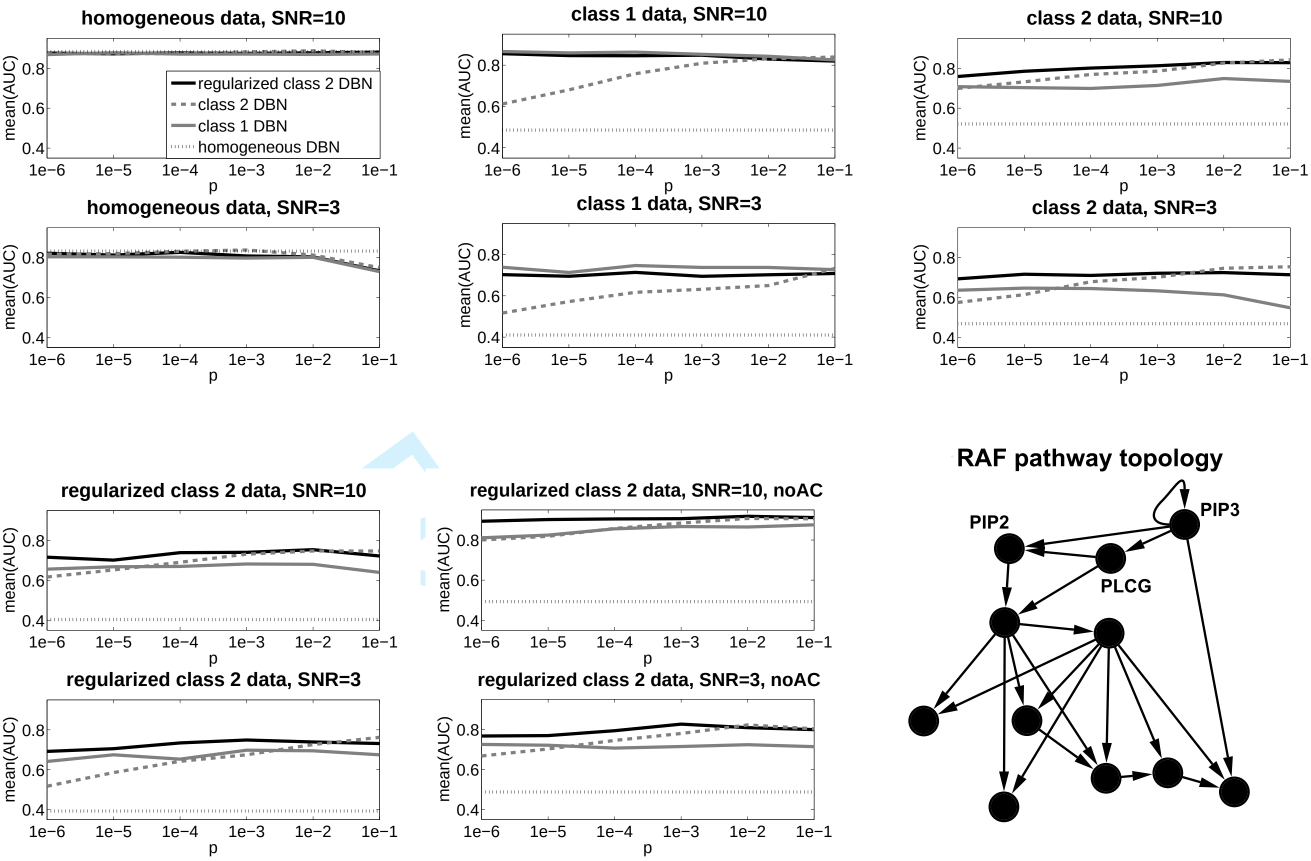


Fig. 2. Network reconstruction accuracy on synthetic data. The figure shows the mean area under the precision-recall curves (AUC) in dependence on the hyperparameter p of the negative binomial point process prior of Eq. (5). For the RAF pathway (bottom right panel) we implemented 5 scenarios of inhomogeneity 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.

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 Eq. (5) via p).

5.3 Synthetic biology in *Saccharomyces cerevisiae*

In the final application we compare the proposed model with other state-of-the art techniques on a topical data set from synthetic biology. We used a synthetically generated network of five genes in *Saccharomyces 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

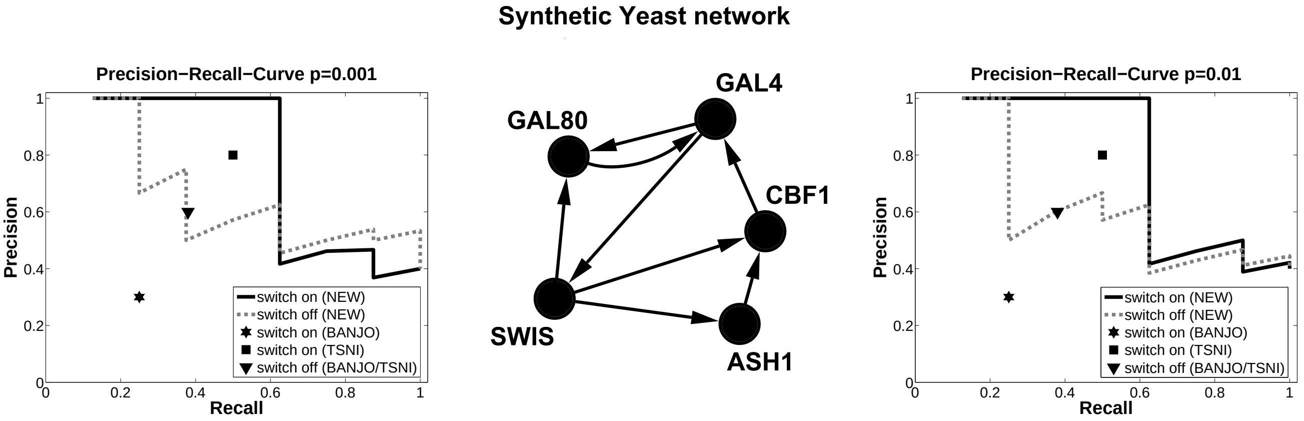


Fig. 3. Network reconstruction accuracy evaluated with synthetic biology. The centre panel shows the true gene regulatory network in *Saccharomyces cerevisiae*, designed in Cantone *et al.* (2009). The outer panels show the precision-recall curves for the proposed *regularized class 2* model (NEW). Results 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).

– defined by p in Eq. (5). This suggests that the method proposed in the present paper achieves a genuine and significant improvement over state-of-the-art schemes reported in the recent systems biology literature.

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 data set from *Arabidopsis 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 *Saccharomyces 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.

REFERENCES

Ahmed, A. and Xing, E. P. (2009) Recovering time-varying networks of dependencies in social and biological studies. *Proceedings of the National Academy of Sciences*, **106**, 11878–11883.

Cantone, I., Marucci, L., Iorio, F., Ricci, M. A., Belcastro, V., Bansal, M., Santini, S., di Bernardo, M., di Bernardo, D. and Cosma1, M. P. (2009) A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. *Cell*, **137**, 172–181.

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. *Statistics and Computing*, **16**, 203–213.

Friedman, N. and Koller, D. (2003) Being Bayesian about network structure. *Machine Learning*, **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. *Statistical 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., Husmeier, D., Edwards, K., Ghazal, P. and Millar, A. (2008) Modelling non-stationary gene regulatory processes with a non-homogeneous Bayesian network and the allocation sampler. *Bioinformatics*, **24**, 2071–2078.

Grzegorczyk, M., Rahnenführer, J. and Husmeier, D. (2010) Modelling non-stationary dynamic gene regulatory processes with the BGM model. *Computational Statistics*. In Press, DOI 10.1007/s00180-010-0201-9.

Kolar, M., Song, L. and Xing, E. (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., Becq, J., Devaux, F., Stumpf, M. P. and Lelandais, G. (2010) Statistical inference of the time-varying structure of gene-regulation networks. *BMC Systems Biology, in print*.

Locke, J., Southern, M., Kozma-Bognar, L., Hibberd, V., Brown, P., Turner, M. and Millar, A. (2005) Extension of a genetic network model by iterative experimentation and mathematical analysis. *Molecular Systems Biology*, **1**, (online).

Madigan, D. and York, J. (1995) Bayesian graphical models for discrete data. *International Statistical Review*, **63**, 215–232.

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

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

Sachs, K., Perez, O., Pe’er, D., Lauffenburger, D. A. and Nolan, G. P. (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. *Journal of the Royal Statistical Society B*, **67**, 321–341.

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

Supplementary material for our paper: 'Improvements in the reconstruction of time-varying gene regulatory networks: dynamic programming and regularization by information sharing among genes' (submitted to *Bioinformatis*, 2010)

Marco Grzegorczyk *(grzegorczyk@statistik.tu-dortmund.de)*
Department of Statistics, TU Dortmund University
44221 Dortmund, Germany

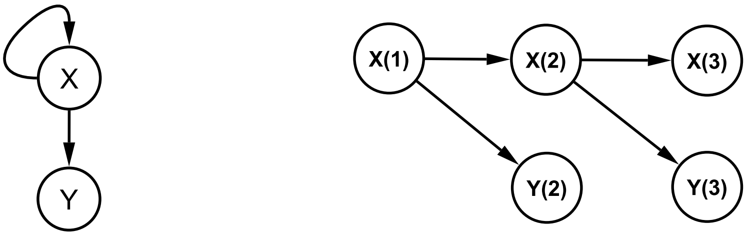
Dirk Husmeier *(dirk@bioss.sari.ac.uk)*
Biomathematics and Statistics Scotland (BioSS)
JCMB, The King's Buildings, Edinburgh EH9 3JZ, United Kingdom

Abstract: *This paper is a supplement to our main paper 'Improvements in the reconstruction of time-varying gene regulatory networks: dynamic programming and regularization by information sharing among genes' (submitted to* Bioinformatis*, 2010). It contains the algorithmic details of the proposed model improvements and provides further results obtained for the circadian genes in Arabidopsis thaliana data. Section 1 gives a comprehensive description of the dynamic programming schemes, the cpBGe model, and the novel regularized cpBGe model. In Section 2 the inferred network topology and the inferred changepoint locations for circadian genes in* Arabidopsis thaliana *are presented and discussed.*

1 Methodology

1.1 The homogeneous dynamic BGe network

DBNs are flexible models for representing probabilistic relationships between interacting variables (nodes) $X_1, \dots, X_N$ via a directed graph $\mathcal{G}$. In most applications first-order DBNs are considered so that all interactions are subject to a time delay $\tau = 1$. An edge pointing from X_j to X_n, symbolically $\mathcal{G}(j,n) = 1$, in a DBN with $\tau = 1$ indicates that the realization of X_n at time point t, symbolically: $X_n(t)$, is conditionally dependent on the realization of X_j at time point $t-1$, symbolically: $X_j(t-1)$. See Figure 1 for an example of a DBN consisting of two nodes X and Y. 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 $\mathcal{G}$. Note that there is a one-to-one mapping between the graph $\mathcal{G}$ and the



(a) recurrent network (b) unfolded dynamic network

Figure 1: **State space graph and corresponding dynamic Bayesian network of order $\tau = 1$.** Panel (a) shows a recurrent state space graph containing two nodes. Node X has a recurrent feedback loop and acts as a regulator of node Y. Panel (b) shows the same graph unfolded in time.

N parent node sets π_n; i.e. $\mathcal{G}(j,n) = 1$ if and only if $X_j \in \pi_n$; and vice-versa $\mathcal{G}(j,n) = 0$ if and only if $X_j \notin \pi_n$. Given a data set $\mathcal{D}$, where $\mathcal{D}_{n,t}$ and $\mathcal{D}_{(\pi_n,t)}$ are the tth realizations $X_n(t)$ and $\pi_n(t)$ of X_n and π_n, respectively, and $1 \leq t \leq m$ represents time, DBNs are based on the following homogeneous Markov chain expansion:

$$P(\mathcal{D}|\mathcal{G},\boldsymbol{\theta}) = \prod_{n=1}^{N}\prod_{t=2}^{m} P\Big(X_n(t) = \mathcal{D}_{n,t}|\pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)},\boldsymbol{\theta}_n\Big) \tag{1}$$

where $\boldsymbol{\theta}$ is the total parameter vector, composed of node-specific subvectors $\boldsymbol{\theta}_n$, which specify the local conditional distributions in the factorization. From Eq. (1) and under the assumption of parameter independence, $P(\boldsymbol{\theta}|\mathcal{G}) = \prod_n P(\boldsymbol{\theta}_n|\pi_n)$, the marginal likelihood is given by

$$\begin{aligned} P(\mathcal{D}|\mathcal{G}) &= \int P(\mathcal{D}|\mathcal{G},\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathcal{G})d\boldsymbol{\theta} = \prod_{n=1}^{N}\Psi(\mathcal{D}_n^{\pi_n}) && (2)\\ \Psi(\mathcal{D}_n^{\pi_n}) &= \int \prod_{t=2}^{m} P\Big(X_n(t) = \mathcal{D}_{n,t}|\pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)},\boldsymbol{\theta}_n\Big)P(\boldsymbol{\theta}_n|\pi_n)d\boldsymbol{\theta}_n && (3)\end{aligned}$$

where $\mathcal{D}_n^{\pi_n} := \{(\mathcal{D}_{n,t},\mathcal{D}_{\pi_n,t-1}) : 2 \leq t \leq m\}$ is the subset of data pertaining to node X_n and parent set π_n. We will refer to $\Psi(\mathcal{D}_n^{\pi_n})$ as *local score* of X_n. For the local scores $\Psi(\mathcal{D}_n^{\pi_n})$ various modelling frameworks, such as sparse Bayesian regression models (e.g. see Rogers and Girolami (2005)), have been proposed and applied in the literature. In this study we focus on the BGe model, which was proposed by Geiger and Heckerman (1994). That is, a linear Gaussian distribution is chosen for the local conditional distribution $P(X_n|\pi_n,\boldsymbol{\theta}_n)$ in Eq.(3), and the conjugate normal-Wishart distribution is assigned to the local prior distributions $P(\boldsymbol{\theta}_n|\pi_n)$. Under fairly weak regularity conditions discussed in Geiger and Heckerman (1994) (parameter modularity), the integral in Eq. (3) has a closed form solution, given by Eq. (24) in Geiger and Heckerman (1994). The resulting expression is called the (local) BGe score.

1.2 The inhomogeneous dynamic changepoint BGe model (cpBGe)

To obtain a inhomogeneous DBN, we generalize Eq. (1) with a node-specific mixture model:

$$P(\mathcal{D}|\mathcal{G},\mathbf{V},\mathbf{K},\boldsymbol{\theta}) = \prod_{n=1}^{N}\prod_{t=2}^{m}\prod_{k=1}^{\mathcal{K}_n} P\Big(X_n(t) = \mathcal{D}_{n,t}|\pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)},\boldsymbol{\theta}_n^k\Big)^{\delta_{\mathbf{V}_n(t),k}} \tag{4}$$

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 kth component of a mixture with $\mathcal{K}_n$ components, and $\mathbf{K} = (\mathcal{K}_1, \dots, \mathcal{K}_n)$. Note that the matrix $\mathbf{V}$ divides the data into several disjoined subsets, each of which can be regarded as pertaining to a separate BGe model with parameters $\boldsymbol{\theta}_n^k$. The vectors $\mathbf{V}_n$ are node-specific, i.e. different nodes can have different changepoints so that the proposed model has a higher flexibility in modelling nonlinear relationships than the BGM model proposed in Grzegorczyk *et al.* (2008). The probability model defined in Eq. (4) is effectively a mixture model with local probability distributions $P(X_n|\pi_n, \boldsymbol{\theta}_n^k)$ and it can hence, under a free allocation of the latent variables, approximate any probability distribution arbitrarily closely. But different from the free allocation of latent variables in Grzegorczyk *et al.* (2008), in the present work, we change the assignment of data points to mixture components from a free allocation to a changepoint process. This allocation scheme provides the approximation of a nonlinear regulation process by a piecewise linear process under the assumption that the temporal processes are sufficiently smooth. Employing a changepoint process effectively reduces the complexity of the latent variable space and incorporates our prior belief that, in a time series, adjacent time points are likely to be assigned to the same component. From Eq. (4), the marginal likelihood conditional on the latent variables $\mathbf{V}$ is given by

$$P(\mathcal{D}|\mathcal{G},\mathbf{V},\mathbf{K}) = \int P(\mathcal{D}|\mathcal{G},\mathbf{V},\mathbf{K},\boldsymbol{\theta})P(\boldsymbol{\theta})d\boldsymbol{\theta} = \prod_{n=1}^{N} \Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathcal{K}_n,\mathbf{V}_n]) \tag{5}$$

$$\Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathcal{K}_n,\mathbf{V}_n]) = \prod_{k=1}^{\mathcal{K}_n} \Psi(\mathcal{D}_n^{\pi_n}[k,\mathbf{V}_n]) \tag{6}$$

where the factors in Eq. (6) are given by:

$$\Psi(\mathcal{D}_n^{\pi_n}[k,\mathbf{V}_n]) = \int \prod_{t=2}^{m} P\Big(X_n(t) = \mathcal{D}_{n,t}|\pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)},\boldsymbol{\theta}_n^k\Big)^{\delta_{\mathbf{V}_n(t),k}} P(\boldsymbol{\theta}_n^k|\pi_n)d\boldsymbol{\theta}_n^k \tag{7}$$

Eq. (7) is similar to Eq. (3), and can be interpreted as a local BGe score restricted to the data subset $\mathcal{D}_n^{\pi_n}[k,\mathbf{V}_n] := \{(\mathcal{D}_{n,t},\mathcal{D}_{\pi_n,t-1}) : \mathbf{V}_n(t) = k, 2 \leq t \leq m\}$. The product $\Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathcal{K}_n,\mathbf{V}_n])$ in Eq. (6) is the *local cpBGe score* of X_n. Note that there is a factor for each mixture component k and that each factor $\Psi(\mathcal{D}_n^{\pi_n}[k,\mathbf{V}_n])$ can be interpreted as a local BGe score for the data subset $\mathcal{D}_n^{\pi_n}[k,\mathbf{V}_n]$.

When the regularity conditions defined in Geiger and Heckerman (1994) are satisfied, then the expression in Eq. (7) has a closed-form solution: it is given by Eq. (24) in Geiger and Heckerman (1994) restricted to the subset of the data pertaining to node X_n and its parents π_n that has been assigned to the kth mixture component (or kth segment).

The joint probability distribution of the proposed cpBGe model is given by:

$$P(\mathcal{G},\mathbf{V},\mathbf{K},\mathcal{D}) = P(\mathcal{G})P(\mathbf{V}|\mathbf{K})P(\mathbf{K})P(\mathcal{D}|\mathcal{G},\mathbf{V},\mathbf{K}) \tag{8}$$

We restrict on graph prior distributions that can be factorized into node-specific factors, symbolically: $P(\mathcal{G}) = \prod_{n=1}^{N} P(\pi_n)$ and in the absence of genuine prior knowledge about the regulatory network structure, we assume for $P(\pi_n)$ a uniform distribution. As done in our earlier work (Grzegorczyk and Husmeier, 2009) and in other Bayesian network studies (e.g. Friedman and Koller (2003) or Grzegorczyk and Husmeier (2008)) we impose a fan-in restriction on the cardinality of the parent node sets, symbolically: $|\pi_n| \leq 3$, to ensure sparsity of the inferred graph structures. Moreover, we assume that the distributions of the node-specific numbers of mixture components and allocation vectors $P(\mathbf{V}_n|\mathcal{K}_n)P(\mathcal{K}_n)$ are independent $(n = 1, \dots, N)$ so that the joint probability distribution in Eq. (8) can be factorized:

$$P(\mathcal{G},\mathbf{V},\mathbf{K},\mathcal{D}) = \prod_{n=1}^{N} P(\pi_n)P(\mathbf{V}_n|\mathcal{K}_n)P(\mathcal{K}_n)\Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathcal{K}_n,\mathbf{V}_n]) \tag{9}$$

Accordingly, the posterior distribution $P(\mathcal{G}, \mathbf{V}, \mathbf{K}|\mathcal{D})$ can be factorized into independent node-specific posterior distributions:

$$P(\mathcal{G}, \mathbf{V}, \mathbf{K}|\mathcal{D}) = \prod_{n=1}^{N} P(\pi_n, \mathbf{V}_n, \mathcal{K}_n|\mathcal{D}_n^{1:N}) \tag{10}$$

where $\mathcal{D}_n^{1:N} := \{(\mathcal{D}_{n,t}, \mathcal{D}_{1,t-1}, \dots, \mathcal{D}_{N,t-1}) : 2 \leq t \leq m\}$ contains the last $m-1$ observations $\mathcal{D}_{n,2}, \dots, \mathcal{D}_{n,m}$ of X_n and the first $m-1$ observations $\mathcal{D}_{j,1}, \dots, \mathcal{D}_{j,m-1}$ of all potential parent nodes X_j $(j = 1, \dots, N)$ of X_n. We note that each factor $P(\pi_n, \mathbf{V}_n, \mathcal{K}_n|\mathcal{D}_n^{1:N})$ in Eq. (10) can be infered independently.

As prior probability distributions on the node-specific numbers of mixture components $\mathcal{K}_n$, $P(\mathcal{K}_n)$, we take iid truncated Poisson distributions with shape parameter $\lambda = 1$, restricted to $1 \leq \mathcal{K}_n \leq \mathcal{K}_{MAX}$ (we set $\mathcal{K}_{MAX} = 10$ in our simulations). The prior distribution on the node-specific latent variable vectors, $P(\mathbf{V}_n|\mathcal{K}_n)$, is implicitly defined via a changepoint process. We identify $\mathcal{K}_n$ components with $\mathcal{K}_n - 1$ changepoints $\mathbf{b}_n = (b_{n,1}, \dots, b_{n,\mathcal{K}_n-1})$ on the *discrete* set $\{2, \dots, m-1\}$. For node X_n the observation at time point t is assigned to the kth component, symbolically $\mathbf{V}_n(t) = k$, if and only if $b_{n,k-1} < t \leq b_{n,k}$, where $b_{n,k}$ is the kth changepoint implied by $\mathbf{V}_n$, and $b_{n,0} = 1$ and $b_{n,\mathcal{K}_n} = m$ are two pseudo changepoints. There is a one-to-one mapping between allocation vectors and changepoints: For $t = 2, \dots, m$ and $k = 1, \dots, \mathcal{K}_n$: $b_{n,k-1} < t \leq b_{n,k} \Leftrightarrow \mathbf{V}_n(t) = k$. To make that more specific, we henceforth use the notation $\mathbf{b}_{\mathbf{V}_n} = (b_{\mathbf{V}_n,1}, \dots, b_{\mathbf{V}_n,\mathcal{K}_n-1})$ for the changepoint vector implied by $\mathbf{V}_n$. Following Green (1995) we assume that the changepoints are distributed as the even-numbered order statistics of $\mathcal{L} := 2(\mathcal{K}_n - 1) + 1$ points $u_1, \dots, u_{\mathcal{L}}$ uniformly and independently distributed on the set $\{2, \dots, m-1\}$. The even-numbered order statistics prior on the discrete changepoint locations induces the following prior distribution on the node-specific allocation vectors $P(\mathbf{V}_n|\mathcal{K}_n)$:

$$P(\mathbf{V}_n|\mathcal{K}_n) = \frac{1}{\binom{m-2}{2(\mathcal{K}_n-1)+1}} \prod_{k=0}^{\mathcal{K}_n-1} (b_{\mathbf{V}_n,k+1} - b_{\mathbf{V}_n,k} - 1) \tag{11}$$

where $b_{\mathbf{V}_n,0} = 1$ and $b_{\mathbf{V}_n,\mathcal{K}_n} = m$. We note that the even-numbered order statistics prior avoids changepoints at neighbouring time points t and $t+1$, and we have: $|b_{\mathbf{V}_n,k+1} - b_{\mathbf{V}_n,k}| > 1$ for $k = 0, \dots, \mathcal{K}_n - 1$.

In the following sections we discuss Metropolis-Hastings and Gibbs MCMC sampling schemes for sampling from the local posterior distributions $P(\pi_n, \mathbf{V}_n, \mathcal{K}_n|\mathcal{D}_n^{1:N})$ $(n = 1, \dots, N)$. The Metropolis-Hastings samplers employ local changepoint birth, death and reallocation moves on $(\mathcal{K}_n, \mathbf{V}_n)$, and the acceptance probabilities depend on $P(\mathcal{K}_n)P(\mathbf{V}_n|\mathcal{K}_n)$ ratios, which are straightforward to compute. For the Gibbs samplers, which include dynamic programming schemes to sample the changepoints from the correct posterior distribution, closed-form expressions for $P(\mathcal{K}_n)P(\mathbf{V}_n|\mathcal{K}_n)$ are crucial.

1.3 MCMC based model inference

1.3.1 Metropolis-Hastings sampling schemes

We now describe a Metropolis-Hastings (MH) MCMC algorithm to obtain a sample $\{\mathcal{G}^i, \mathbf{V}^i, \mathbf{K}^i\}_{i=1,\dots,I}$ from the posterior distribution $P(\mathcal{G}, \mathbf{V}, \mathbf{K}|\mathcal{D}) \propto P(\mathcal{G}, \mathbf{V}, \mathbf{K}, \mathcal{D})$ of Eq. (10). Our MH samplers combine the structure MCMC algorithm for Bayesian networks (Giudici and Castelo, 2003; Madigan and York, 1995) with the reversible jump MCMC sampling scheme for changepoints presented in Green (1995). This can be done straightforwardly, since conditional on the node-specific allocation vectors $\mathbf{V}_n$ the model parameters can be integrated out to obtain the local cpBGe scores $\Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathcal{K}_n, \mathbf{V}_n])$ in closed form, as shown in the previous Section 1.2. The resulting algorithm is effectively an RJMCMC scheme (Green, 1995) in the discrete space of network structures and latent allocation vectors, where the Jacobian in the acceptance criterion is always 1 and can be omitted. With probability $p_G = 0.5$ we perform a single edge move on the current graph $\mathcal{G}^i$ and leave the latent variable matrix and the numbers of mixture components unchanged, symbolically: $\mathbf{V}^{i+1} = \mathbf{V}^i$ and $\mathbf{K}^{i+1} = \mathbf{K}^i$. The new candidate graph is obtained by randomly selecting one of the domain nodes X_n and changing its parent set π_n^i by either adding or removing a parent node. There are $|\pi_n^i|$ nodes that can be removed from π_n^i and there are $N - |\pi_n^i|$ nodes that can be added to π_n^i, unless the maximal fan-in $\mathcal{F}$ is reached; for $|\pi_n^i| = \mathcal{F}$ no more edges can be added. This gives a set $\mathcal{N}(\pi_n^i)$ of new candidate parent sets with $|\mathcal{N}(\pi_n^i)| \in \{\mathcal{F}, N\}$ from which we randomly select a new candidate parent set π_n^{i+1}. The MH sampler proposes the new candidate graph $\mathcal{G}^{i+1}$ which results from $\mathcal{G}^i$ by replacing π_n^i by π_n^{i+1}, and the new graph is accepted with probability:

$$A(\mathcal{G}^{i+1}|\mathcal{G}^i) = min\left\{1, \frac{\Psi^{\dagger}(\mathcal{D}_n^{\pi_n^{i+1}}[\mathcal{K}_n^i, \mathbf{V}_n^i])}{\Psi^{\dagger}(\mathcal{D}_n^{\pi_n^i}[\mathcal{K}_n^i, \mathbf{V}_n^i])} \frac{P(\pi_n^{i+1})}{P(\pi_n^i)} \frac{|\mathcal{N}(\pi_n^i)|}{|\mathcal{N}(\pi_n^{i+1})|}\right\} \tag{12}$$

where $|.|$ is the cardinality, and the local $\Psi^{\dagger}(.)$ scores have been specified in Eq. (6). The graph is left unchanged, symbolically $\mathcal{G}^{i+1} := \mathcal{G}^i$, if the move is not accepted.

With the complementary probability $1 - p_G$ we leave the graph $\mathcal{G}^i$ unchanged and perform a move on $(\mathbf{V}^i, \mathbf{K}^i)$, where $\mathbf{V}_n^i$ is the latent variable vector of X_n in $\mathbf{V}^i$, and $\mathbf{K}^i = (\mathcal{K}_1^i, \ldots, \mathcal{K}_N^i)$. We randomly select a node X_n and change its current number of components $\mathcal{K}_n^i$ and its allocation vector $\mathbf{V}_n^i$ via a changepoint birth or death move, or we keep $\mathcal{K}_n^i$ and change its latent variable vector $\mathbf{V}_n^i$ by a changepoint re-allocation move along the lines of the RJMCMC algorithm of Green (1995).
The changepoint birth (death) move increases (decreases) $\mathcal{K}_n^i$ by 1 and changes $\mathbf{V}_n^i$ correspondingly. The changepoint reallocation move leaves $\mathcal{K}_n^i$ unchanged and modifies $\mathbf{V}_n^i$ only. If with probability $(1 - p_G)/N$ a changepoint move on $(\mathcal{K}_n^i, \mathbf{V}_n^i)$ is performed, we randomly draw the move type. Under fairly mild regularity conditions (ergodicity), the MH MCMC sampling scheme converges to the desired posterior distribution (Green, 1995) if the acceptance probabilities for the three changepoint moves $(\mathcal{K}_n^i, \mathbf{V}_n^i) \rightarrow (\mathcal{K}_n^{i+1}, \mathbf{V}_n^{i+1})$ are chosen of the form $min(1, R)$, with

$$R = \frac{\Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathcal{K}_n^{i+1}, \mathbf{V}_n^{i+1}])}{\Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathcal{K}_n^i, \mathbf{V}_n^i])} \times A \times B = \frac{\prod_{k=1}^{\mathcal{K}_n^{i+1}} \Psi(\mathcal{D}_n^{\pi_n}[k, \mathbf{V}_n^{i+1}])}{\prod_{k=1}^{\mathcal{K}_n^i} \Psi(\mathcal{D}_n^{\pi_n}[k, \mathbf{V}_n^i])} \times A \times B \tag{13}$$

where $A = P(\mathbf{V}_n^{i+1}|\mathcal{K}_n^{i+1})P(\mathcal{K}_n^{i+1})/P(\mathbf{V}_n^i|\mathcal{K}_n^i)P(\mathcal{K}_n^i)$ is the prior probability ratio, B is the inverse proposal probability ratio, and the $\Psi(.)^{\dagger}$- and $\Psi(.)$-terms have been specified in Eqn. (6) and (7).

In our implementation we choose $\mathcal{K}_n^i$-dependent proposal probabilities $b_{\mathcal{K}_n^i}$, $d_{\mathcal{K}_n^i}$, and $r_{\mathcal{K}_n^i}$ for birth (b), death (d) and re-allocation (r) moves. Like Green (1995) we set: $b_{\mathcal{K}_n^i} = c\ min\{1, \frac{P(\mathcal{K}_n^i+1)}{P(\mathcal{K}_n^i)}\}$ and $d_{\mathcal{K}_n^i} = c\ min\{1, \frac{P(\mathcal{K}_n^i-1)}{P(\mathcal{K}_n^i)}\}$ with the constant c as large as possible subject to the constraint $b_{\mathcal{K}_n^i} + d_{\mathcal{K}_n^i} \leq 0.9$ for all i so that the ratio of the proposal probabilities for birth versus death moves $d_{(\mathcal{K}_n^i+1)}/b_{\mathcal{K}_n^i}$ cancels out against the prior ratio $P(\mathcal{K}_n^i + 1)/P(\mathcal{K}_n^i)$. The proposal probability for a changepoint (re-)allocation move is given by: $r_{\mathcal{K}_n^i} = 1 - b_{\mathcal{K}_n^i} - d_{\mathcal{K}_n^i}$.

(i) For a changepoint reallocation (r) we randomly select one of the existing changepoints $b_{\mathbf{V}_n^i,j}$ from the vector $(b_{\mathbf{V}_n^i,1}, \ldots, b_{\mathbf{V}_n^i,\mathcal{K}_n-1})$, and the replacement value $b^{\dagger}$ is drawn from a uniform distribution on the discrete set $\left\{b_{\mathbf{V}_n^i,j-1} + 2, \ldots, b_{\mathbf{V}_n^i,j+1} - 2\right\}$ where $b_{V_n^i,0} = 1$ and $b_{\mathbf{V}_n^i,\mathcal{K}_n} = m$. The proposal probability ratio is one and the prior probabilities $P(\mathcal{K}_n^{i+1}) = P(\mathcal{K}_n^i)$ cancel out, $B_r = 1$. From Eq. (11) it can be seen that the remaining prior probability ratio is $P(\mathbf{V}_n^{i+1}|\mathcal{K}_n^{i+1})/P(\mathbf{V}_n^i|\mathcal{K}_n^i)$ is given by:

$$A_r = \frac{(b_{\mathbf{V}_n^i,j+1} - b^{\dagger} - 1)(b^{\dagger} - b_{\mathbf{V}_n^i,j-1} - 1)}{(b_{\mathbf{V}_n^i,j+1} - b_{\mathbf{V}_n^i,j} - 1)(b_{\mathbf{V}_n^i,j} - b_{\mathbf{V}_n^i,j-1} - 1)}, \tag{14}$$

If there is no changepoint ($\mathcal{K}_n^i = 1$) the move is rejected and the Markov chain is left unchanged.

(ii) If a changepoint birth move (b) on $(\mathcal{K}_n^i, \mathbf{V}_n^i)$ is proposed, the location of the new changepoint $b^\dagger$ is randomly drawn from a uniform distribution on the set of all valid new changepoint locations:

$$B^\dagger(\mathbf{V}_n^i) := \left\{b : 2 \leq b \leq m-1 \wedge \forall j \in \{1, \ldots, \mathcal{K}_n - 1\} : |b - b_{\mathbf{V}_n^i,j}| > 1\right\} \tag{15}$$

The new candidate changepoint $b^\dagger$ with $b_{\mathbf{V}_n^i,j} < b^\dagger < b_{\mathbf{V}_n^i,j+1}$ yields $\mathcal{K}_n^{i+1} = \mathcal{K}_n^i + 1$ mixture components and a new candidate allocation vector $\mathbf{V}_n^{i+1}$ in which one segment has been subdivided into 2 segments. The proposal probability for this move is $b_{\mathcal{K}_n^i}/|B^\dagger(\mathbf{V}_n^i)|$, where $|B^\dagger(\mathbf{V}_n^i)|$ is the number of valid changepoint locations for $b^\dagger$. The reverse death move, which is selected with probability $d_{(\mathcal{K}_n^i+1)}$, consists in discarding randomly one of the $(\mathcal{K}_n^i + 1) - 1 = \mathcal{K}_n^i$ changepoints from $(\mathcal{K}_n^{i+1}, \mathbf{V}_n^{i+1})$. The prior probability ratio A_b can be computed with Eq. (11):

$$A_b = \frac{P(\mathcal{K}_n^i + 1)}{P(\mathcal{K}_n^i)} \frac{(2\mathcal{K}_n^i + 1)(2\mathcal{K}_n^i)}{(m - 2\mathcal{K}_n^i - 1)(m - 2\mathcal{K}_n^i - 2)} \frac{(b_{\mathbf{V}_n^i,j+1} - b^\dagger - 1)(b^\dagger - b_{\mathbf{V}_n^i,j} - 1)}{(b_{\mathbf{V}_n^i,j+1} - b_{\mathbf{V}_n^i,j} - 1)}, \tag{16}$$

and the inverse proposal probability ratio is $B_b = \frac{d_{(\mathcal{K}_n^i+1)}|B^\dagger(\mathbf{V}_n^i)|}{(b_{\mathcal{K}_n^i}\mathcal{K}_n^i)}$. This can be simplified to:

$$A_b B_b = \frac{(2\mathcal{K}_n^i + 1)(2\mathcal{K}_n^i)}{(m - 2\mathcal{K}_n^i - 1)(m - 2\mathcal{K}_n^i - 2)} \frac{(b_{\mathbf{V}_n^i,j+1} - b^\dagger - 1)(b^\dagger - b_{\mathbf{V}_n^i,j} - 1)}{(b_{\mathbf{V}_n^i,j+1} - b_{\mathbf{V}_n^i,j} - 1)} \frac{|B^\dagger(\mathbf{V}_n^i)|}{\mathcal{K}_n^i} \tag{17}$$

For $\mathcal{K}_n^i = \mathcal{K}_{max}$ the birth of a new changepoint is invalid and the Markov chain is left unchanged.

(iii) A changepoint death move (d) on the current state $(\mathcal{K}_n^i, \mathbf{V}_n^i)$ is the reverse of the birth move. There are $\mathcal{K}_n^i - 1$ changepoints and we randomly select and delete one of them. Let $b^\dagger = b_{\mathbf{V}_n^i,j}$ be the selected changepoint and let $\mathbf{V}_n^{i+1}$ be the new candidate allocation vector after deletion of the selected changepoint $b^\dagger$. We obtain for the product of the prior probability ratio and the inverse proposal probability ratio:

$$A_d B_d = \frac{(m - 2\mathcal{K}_n^i - 3)(m - 2\mathcal{K}_n^i - 4)}{(2\mathcal{K}_n^i - 1)(2\mathcal{K}_n^i - 2)} \frac{(b_{\mathbf{V}_n^i,j+1} - b_{\mathbf{V}_n^i,j-1} - 1)}{(b_{\mathbf{V}_n^i,j+1} - b^\dagger - 1)(b^\dagger - b_{\mathbf{V}_n^i,j-1} - 1)} \frac{\mathcal{K}_n^i - 1}{|B^\dagger(\mathbf{V}_n^{i+1})|} \tag{18}$$

where $|B^\dagger(\mathbf{V}_n^{i+1})|$ is the number of valid new changepoint locations that can be added during a birth move. For $\mathcal{K}_n^i = 1$ there is no changepoint that can be deleted during a death move and the Markov chain is left unchanged.

1.4 Sampling parent node sets from the Boltzmann distribution

The Metropolis-Hastings (MH) sampler presented in Section 1.3.1 changes the current graph $\mathcal{G}$ by single-edge operations. An improvement can be achieved by sampling new parent node sets $\pi_n^\star$ for each node X_n directly from the posterior distribution:

$$P(\pi_n^\star|\mathcal{D}_n^{1:N}) = \frac{\Psi(\mathcal{D}_n^{\pi_n^\star})}{\sum_{\pi_n:|\pi_n|\leq\mathcal{F}} \Psi(\mathcal{D}_n^{\pi_n})} \tag{19}$$

where the local $\Psi(.)$-scores of the standard (homogeneous) DBN were specified in Eq. (3) and the sum is over all valid parent node sets π_n subject to a fan-in restrition $\mathcal{F}$. Eq. (19) is similar to Eq. (10) in Friedman and Koller (2003). The main difference is that Friedman and Koller (2003) apply this scheme to static Bayesian networks subject to an order constraint, where the latter has to be imposed on the system to render it modular. A DBN without intra-time-slice connectivities, on the other hand, is intrinsically modular, i.e. Eq. (19) exploits modularities that already exist and do not need to be enforced via an additional constraint.

In standard (homogeneous) DBNs the Boltzmann distributions can be pre-computed and stored for each node so that sampling from them may become computationally very effective and superior to MH samplers that are based on single edge operations. For our changepoint model it turns out that sampling from the Boltzmann distribution is ineffective, as the local scores depend on the node-specific changepoints and would have to be re-computed in every single MCMC step. In our cpBGe model we have the following node-specific Boltzmann distributions conditional on the number of changepoints $\mathcal{K}_n$ and the allocation vector $\mathbf{V}_n$:

$$P(\pi_n^\star|\mathcal{K}_n,\mathbf{V}_n,\mathcal{D}_n^{1:N}) = \frac{\Psi^\dagger(\mathcal{D}_n^{\pi_n^\star}[\mathcal{K}_n,\mathbf{V}_n])}{\sum_{\pi_n:|\pi_n|\leq\mathcal{F}}\Psi^\dagger(\mathcal{D}_n^{\pi_n}[\mathcal{K}_n,\mathbf{V}_n])} = \frac{\prod_{k=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n^\star}[k,\mathbf{V}_n])}{\sum_{\pi_n:|\pi_n|\leq\mathcal{F}}\prod_{k=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n}[k,\mathbf{V}_n])} \tag{20}$$

where the local cpBGe scores $\Psi^\dagger(\mathcal{D}_n^{\pi_n}[\mathcal{K}_n,\mathbf{V}_n])$ and the local BGe scores $\Psi(\mathcal{D}_n^{\pi_n}[k,\mathbf{V}_n])$ can be computed with Eqn. (6) and (7). Although the three changepoint moves affect only two local BGe scores in the products, the re-computation of the Boltzmann distribution after each changepoint move becomes computationally expensive. The bottleneck becomes obvious when taking into consideration that the three changepoint moves give relatively small steps in the configuration space of the allocation vector $\mathbf{V}_n$ so that a large amount of re-computations is required.

In Sections 1.5 and 1.6 we will discuss a dynamic programming scheme for sampling the node-specific numbers of changepoints $\mathcal{K}_n$ and the node-specific allocation vectors $\mathbf{V}_n$ directly from the conditional posterior distribution: $P(\mathbf{V}_n,\mathcal{K}_n|\pi_n,\mathcal{D}_n^{\pi_n})$. This dynamic programming scheme for sampling from $P(\mathbf{V}_n,\mathcal{K}_n|\pi_n,\mathcal{D}_n^{\pi_n})$ in combination with sampling parent node configurations π_n from the Boltzmann distribution $P(\pi_n|\mathcal{K}_n,\mathbf{V}_n,\mathcal{D}_n^{1:N})$ can be used to construct a Gibbs MCMC sampling scheme.

1.5 Sampling changepoints by dynamic programming

In the proposed cpBGe model we have a parent node set π_n, a number of components $\mathcal{K}_n$, and an allocation vector $\mathbf{V}_n$ for each domain node X_n $(n = 1,\ldots,N)$. $\mathcal{K}_n$ can be identified with $\mathcal{K}_n - 1$ changepoints on the *discrete* set $\{2,\ldots,m-1\}$ and there is a one-to-one mapping between $\mathbf{V}_n$ and the changepoint vector $\mathbf{b}_{\mathbf{V}_n} := (b_{\mathbf{V}_n,0},\ldots,b_{\mathbf{V}_n,\mathcal{K}_n})$ where $b_{\mathbf{V}_n,0} = 1$ and $b_{\mathbf{V}_n,\mathcal{K}_n} = m$ are pseudo changepoints.

We now want to apply a dynamic programming scheme to sample for each domain node X_n from the joint posterior distribution of $(\mathcal{K}_n,\mathbf{V}_n)$ conditional on the parent node set π_n:

$$P(\mathcal{K}_n,\mathbf{V}_n|\pi_n,\mathcal{D}_n^{\pi_n}) = P(\mathcal{K}_n|\pi_n,\mathcal{D}_n^{\pi_n})P(\mathbf{V}_n|\mathcal{K}_n,\pi_n,\mathcal{D}_n^{\pi_n}) \tag{21}$$

where $\mathcal{D}_n^{\pi_n}$ denotes the set of observations $\{(\mathcal{D}_{n,i},\mathcal{D}_{\pi_n,i-1}) : 2 \leq i \leq m\}$ pertaining to node X_n and its parent node set π_n. Accordingly, let $\mathcal{D}_n^{\pi_n}[s:t]$ denote the sub-segment $\{(\mathcal{D}_{n,i},\mathcal{D}_{\pi_n,i-1}) : s \leq i \leq t\}$ of adjacent observations, and we also define $\mathcal{D}_{n,s:t} = \{\mathcal{D}_{n,i} : s \leq i \leq t\}$ and $\mathcal{D}_{\pi_n,s:t} = \{\mathcal{D}_{\pi_n,i} : s-1 \leq i \leq t-1\}$

The local cpBGe score $\Psi^\dagger(\mathcal{D}_n^{\pi_n}[\mathcal{K}_n,\mathbf{V}_n])$ of X_n is the probability of the observations $\mathcal{D}_{n,2:m}$ of X_n given the parent set π_n and its observations $\mathcal{D}_{\pi_n,2:m}$, $\mathcal{K}_n$ mixture components, and the allocation vector $\mathbf{V}_n$. The local score of X_n can be factorized using Eq. (6). Mapping the allocation vector $\mathbf{V}_n$ onto the changepoint vector $\mathbf{b}_{\mathbf{V}_n}$ we obtain as alternative representation:

$$\Psi^\dagger(\mathcal{D}_n^{\pi_n}[\mathcal{K}_n,\mathbf{V}_n]) = P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,2:m},\mathcal{K}_n,\mathbf{b}_{\mathbf{V}_n}) = \prod_{k=0}^{\mathcal{K}_n-1}\Psi(\mathcal{D}_n^{\pi_n}[(b_{\mathbf{V}_n,k}+1):b_{\mathbf{V}_n,k+1}]) \tag{22}$$

When just conditioning on $\mathcal{K}_n$ with $\mathcal{K}_n > 1$, we obtain the following marginal distribution:

$$P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,2:m},\mathcal{K}_n) = \sum_{\mathbf{b}_n\in\mathcal{B}(\mathcal{K}_n)} P(\mathbf{b}_n)\prod_{k=0}^{\mathcal{K}_n-1}\Psi(\mathcal{D}_n^{\pi_n}[(b_{n,k}+1):b_{n,k+1}]) \tag{23}$$

where $\mathcal{B}(\mathcal{K}_n)$ is the set of all valid changepoint vectors $\mathbf{b}_n = (b_{n,0}, \dots, b_{n,\mathcal{K}_n})$ of cardinality $\mathcal{K}_n + 1$ with $b_{n,i+1} - b_{n,i} > 1$, $b_{n,0} = 1$ and $b_{n,\mathcal{K}_n} = m$, and $P(\mathbf{b}_n) = P(\mathbf{V}_n(b_n))$ is the prior probability of the unique allocation vector $\mathbf{V}_n(b_n)$ and can be computed with Eq. (11) after having extracted the allocation vector $\mathbf{V}_n(\mathbf{b}_n)$ from $\mathbf{b}_n$. Now we additionally fix the j-th changepoint location, symbolically: $b_{n,j} = t - 1$, and restrict on the data sub-segment $\mathcal{D}_n^{\pi_n}[t : m]$:

$$P(\mathcal{D}_{n,t:m}|\mathcal{D}_{\pi_n,t:m}, \mathcal{K}_n, b_{n,j} = t-1) = \sum_{\mathbf{b}_n^j \in \mathcal{B}^j(\mathcal{K}_n|b_{n,j}=t-1)} P(\mathbf{b}_n^j) \prod_{k=j}^{\mathcal{K}_n - 1} \Psi(\mathcal{D}_n^{\pi_n}[(b_{n,k}+1) : b_{n,k+1}]) \qquad (24)$$

where $\mathcal{B}^j(\mathcal{K}_n|b_{n,j} = t-1)$ is the set of all valid changepoint vectors $\mathbf{b}_n^j = (b_{n,j+1}, \dots, b_{n,\mathcal{K}_n})$ on the discrete interval $\{t+1, \dots, m-2\}$ with $b_{n,i+1} - b_{n,i} > 1$, $b_{n,j} = t - 1$ and $b_{n,\mathcal{K}_n} = m$. Different from Eq. (23) the prior probability $P(\mathbf{b}_n^j)$ of the changepoint subset $\mathbf{b}_n^j$ cannot be computed in closed-form for $j > 0$.

For $\mathcal{K}_n > 1$ and $j = 0, \dots, \mathcal{K}_n - 1$ we set $Q_j^{\mathcal{K}_n}(t|n, \pi_n) = P(\mathcal{D}_{n,t:m}|\mathcal{D}_{\pi_n,t:m}, \mathcal{K}_n, b_{n,j} = t-1)$ for $t = 2(j+1), \dots, m - 2(\mathcal{K}_n - j) + 1$ and let $Q_j^{\mathcal{K}_n}(t|n, \pi_n)$ be zero otherwise, i.e. for $t < 2(j+1)$ and $t > m - 2(\mathcal{K}_n - j) + 1$.

It can be seen from Eq. (23) that $Q_0^{\mathcal{K}_n}(2|n, \pi_n)$ is equal to $P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,2:m}, \mathcal{K}_n)$, since $b_{n,0} = 1$ is a fixed pseudo changepoint, and we have for $t = 2\mathcal{K}_n, \dots, m-1$:

$$Q_{\mathcal{K}_n - 1}^{\mathcal{K}_n}(t|n, \pi_n) = \Psi(\mathcal{D}_n^{\pi_n}[t : m]) \qquad (25)$$

so that the Q terms can be computed straightforwardly for $j = \mathcal{K}_n - 1$.

Afterwards – as a special case of the recursions given in Fearnhead (2006) – we obtain the following recursions: For $\mathcal{K}_n > 1$, $j = 0, \dots, \mathcal{K}_n - 2$ and $t = 2(j+1), \dots, m - 2(\mathcal{K}_n - j) + 1$:

$$Q_j^{\mathcal{K}_n}(t|n, \pi_n) = \sum_{s=t+1}^{m-2(\mathcal{K}_n - j - 1)} \Psi(\mathcal{D}_n^{\pi_n}[t : s]) Q_{j+1}^{\mathcal{K}_n}(s+1|n, \pi_n) P(b_{n,j} = t-1|b_{n,j+1} = s, \mathcal{K}_n) \qquad (26)$$

where the bounds of t as well as the upper summation index allow for the changepoints that still need to be included[1].

In our changepoint model the probability distribution $P(b_{n,j} = t-1|b_{n,j+1} = s, \mathcal{K}_n)$ of changepoint $b_{n,j}$ conditional on $\mathcal{K}_n$ changepoints and the $b_{n,j+1}$ changepoint being located at time point s cannot be computed in closed-form. Following Fearnhead (2006) we set:

$$P(b_{n,j} = t-1|b_{n,j+1} = s, \mathcal{K}_n) = P(m, \mathcal{K}_n, s, t) := \frac{s - t}{\begin{pmatrix} m-2 \\ 2(\mathcal{K}_n - 1) + 1 \end{pmatrix}} \qquad (27)$$

This is a 'computational trick' which also yields: $Q_0^{\mathcal{K}_n}(2|n, \pi_n) = P(\mathcal{D}_n^{\pi_n}|\mathcal{K}_n)$ (Fearnhead, 2006). Thus, the modified recursions can be employed to compute: $P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,2:m}, \mathcal{K}_n)$ for $\mathcal{K}_n = 2, \dots, \mathcal{K}_{MAX}$. Note that there is no changepoint for $\mathcal{K}_n = 1$ so that the local cpBGe score (see Eq. (6)) is equal to the local BGe score of X_n (see Eq. (3)).

$$P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,2:m}, \mathcal{K}_n = 1) = \Psi(\mathcal{D}_n^{\pi_n}) \qquad (28)$$

Subsequently, the marginal posterior probability of the number of mixture components $\mathcal{K}_n$ can be computed as follows:

$$P(\mathcal{K}_n = k^\star|\mathcal{D}_{n,2:m}, \mathcal{D}_{\pi_n,2:m}) = \frac{P(\mathcal{K}_n = k^\star) P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,2:m}, \mathcal{K}_n = k^\star)}{\sum_{k=1}^{\mathcal{K}_{MAX}} P(\mathcal{K}_n = k) P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,2:m}, \mathcal{K}_n = k)} \qquad (29)$$

[1] Note that there must be room for including $j-1$ changepoints $b_{n,1}, \dots, b_{n,j-1}$ on the locations $2, \dots, t-2$ with $b_{n,j} - b_{n,j-1} > 1$ $(j = 1, \dots, j)$, $b_{n,0} = 1$ and $b_{n,j} = t-1$. And there must be room for $\mathcal{K}_n - 1 - j$ changepoints $b_{n,j+1}, \dots, b_{n,\mathcal{K}_n - 1}$ on the locations $t, \dots, m-1$ with $b_{n,j} - b_{n,j-1} > 1$ $(j = j+1, \dots, \mathcal{K}_n)$, $b_{n,j} = t-1$ and $b_{n,\mathcal{K}_n} = m$.

where $P(\mathcal{K}_n)$ is a Poisson distribution with $\lambda = 1$ truncated to $1 \leq \mathcal{K}_n \leq \mathcal{K}_{MAX}$ in our cpBGe model.
After having sampled $\mathcal{K}_n = k$ from $P(\mathcal{K}_n|\mathcal{D}_{n,2:m}, \mathcal{D}_{\pi_n,2:m})$, we can sample an allocation vector $\mathbf{V}_n$ from $P(\mathbf{V}_n|\mathcal{K}_n = k, \mathcal{D}_{n,2:m}, \mathcal{D}_{\pi_n,2:m})$ by sampling the j-th changepoint $b_{\mathbf{V}_n,j}$ conditional on the $(j-1)$-th changepoint $b_{\mathbf{V}_n,j-1}$ for $j = 1, \ldots, k-1$ from the following distribution:

$$P(b_{\mathbf{V}_n,j} = s|b_{\mathbf{V}_n,j-1}, \mathcal{D}_n^{\pi_n}, \mathcal{K}_n = k) = \frac{\Psi(\mathcal{D}_n^{\pi_n}[(b_{\mathbf{V}_n,j-1}+1) : s])Q_j^k(s+1|n,\pi_n)P(m,k,s,b_{\mathbf{V}_n,j-1}+1)}{Q_{j-1}^k(b_{\mathbf{V}_n,j-1}+1|n,\pi_n)} \quad (30)$$

as shown in Fearnhead (2006). The dynamic programming scheme works as follows: (i) We sample $\mathcal{K}_n = k$ from Eq. (29). (ii) For $k = 1$ we have no changepoints and for $k > 1$ we can subsequently employ Eq. (30) to sample the locations of the $k-1$ changepoints. Because of the one-to-one mapping between changepoints and allocation vectors, the sampled changepoints $b_{\mathbf{V}_n,1}, \ldots, b_{\mathbf{V}_n,k-1}$ give a unique allocation vector $\mathbf{V}_n$ which can be seen as directly sampled from $P(\mathbf{V}_n|\mathcal{K}_n = k, \mathcal{D}_{n,2:m}, \mathcal{D}_{\pi_n,2:m})$.
As a summary: By employing the dynamic programming scheme presented in this Section for each node X_n with parent set π_n, the number of mixture components $\mathcal{K}_n$ and the allocation vector $\mathbf{V}_n$ can be sampled from the conditional posterior distribution of $P(\mathcal{K}_n, \mathbf{V}_n|\pi_n, \mathcal{D}_n^{\pi_n})$.

1.6 Sampling changepoints from a point process prior

As shown by Fearnhead (2006) the computational costs of the dynamic programming scheme can be reduced by a slightly modified prior distribution for $(\mathcal{K}_n, \mathbf{V}_n)$. Instead of modelling $P(\mathcal{K}_n)$, and afterwards the allocation vectors $\mathbf{V}_n$ conditional on $\mathcal{K}_n$, a point process prior can be used to model the distances between successive changepoints. In the point process model $g(t)$ $(t = 1, 2, 3, \ldots)$ denotes the prior probability that there are t time points between two successive changepoints $b_{n,j-1}$ and $b_{n,j}$ on the discrete interval $\{2, \ldots, m-1\}$. The prior probability of $\mathcal{K}_n - 1$ changepoints being located at time points $b_{n,1}, \ldots, b_{n,\mathcal{K}_n-1}$ is:

$$P(b_{n,1}, \ldots, b_{n,\mathcal{K}_n-1}) = g_0(b_{n,1}) \left(\prod_{j=2}^{\mathcal{K}_n-1} g(b_{n,j} - b_{n,j-1}) \right) (1 - G(b_{n,\mathcal{K}_n} - b_{n,\mathcal{K}_n-1})) \quad (31)$$

where $b_{n,0} = 1$ and $b_{n,\mathcal{K}_n} = m$ are again pseudo changepoints, $G(t) = \sum_{s=1}^{t} g(t)$, and $g_0(.)$ is the prior distribution for the first changepoint $b_{n,1}$. For $g(.)$ the probability mass function of the negative binomial distribution NBIN(p,a) with parameters p and a can be used:

$$g(t) = \begin{pmatrix} t-1 \\ a-1 \end{pmatrix} p^k (1-p)^{t-a} \quad (32)$$

In a point process model on the positive *and* negative integers the probability mass function of the first changepoint $b_{n,1} \in \{2, \ldots, m-1\}$ is a mixture of k negative binomial distributions:

$$g_0(b_{n,1}) = \frac{1}{k} \sum_{i=1}^{k} \begin{pmatrix} (b_{n,1}-1)-1 \\ i-1 \end{pmatrix} p^i (1-p)^{(b_{n,1}-1)-i} \quad (33)$$

For each node X_n we define $Q(t|n,\pi_n)$ as the probability of its observations $\mathcal{D}_{n,t:m}$ given the observations $\mathcal{D}_{\pi_n,(t-1):(m-1)}$ of π_n and a changepoint $b^\dagger$ at time point $t-1$ $(t = 2, \ldots, m)$:

$$Q(t|n,\pi_n) = P(\mathcal{D}_{n,t:m}|\mathcal{D}_{\pi_n,(t-1):(m-1)}, b^\dagger = t-1) \quad (34)$$

$Q(m|n,\pi_n)$ is then equal to $\Psi(\mathcal{D}_n^{\pi_n}[m : m])$, defined below Eq. (21). For $t = 3, \ldots, m-1$ the following recursion can be used:

$$Q(t|n,\pi_n) = \left(\sum_{s=t}^{m-1} \Psi(\mathcal{D}_n^{\pi_n}[t : s])Q(s+1|n,\pi_n)g(s+1-t) \right) + \Psi(\mathcal{D}_n^{\pi_n}[t : m])(1 - G(m-t)) \quad (35)$$

and

$$Q(2|n,\pi_n) = \left(\sum_{s=2}^{m-1} \Psi(\mathcal{D}_n^{\pi_n}[2:s])Q(s+1|n,\pi_n)g_0(s-1)\right) + \Psi(\mathcal{D}_n^{\pi_n})(1-G_0(m-2)) \quad (36)$$

where $G_0(t) = \sum_{s=1}^{t} g_0(s)$. The posterior distribution of the first changepoint $b_{n,1}$ given the parent set π_n is:

$$P(b_{n,1} = t|\mathcal{D}_n^{\pi_n}) = \Psi(\mathcal{D}_n^{\pi_n}[2:t])Q(t+1|n,\pi_n)\frac{g_0(t-1)}{Q(2|n,\pi_n)} \quad (37)$$

for $t = 2,\ldots,m-1$ and the probability of no changepoint ($P(\mathcal{K}_n = 1)$) is given by:

$$P(\mathcal{K}_n = 1|\pi_n,\mathcal{D}_n^{\pi_n}) = \Psi(\mathcal{D}_n^{\pi_n}[2:m])\frac{1-G_0(m-2)}{Q(2|n,\pi_n)} \quad (38)$$

The posterior distribution of the j-th changepoint $b_{n,j}$ given the parent node set π_n and the previous changepoint $b_{n,j-1}$ is:

$$P_t := P(b_{n,j} = t|b_{n,j-1},\mathcal{D}_n^{\pi_n}) = \Psi(\mathcal{D}_n^{\pi_n}[(b_{n,j-1}+1):t])Q(t+1|n,\pi_n)\frac{g(t-b_{n,j-1})}{Q(b_{n,j-1}+1|n,\pi_n)} \quad (39)$$

for $t = b_{n,j-1}+1,\ldots,m-1$ and the probability of no further changepoint is given by:

$$P_{\geq m} := \Psi(\mathcal{D}_n^{\pi_n}[(b_{n,j-1}+1):m])\frac{1-G_0(m-b_{n,j-1}-1)}{Q(b_{n,j-1}+1|n,\pi_n)} \quad (40)$$

Consequently, if there is a changepoint at $b_{n,j-1} = t$, then the location of the next changepoint can be sampled from the discrete mass probability distribution $[P_{b_{n,j-1}+1},\ldots,P_{m-1},P_{\geq m}]$ where $P_{\geq m}$ is the probability for no further changepoints. Having sampled changepoints $b_{n,1},\ldots,b_{n,k-1}$ from these conditional distributions, the number of mixture components is $\mathcal{K}_n = k$ and the allocation vector $\mathbf{V}_n$ can be computed from the changepoints.
As a summary: For each node X_n with parent set π_n, $(\mathcal{K}_n,\mathbf{V}_n)$ can be sampled from $P(\mathcal{K}_n,\mathbf{V}_n|\pi_n,\mathcal{D}_n^{\pi_n})$ when the prior distribution $P(\mathcal{K}_n,\mathbf{V}_n)$ is replaced by a point-process model as described above.

1.7 Sampling changepoints from a point process prior for the regularized cpBGe model

The dynamic programming scheme presented in Section 1.6 can also be used to sample changepoints for the novel regularized cpBGe model. In this Section we describe the modifications that have to be made. We employ the same point process prior for cluster-specific changepoints. The prior probability that there are $\mathcal{K}_i - 1$ changepoints being located at time points $b_{i,1},\ldots,b_{i,\mathcal{K}_i-1}$ for the nodes in the ith cluster is given by:

$$P(b_{i,1},\ldots,b_{i,\mathcal{K}_i-1}) = g_0(b_{i,1})\left(\prod_{j=2}^{\mathcal{K}_i-1} g(b_{i,j}-b_{i,j-1})\right)(1-G(b_{i,\mathcal{K}_i}-b_{i,\mathcal{K}_i-1})) \quad (41)$$

where $g(.)$, $G(.)$, $g_0(.)$ have been specified in Section 1.6 (see Eqn. (32)-(33)).

Different from the original cpBGe model we now have changepoints for each of c clusters of nodes induced by the clustering $\mathcal{C}$ rather than node-specific changepoints for each individual node. We want to sample changepoints for each cluster, which are then common to all the nodes in that cluster. We consider the ith cluster ($1 \leq i \leq c$), that is the set of nodes $\{X_n : \mathcal{C}(n) = i\}$. The nodes in the ith cluster share K_i components and there is a set of changepoints $b_{i,1},\ldots,b_{i,\mathcal{K}_i-1}$

that can be mapped onto the allocation vector of the ith cluster $\mathbf{V}_i^{\mathcal{C}}$: $\mathbf{V}_i^{\mathcal{C}}(t) = k \Leftrightarrow b_{i,k-1} < t \leq b_{i,k}$ $(t = 2, \dots, m$ and $k = 1, \dots, \mathcal{K}_i)$.

We define $Q(t|i, C, \mathcal{G})$ as the probability of the observations for the nodes in the ith cluster $\mathcal{D}_{n,t:m}$ $(n : \mathcal{C}(n) = i)$ conditional on the corresponding realisations of the parent nodes $\mathcal{D}_{\pi_n,(t-1):(m-1)}$ $(n : \mathcal{C}(n) = i)$ *and* a changepoint $b^\dagger$ at time point $t - 1$ $(t = 2, \dots, m)$.

For $t = m$ we have:

$$Q(m|i, \mathcal{C}, \mathcal{G}) = \prod_{n:\mathcal{C}(n)=i} \Psi(\mathcal{D}_n^{\pi_n}[m : m]) \tag{42}$$

and for $t = 3, \dots, m - 1$ the same recursion as in Section 1.6 can be used:

$$\begin{aligned} Q(t|i, \mathcal{C}, \mathcal{G}) &= \sum_{s=t}^{m-1} \left(\prod_{n:\mathcal{C}(n)=i} \Psi(\mathcal{D}_n^{\pi_n}[t : s]) \right) Q(s + 1|i, \mathcal{C}, \mathcal{G}) g(s + 1 - t) && (43) \\ &+ \left(\prod_{n:\mathcal{C}(n)=i} \Psi(\mathcal{D}_n^{\pi_n}[t : m]) \right) (1 - G(m - t)) && (44) \end{aligned}$$

and

$$\begin{aligned} Q(2|i, \mathcal{C}, \mathcal{G}) &= \sum_{s=2}^{m-1} \left(\prod_{n:\mathcal{C}(n)=i} \Psi(\mathcal{D}_n^{\pi_n}[2 : s]) Q(s + 1|i, \mathcal{C}, \mathcal{G}) \right) g_0(s - 1) && (45) \\ &+ \left(\prod_{n:\mathcal{C}(n)=i} \Psi(\mathcal{D}_n^{\pi_n}) \right) (1 - G_0(m - 2)) && (46) \end{aligned}$$

where $G_0(t) = \sum_{s=1}^{t} g_0(s)$. The posterior distribution of the first changepoint $b_{i,1}$ of cluster i given the graph $\mathcal{G}$ that implies the parent sets for the nodes in the ith cluster, symbolically $\{\pi_n | n : \mathcal{C}(n) = i\}$, is:

$$P(b_{i,1} = t|\mathcal{G}, \mathcal{C}, i) = \left(\prod_{n:\mathcal{C}(n)=i} \Psi(\mathcal{D}_n^{\pi_n}[2 : t]) \right) Q(t + 1|i, \mathcal{C}, \mathcal{G}) \frac{g_0(t - 1)}{Q(2|i, \mathcal{C}, \mathcal{G})} \tag{47}$$

for $t = 2, \dots, m - 1$ and the probability of no changepoint for the ith cluster $(P(\mathcal{K}_i = 1))$ is given by:

$$P(\mathcal{K}_i = 1|i, \mathcal{C}, \mathcal{G}) = \left(\prod_{n:\mathcal{C}(n)=i} \Psi(\mathcal{D}_n^{\pi_n}[2 : m]) \right) \frac{1 - G_0(m - 2)}{Q(2|i, \mathcal{C}, \mathcal{G})} \tag{48}$$

The posterior distribution of the j-th changepoint for the ith cluster $b_{i,j}$ given the parent node sets π_n $(\{n : \mathcal{C}(n) = i\})$ and the previous changepoint $b_{i,j-1}$ is:

$$P_t := P(b_{i,j} = t|b_{i,j-1}, i, \mathcal{C}, \mathcal{G}) = \left(\prod_{n:\mathcal{C}(n)=i} \Psi(\mathcal{D}_n^{\pi_n}[(b_{i,j-1} + 1) : t]) \right) Q(t{+}1|i, \mathcal{C}, \mathcal{G}) \frac{g(t - b_{i,j-1})}{Q(b_{i,j-1} + 1|i, \mathcal{C}, \mathcal{G})} \tag{49}$$

for $t = b_{i,j-1} + 1, \dots, m - 1$ and the probability of no further changepoint is given by:

$$P_{\geq m} := \left(\prod_{n:\mathcal{C}(n)=i} \Psi(\mathcal{D}_n^{\pi_n}[(b_{i,j-1} + 1) : m]) \right) \frac{1 - G_0(m - b_{i,j-1} - 1)}{Q(b_{i,j-1} + 1|i, \mathcal{C}, \mathcal{G})} \tag{50}$$

We note that Eqn. (49-50) are the regularized cpBGe equivalents of Eqn. (39-40) in Section 1.6. If there is a changepoint at $b_{i,j-1} = t$, then the location of the next changepoint can be sampled from the discrete mass probability distribution $[P_{b_{i,j-1}+1}, \dots, P_{m-1}, P_{\geq m}]$ where $P_{\geq m}$ is the probability for no further changepoints. Having sampled changepoints $b_{i,1}, \dots, b_{i,k-1}$ from these conditional distributions, the number of mixture components for the ith cluster of nodes is $\mathcal{K}_i = k$ and the allocation vector for the nodes in the ith cluster $\mathbf{V}_i^{\mathcal{C}}$ can be extracted from the changepoints: $\mathbf{V}_i^{\mathcal{C}}(t) = k \Leftrightarrow b_{i,k-1} < t \leq b_{i,k}$ $(t = 2, \dots, m$ and $k = 1, \dots, \mathcal{K}_i)$.

As a summary: Conditional on the graph $\mathcal{G}$ for each cluster i $(1 \leq i \leq c)$ the number of changepoints and the changepoint locations, symbolically $(\mathcal{K}_i, \mathbf{V}_i^{\mathcal{C}})$ can be sampled from $P(\mathcal{K}_i, \mathbf{V}_i^{\mathcal{C}} | i, \mathcal{C}, \mathcal{G})$.

With regard to Section 1.9 we note that $Q(t|i, \mathcal{C}, \mathcal{G})$ was defined such that we have for $t = 2$:

$$Q(2|i, \mathcal{C}, \mathcal{G}) = \sum_{\mathbf{V}_j^{\mathcal{C}}} P(\mathbf{V}_j^{\mathcal{C}} | \mathcal{C}) \prod_{n:\mathcal{C}(n)=j} \Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathbf{V}_j^{\mathcal{C}}]) \tag{51}$$

where the sum is over all possible allocation vectors $\mathbf{V}_j^{\mathcal{C}}$ for the jth cluster induced by the clustering $\mathcal{C}$. The probability of the observations for the nodes in the jth cluster $\mathcal{D}_{n,2:m}$ $(n : \mathcal{C}(n) = j)$ conditional on the corresponding realisations of the parent nodes $\mathcal{D}_{\pi_n,1:m-1}$ $(n : \mathcal{C}(n) = j)$ can be thought of as the marginal distribution over all possible allocation vectors. In Section 1.9 and in the main paper we refer to $Q(2|j, \mathcal{C}, \mathcal{G})$ as $Q_j(\mathcal{D}, \mathcal{C}, \mathcal{G})$, and as we have seen in this section $Q_j(\mathcal{D}, \mathcal{C}, \mathcal{G}) = Q(2|j, \mathcal{C}, \mathcal{G})$ can be computed efficiently by applying the recursions of Fearnhead (2006).

1.8 MCMC convergence

For our Matlab implementation of the cpBGe model we observed for the Arabidopsis data sets with $N = 9$ variables and $m = 49$ data points that the computational costs of 2000 MCMC iterations of the Metropolis-Hastings (MH) RJMCMC sampling scheme are comparable to the computational costs of approximately 1 Gibbs sampling step, when the same Poisson/changepoint process prior is used and the maximal number of components is set to $\mathcal{K}_{MAX} = 10$. Each single Metropolis-Hastings step proposes the change of either a parent node set π_n *or* a node-specific allocation vector $\mathbf{V}_n$. Each Gibbs iteration consists of two steps, i.e. a new parent node set π_n *and* a new node-specific allocation vector $\mathbf{V}_n$ are sampled. We refer to this Gibbs sampler as *Gibbs*$(K = 10)$. We tried two other variants of this Gibbs sampling scheme, with the objective to increase the number of Gibbs steps at the same computational costs. (i) Setting $\mathcal{K}_{MAX} = 5$ approximately halves the computational costs of the Gibbs sampler, so that 2 moves are approximately as expensive as 2000 MH iterations. We refer to this version of the Gibbs sampler as Gibbs(K=5). (ii) For the Poisson/changepoint process prior with the hyperparameters $p = 0.05$ and $a = 2$ of the negative binomial distribution gained a tenfold increase in the number of Gibbs steps at the same computational costs. We will refer to this version of the Gibbs sampler as Gibbs-NBIN, and we note that performing 10 Gibbs-NBIN steps required the same computational costs as 2000 MH steps.

During the sampling phase the cpBGe model outputs a graph sample $\mathcal{G}^1, \dots, \mathcal{G}^I$ from the posterior distribution from which marginal edge posterior probabilities can be computed. For a network domain with N nodes an estimator $e_{n,j}$ for the marginal posterior probability of the individual edge $X_n \to X_j$ $(\mathcal{G}(n,j))$ is given by:

$$e_{n,j} = \frac{1}{I} \sum_{i=1}^{I} \mathcal{G}^i(n,j) \tag{52}$$

where $\mathcal{G}^i(n,j)$ is an indicator function which is 1 if the ith graph in the sample contains the edge $X_n \to X_j$, and 0 otherwise ($n,j \in \{1,\dots,N\}$). A standard diagnostic that we apply to evaluate convergence is based on potential scale reduction factors (PSRFs), which are usually monitored along the number of MCMC iterations. In the following representation we assume that H independent MCMC simulations with $2s$ iterations each have been performed on the same single data set. Discarding the first s iterations as burn-in phase, I_s graph samples can be taken from the remaining s MCMC iterations. Note that the number of samples I_s that can be taken in the sampling-phase is limited by the number of MCMC iterations s and the distance (no. of iterations) between samples.

For each of the H independent MCMC simulations $h = 1,\dots,H$ we compute the posterior probabilities of all edges $e_{n,j,h}$ ($n,j \in \{1,\dots,N\}$) from the graph samples $\mathcal{G}^{h,1},\dots,\mathcal{G}^{h,I_s}$ as described above. For each individual edge $X_n \to X_j$ the 'between-chain' variance $\mathcal{B}(n,j)$ and the 'within-chain' variance $\mathcal{W}(n,j)$ of its edge posterior probability are defined as (see Brooks and Gelman (1998)):

$$\mathcal{B}(n,j) = \frac{1}{H-1}\sum_{h=1}^{H}(e_{n,j,h} - \bar{e}_{n,j,.})^2 \tag{53}$$

where $\bar{e}_{n,j,.}$ is the mean of $e_{n,j,1},\dots,e_{n,j,H}$, and:

$$\mathcal{W}(n,j) = \frac{1}{H(I_s-1)}\sum_{h=1}^{H}\sum_{i=1}^{I_s}(G^{h,i}(n,j) - e_{n,j,h})^2 \tag{54}$$

where $G^{h,i}(n,j)$ is 1 if the ith graph in the sample taken in the hth simulation contains the edge $X_n \to X_j$, and 0 otherwise. Following Brooks and Gelman (1998) the $PSRF(n,j)$ of the individual edge $X_n \to X_j$ is then given by:

$$PSRF(n,j) = \frac{(1-\frac{1}{I_s})\mathcal{W}(n,j) + (1+\frac{1}{H})\mathcal{B}(n,j)}{\mathcal{W}(n,j)} \tag{55}$$

where PSRF values near 1 indicate that each of the H MCMC simulations is close to the stationary distribution. In our study we use as PSRF-based convergence diagnostic the fraction of edges $\mathcal{C}(\xi)$ whose PSRF is lower than a pre-defined threshold value ξ:

$$\mathcal{C}(\xi) = \frac{1}{N^2}\sum_{n=1}^{N}\sum_{j=1}^{N} Z_{PSRF<\xi}(PSRF(n,j)) \tag{56}$$

where $Z_{PSRF<\xi}(PSRF(n,j))$ is 1 if $PSRF(n,j) < \xi$ and 0 otherwise.

For the *Arabidopsis thaliana* data $2s = 1100k$ MCMC iterations were performed. From the last $s = 550k$ iterations we sampled $I_s = 550$ graphs by sampling every 1000th iteration. The focus of our study is on the convergence of the four MCMC sampling schemes for the cpBGe model. We perform $H = 10$ independent MCMC simulations and consider three different thresholds for ξ ($\xi = 1.02, 1.05, 1.1$).

1.9 Information coupling between nodes based on Bayesian clustering (Extended version of the main paper)

We instantiate the *class 2* model from Eq. (4) of the main paper by following Fearnhead (2006) and employing the point process prior for the changepoint locations defined in Eq. (5) of the main paper, i.e. the terms $\mathbf{K}$ and $\mathcal{K}_n$ in Eqn. (1-4) become obsolete. We extend the *class 2* model by introducing a cluster function $\mathcal{C}(.)$ that allocates the nodes $X_1, \dots, X_n$ to c $(1 \leq c \leq N)$ non-empty clusters, each characterized by its own changepoint vector $\mathbf{V}_i^{\mathcal{C}}$, $1 \leq i \leq c$:

$$\begin{aligned} P(\mathcal{G}, \mathbf{V}^{\mathcal{C}}, \mathcal{D}, \mathcal{C}) &= P(\mathcal{C}) P(\mathbf{V}^{\mathcal{C}}|\mathcal{C}) P(\mathcal{G}) P(\mathcal{D}|\mathcal{G}, \mathbf{V}^{\mathcal{C}}, \mathcal{C}) \\ &= P(\mathcal{C}) \left(\prod_{i=1}^{c} P(\mathbf{V}_i^{\mathcal{C}}|\mathcal{C}) \right) \prod_{n=1}^{N} P(\pi_n) \Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathbf{V}_{\mathcal{C}(n)}^{\mathcal{C}}]) \end{aligned} \tag{57}$$

with $\mathbf{V}^{\mathcal{C}} = (\mathbf{V}_1^{\mathcal{C}}, \dots, \mathbf{V}_c^{\mathcal{C}})$, where c is the number of non-empty node clusters induced by $\mathcal{C}$. We assume for $\mathrm{P}(\mathcal{C})$ a uniform distribution on all functions $\mathcal{C}$ that give c $(1 \leq c \leq N)$ clusters. The key idea behind the model of Eq. (57) is to encourage information sharing among nodes with respect to changepoint locations. Moreover, nodes that are in the same cluster i $(1 \leq i \leq c)$ share the same allocation vector $\mathbf{V}_i^{\mathcal{C}}$ and will be "penalized" only once[2]. 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 of the main paper.

For the third step, sampling from $P(\mathcal{C}|\mathbf{V}_i^{\mathcal{C}}, \mathcal{D}, \mathcal{G})$, we adopt an RJMCMC (Green, 1995) scheme based on cluster birth (b), death (d), and re-clustering (r) moves.[3] 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 ith cluster to a new cluster $c+1$. Denote by $\mathcal{C}^{\star}$ the new cluster formation thus obtained. For the ith 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 $P(\mathbf{V}_{c+1}^{\mathcal{C}^{\star}}|\mathcal{G}, \mathcal{D}, \mathcal{C}^{\star})$ and $P(\mathbf{V}_i^{\mathcal{C}^{\star}}|\mathcal{G}, \mathcal{D}, \mathcal{C}^{\star})$, defined in Eq. (59), with Fearnhead's dynamic programming scheme (Fearnhead, 2006), as discussed in Section 2.2 of the main paper. 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 dynamic programming (Fearnhead, 2006), where $\mathcal{C}^{\star}$ denotes the proposed cluster formation. In a re-clustering move we randomly choose two clusters i and j $(i \neq j)$ 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 Fearnhead's dynamic programming scheme (Fearnhead, 2006).
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 $(\mathcal{C}, \mathbf{V}^{\mathcal{C}}) \rightarrow (\mathcal{C}^{\star}, \mathbf{V}^{\mathcal{C}^{\star}})$, we thus get: $R_{(b)} = LR \times HR$

$$R_{(b)} = \frac{P(\mathcal{G}, \mathbf{V}^{\mathcal{C}^{\star}}, \mathcal{C}^{\star}, \mathcal{D})}{P(\mathcal{G}, \mathbf{V}^{\mathcal{C}}, \mathcal{C}, \mathcal{D})} \times \frac{c^{\dagger} c^{\ddagger} P(\mathbf{V}_i^{\mathcal{C}}|\mathcal{G}, \mathcal{D}, \mathcal{C})}{c^{\star} P(\mathbf{V}_{c+1}^{\mathcal{C}^{\star}}|\mathcal{G}, \mathcal{D}, \mathcal{C}^{\star}) P(\mathbf{V}_i^{\mathcal{C}^{\star}}|\mathcal{G}, \mathcal{D}, \mathcal{C}^{\star})} \tag{58}$$

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 ith 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 *regularized class 2* model the recursions of Fearnhead (2006) can be

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

[3] Each RJMCMC step was repeated 5 times.

employed as described in Section 1.7 to sample the j-th ($1 \leq j \leq c$) allocation vector $\mathbf{V}_j^{\mathcal{C}}$. We have:

$$P(\mathbf{V}_j^{\mathcal{C}}|\mathcal{G},\mathcal{D},\mathcal{C}) = \frac{q_j(\mathcal{D},\mathcal{C},\mathcal{G},\mathbf{V}_j^{\mathcal{C}})}{\sum_{\mathbf{V}_j^{\mathcal{C}^\star}} q_j(\mathcal{D},\mathcal{C}^\star,\mathcal{G},\mathbf{V}_j^{\mathcal{C}^\star})} \tag{59}$$

where

$$q_j(\mathcal{D},\mathcal{C},\mathcal{G},\mathbf{V}_j^{\mathcal{C}}) = P(\mathbf{V}_j^{\mathcal{C}}|\mathcal{C}) \prod_{n:\mathcal{C}(n)=j} \Psi^\dagger(\mathcal{D}_n^{\pi_n}[\mathbf{V}_j^{\mathcal{C}}]) \tag{60}$$

and the sum in Eq. (59) is over all valid allocation vectors $\mathbf{V}_j^{\mathcal{C}^\star}$ for the variables in the jth cluster of $\mathcal{C}^\star$.

It follows from Eqn. (57-58) that all factors except for the $(c+1)$th in the nominator and the ith ones cancel out in the likelihood ratio:

$$LR = \frac{q_i(\mathcal{D},\mathcal{C}^\star,\mathcal{G},\mathbf{V}_i^{\mathcal{C}^\star}) \cdot q_{c+1}(\mathcal{D},\mathcal{C}^\star,\mathcal{G},\mathbf{V}_{c+1}^{\mathcal{C}^\star})}{q_i(\mathcal{D},\mathcal{C},\mathcal{G},\mathbf{V}_i^{\mathcal{C}})} \tag{61}$$

Hence, $R_{(b)} = LR \times HR$ in Eq. (58) reduces to:

$$R_{(b)} = \frac{c^\dagger c^\ddagger}{c^\star} \frac{Q_i(\mathcal{D},\mathcal{C}^\star,\mathcal{G})Q_{c+1}(\mathcal{D},\mathcal{C}^\star,\mathcal{G})}{Q_i(\mathcal{D},\mathcal{C},\mathcal{G})} \tag{62}$$

where the terms

$$Q_j(\mathcal{D},\mathcal{C},\mathcal{G}) = \sum_{\mathbf{V}_j^{\mathcal{C}}} q_j(\mathcal{D},\mathcal{C},\mathcal{G},\mathbf{V}_j^{\mathcal{C}}) \tag{63}$$

can be computed efficiently with Fearnhead's dynamic programming scheme as described in Section 1.7. More precisely, as explained in the paragraph below Eq. (51) we have

$$Q_j(\mathcal{D},\mathcal{C},\mathcal{G}) = Q(2|j,\mathcal{C},\mathcal{G}) \tag{64}$$

where $Q(2|j,\mathcal{C},\mathcal{G})$ was specified in Section 1.7 (see Eq. (45)) and can be computed efficiently with Fearnhead's recursions.

The acceptance probabilities for cluster death and re-clustering moves can be derived as follows: For the cluster death move (d), $(\mathcal{C},\mathbf{V}^{\mathcal{C}}) \rightarrow (\mathcal{C}^\star,\mathbf{V}^{\mathcal{C}^\star})$, we assume that $c^\dagger$ is the number of clusters induced by $\mathcal{C}$ with one single node and that the ith cluster belongs to this group and is selected. Removing the single node from the ith cluster, such that the ith cluster is unoccupied and can be removed, and adding this node to the jth ($i \neq j$) cluster induced by $\mathcal{C}$ gives a new clustering $\mathcal{C}^\star$. We then get: $R_{(d)} = LR \times HR$

$$R_{(d)}{=}\frac{P(\mathcal{G},\mathbf{V}^{\mathcal{C}^\star},\mathcal{C}^\star,\mathcal{D})}{P(\mathcal{G},\mathbf{V}^{\mathcal{C}},\mathcal{C},\mathcal{D})} \times \frac{c^\star P(\mathbf{V}_j^{\mathcal{C}}|\mathcal{G},\mathcal{D},\mathcal{C})P(\mathbf{V}_i^{\mathcal{C}}|\mathcal{G},\mathcal{D},\mathcal{C})}{c^\dagger(c^\ddagger-1)P(\mathbf{V}_j^{\mathcal{C}^\star}|\mathcal{G},\mathcal{D},\mathcal{C}^\star)} \tag{65}$$

where $c^\dagger$ is the number of clusters induced by $\mathcal{C}$ that contain only one single node, $c^\ddagger$ is the number of clusters induced by $\mathcal{C}$, and $c^\star$ is the number of clusters induced by $\mathcal{C}^\star$ with at least two nodes.

It follows from Eqn. (57) and(65) that all factors except for the ith ones and the jth ones in the denominator cancel out in the likelihood ratio:

$$LR = \frac{q_i(\mathcal{D},\mathcal{C}^\star,\mathcal{G},\mathbf{V}_i^{\mathcal{C}^\star})}{q_i(\mathcal{D},\mathcal{C},\mathcal{G},\mathbf{V}_j^{\mathcal{C}})q_j(\mathcal{D},\mathcal{C},\mathcal{G},\mathbf{V}_j^{\mathcal{C}})} \tag{66}$$

Hence, Eq. (65) reduces to:

$$R_{(d)} = \frac{c^\star}{c^\dagger c^\ddagger} \frac{Q_i(\mathcal{D},\mathcal{C}^\star,\mathcal{G})}{Q_i(\mathcal{D},\mathcal{C},\mathcal{G})Q_j(\mathcal{D},\mathcal{C},\mathcal{G})} \tag{67}$$

where the $Q(.)$ terms defined in Eq. (63) can be computed efficiently as described in Section 1.7.

For the re-clustering move (r), $(\mathcal{C}, \mathbf{V}^{\mathcal{C}}) \to (\mathcal{C}^\star, \mathbf{V}^{\mathcal{C}^\star})$, we assume that $\mathcal{C}$ induces c clusters, and we further assume there are $c^\dagger$ clusters with at least two nodes and that the ith cluster belongs to this group and is selected. One of the nodes from the ith cluster is randomly selected and moved to the jth $(i \neq j)$ cluster of $\mathcal{C}$. Let n^i and n^j be the numbers of nodes in the ith and jth cluster of $\mathcal{C}$. We obtain: $R_{(r)} = LR \times HR$

$$R_{(r)}=\frac{P(\mathcal{G}, \mathbf{V}^{\mathcal{C}^\star}, \mathcal{C}^\star, \mathcal{D})}{P(\mathcal{G}, \mathbf{V}^{\mathcal{C}}, \mathcal{C}, \mathcal{D})} \times \frac{c^\dagger n^i (c-1) P(\mathbf{V}_i^{\mathcal{C}}|\mathcal{G}, \mathcal{D}, \mathcal{C}) P(\mathbf{V}_j^{\mathcal{C}}|\mathcal{G}, \mathcal{D}, \mathcal{C})}{c^\diamond (n^j+1)(c-1) P(\mathbf{V}_i^{\mathcal{C}^\star}|\mathcal{G}, \mathcal{D}, \mathcal{C}^\star) P(\mathbf{V}_j^{\mathcal{C}^\star}|\mathcal{G}, \mathcal{D}, \mathcal{C}^\star)} \tag{68}$$

where $c^\diamond$ is the number of clusters induced by $\mathcal{C}^\star$ that contain at least 2 nodes.

It follows from Eqn. (57) and (68) that all factors except for the ith and the jth ones cancel out in the likelihood ratio:

$$LR = \frac{q_i(\mathcal{D}, \mathcal{C}^\star, \mathcal{G}, \mathbf{V}_i^{\mathcal{C}^\star}) q_j(\mathcal{D}, \mathcal{C}^\star, \mathcal{G}, \mathbf{V}_j^{\mathcal{C}^\star})}{q_i(\mathcal{D}, \mathcal{C}, \mathcal{G}, \mathbf{V}_j^{\mathcal{C}}) q_j(\mathcal{D}, \mathcal{C}, \mathcal{G}, \mathbf{V}_j^{\mathcal{C}})} \tag{69}$$

Hence, Eq. (68) reduces to:

$$R_{(r)} = \frac{c^\dagger n^i}{c^\diamond (n^j+1)} \frac{Q_i(\mathcal{D}, \mathcal{C}^\star, \mathcal{G}) Q_j(\mathcal{D}, \mathcal{C}^\star, \mathcal{G})}{Q_i(\mathcal{D}, \mathcal{C}, \mathcal{G}) Q_j(\mathcal{D}, \mathcal{C}, \mathcal{G})} \tag{70}$$

where the $Q(.)$ terms can be computed effiviently as described in Section 1.7.

2 *Arabidopsis thaliana* gene expression time series

Plants assimilate carbon via photosynthesis during the day, but have a negative carbon balance at night. They buffer these daily alternations in their carbon budget by storing some of the assimilated carbon as starch in their leaves in the light, and utilising it as a carbon supply during the night. In order to synchronize these processes with the external 24 hour photo period, plants possess a circadian clock that can potentially provide predictive, temporal regulation of metabolic processes over the day/night cycle. The proper working of this circadian regulation is paramount to biomass production and growth, and considerable research efforts are therefore underway to elucidate its underlying molecular mechanism. In the present article, we aim to reconstruct the regulatory network of nine circadian genes in the model plant *Arabidopsis thaliana*.
We apply our method to microarray gene expression time series related to the study of circadian regulation in plants. *Arabidopsis thaliana* seedlings, grown under artificially controlled T_e-hour-light/T_e-hour-dark cycles, were transferred to constant light and harvested at 13 time points in τ-hour intervals. From these seedlings, RNA was extracted and assayed on Affymetrix GeneChip oligonucleotide arrays. The data were background-corrected and normalized according to standard procedures[4], using the GeneSpring© software (Agilent Technologies).

We combine four time series, which differed with respect to the pre-experiment entrainment condition and the time intervals: $T_e \in \{10h, 12h, 14h\}$, and $\tau \in \{2h, 4h\}$. The data, with detailed information about the experimental protocols, can be obtained from Edwards *et al.* (2006), Grzegorczyk *et al.* (2008), and Mockler *et al.* (2007). For an overview see Table 1. We focus our analysis on 9 circadian genes: LHY, TOC1, CCA1, ELF4, ELF3, GI, PRR9, PRR5, and PRR3, and we merge all four time series into one single data set. The objective is to employ the cpBGe model (Grzegorczyk and Husmeier, 2009), a (*class 2*) model with node-specific changepoints, to detect the different experimental phases. Since the gene expression values at the first time point of a

[4] We used RMA rather than GCRMA for reasons discussed in Lim *et al.* (2007).

	Segment 1	Segment 2	Segment 3	Segment 4
Source	Mockler et al.(2007)	Edwards et al. (2006)	Grzegorcyk et al. (2008)	Grzegorcyk et al. (2008)
Time points	12	13	13	13
Time interval	4h	4h	2h	2h
Pretreatment entrainment	12h:12h light:dark cycle	12h:12h light:dark cycle	10h:10h-dark light:dark cycle	14h:14h light:dark cycle
Measurements	Constant light	Constant light	Constant light	Constant light
Laboratory	Kay Lab	Millar Lab	Millar Lab	Millar Lab

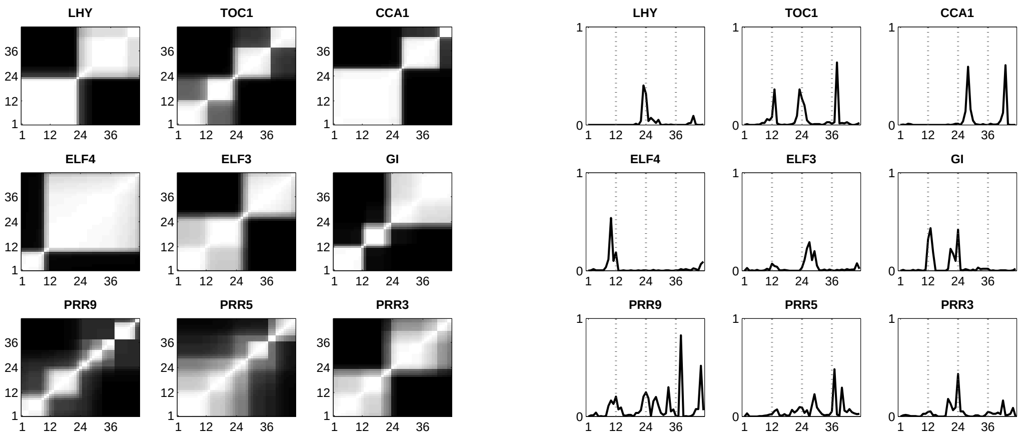
Table 1: **Gene expression time series segments for Arabidopsis.** The table contains an overview of the experimental conditions under which each of the gene expression experiments was carried out.

time series segment have no relation with the expression values at the last time point of the preceding segment, the corresponding boundary time points are appropriately removed from the data as described in Grzegorczyk and Husmeier (2009). This ensures that for all pairs of consecutive time points a proper conditional dependence relation determined by the nature of the regulatory cellular processes is given.

We elected to use these data as a test case for evaluating the efficiency of different sampling schemes for the cpBGe model (Grzegorczyk and Husmeier, 2009). Figure 1 of the main paper shows that the three Gibbs sampling schemes outperform the original RJMCMC sampler proposed in Grzegorczyk and Husmeier (2009) in terms of convergence and mixing. Since it appears that the GIBBS-NBIN algorithm performs slightly better than the other two Gibbs sampling schemes (see Figure 1 of the main paper), we report the results obtained with the GIBBS-NBIN algorithm:

Figure 2 shows the marginal posterior probability of the changepoint locations (right panel), and the posterior probability of the co-allocation of two time points to the same component (left panel). It is seen that, overall, the true segment boundaries tend to be detected. Different genes tend to be affected by the concatenation of the expression time series differently, though. For two genes (TOC1 and PRR9), all true changepoints are correctly predicted. Gene PRR9 shows various additional changepoints; this might indicate that it is affected by additional heterogeneities beyond the four experimental phases. Three of the genes (CCA1, ELF3, GI) show two changepoints, at the true locations (GI) or with a short time lag (CCA1). For genes LHY and ELF4 only one changepoint is predicted, at the location of the first or second concatenation point. A comparison of Table 1 with the locations of the peaks in Figure 2 suggests that gene CCA1 is mainly affected by a change of the entrainment condition, gene ELF4 is mainly affected by factors associated with the laboratory context, and genes ELF3 and PRR3 are mainly affected by a change of the sampling time interval (2 versus 4 hours). This deviation indicates that the genes are affected by the changing experimental conditions (entrainment, time interval) in different ways and that the node-specific changepoint model can be exploited as an exploratory tool for hypothesis generation.

Figure 3 shows the gene interaction network that is predicted when keeping all edges with marginal posterior probability above 0.5. There are two groups of genes. Empty circles in the figure represent morning genes (i.e. genes whose expression peaks in the morning), shaded circles represent evening genes (i.e. genes whose expression peaks in the evening). There are several directed edges pointing from the group of morning genes to the evening genes, mostly originating from gene CCA1. This result is consistent with the findings in McClung (2006), where the morning genes were found to activate the evening genes, with CCA1 and/or its partially redundant homologue LHY (Miwa *et al.*, 2007) being central regulators. E.g. Alabadi *et al.* (2001) found that CCA1 (and/or LHY) repress TOC1 and potentially other evening genes, and Kikis *et al.* (2005) report that CCA1 (and LHY) act negatively on ELF4 expression. Our reconstructed network also contains edges pointing



(a) Connectivity structures

(b) Transition probabilities

Figure 2: **Results on the Arabidopsis gene expression time series. Panel (a)**: Co-allocation matrices for the nine circadian genes. The axes represent time. The grey shading indicates the posterior probability of two time points being assigned to the same mixture component, ranging from 0 (black) to 1 (white). **Panel (b)**: Average posterior probability of a changepoint (vertical axis) at a specific transition time plotted against the transition time (horizontal axis) for the nine circadian genes. The vertical dotted lines indicate the boundaries of the time series segments, which are related to different experimental conditions (see Table 1).

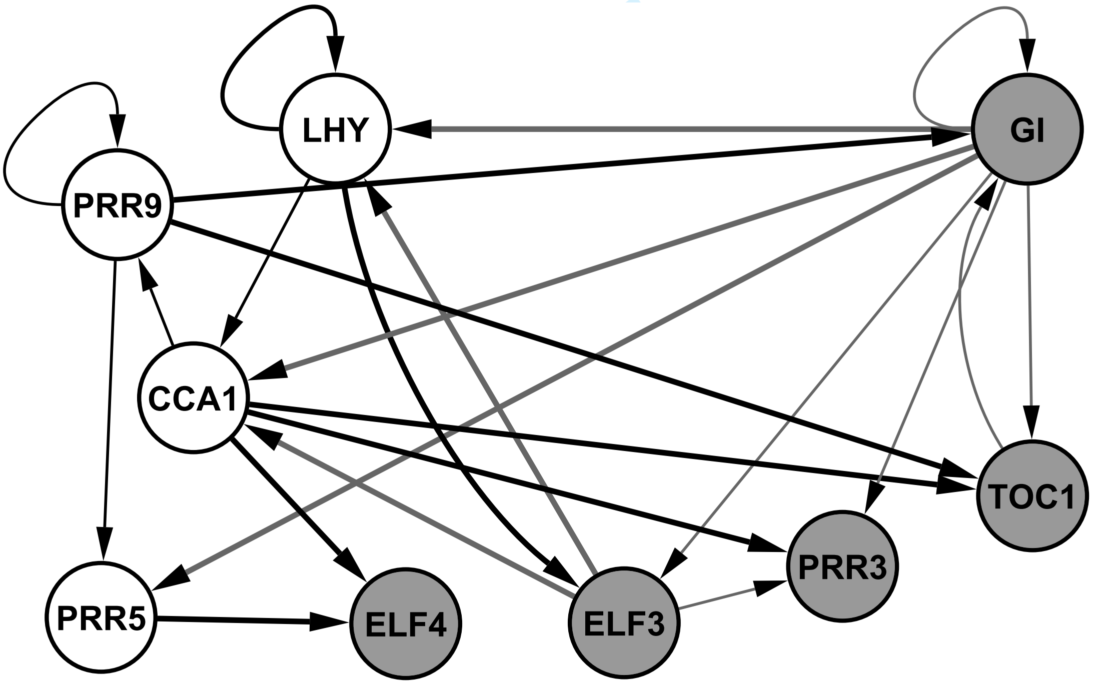


Figure 3: **Circadian gene regulatory network in Arabidopsis learnt from gene expression time series.** Predicted regulatory network of nine circadian genes in *Arabidopsis thaliana*. Empty circles represent morning genes. Shaded circles represent evening genes. Edges indicate predicted interactions with a marginal posterior probability greater than 0.5.

into the opposite direction, from the evening genes back to the morning genes. This finding is also consistent with McClung (2006), where the evening genes were found to inhibit the morning genes via a negative feedback loop. E.g. the edges $ELF3 \rightarrow CCA1$ and $ELF3 \rightarrow LHY$ in Figure 3 are consistent with the biological finding in Kikis *et al.* (2005) that ELF3 is necessary for light-induced CCA1 and LHY expression. Moreover, it is also known that GI and ELF3 play important roles in the circadian clock network and in that they are involved in the regulatory interactions between the morning genes LHY/CCA1 and the evening gene TOC1 (Miwa *et al.*, 2006). Within the group of evening genes, the reconstructed network contains a feedback loop between GI and TOC1, symbolically $GI \leftrightarrow TOC1$. This feedback loop has also been found in Locke *et al.* (2005) and is an improvement on our earlier work (Grzegorczyk and Husmeier, 2009), where only a unidirectional interaction $GI \rightarrow TOC1$ was extracted.
Hence while a proper evaluation of the reconstruction accuracy is currently unfeasible – like Robinson and Hartemink (2009) and many related studies, we lack a gold-standard owing to the unknown nature of the true interaction network – our study suggests that the essential features of the reconstructed network are biologically plausible and consistent with the literature.

References

Alabadi, D., Oyama, T., Yanovsky, M. J., Harmon, F. G., Mas, P. and Kay, S. A. (2001) Reciprocal regulation between TOC1 and LHY/CCA1 within the Arabidopsis circadian clock. *Science*, **293**, 880–883.

Brooks, S. and Gelman, A. (1998) General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphial Statistics*, **7**, 434–455.

Edwards, K. D., Anderson, P. E., Hall, A., Salathia, N. S., Locke, J. C., Lynn, J. R., Straume, M., Smith, J. Q. and Millar, A. J. (2006) Flowering locus C mediates natural variation in the high-temperature response of the Arabidopsis circadian clock. *The Plant Cell*, **18**, 639–650.

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

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

Geiger, D. and Heckerman, D. (1994) Learning Gaussian networks. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 235–243. Morgan Kaufmann, San Francisco, CA.

Giudici, P. and Castelo, R. (2003) Improving Markov chain Monte Carlo model search for data mining. *Machine Learning*, **50**, 127–158.

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

Grzegorczyk, M. and Husmeier, D. (2008) Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move. *Machine Learning*, **71**, 265–305.

Grzegorczyk, M. and Husmeier, D. (2009) Non-stationary continuous dynamic Bayesian networks. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I. and Culotta, A. (eds.), *Advances in Neural Information Processing Systems (NIPS)*, volume 22, pp. 682–690.

Grzegorczyk, M., Husmeier, D., Edwards, K., Ghazal, P. and Millar, A. (2008) Modelling non-stationary gene regulatory processes with a non-homogeneous Bayesian network and the allocation sampler. *Bioinformatics*, **24**, 2071–2078.

Kikis, E., Khanna, R. and Quail, P. (2005) ELF4 is a phytochrome-regulated component of a negative-feedback loop involving the central oscillator components CCA1 and LHY. *Plant J.*, **44**, 300–313.

Lim, W., Wang, K., Lefebvre, C. and Califano, A. (2007) Comparative analysis of microarray normalization procedures: effects on reverse engineering gene networks. *Bioinformatics*, **23**, i282–i288.

Locke, J., Southern, M., Kozma-Bognar, L., Hibberd, V., Brown, P., Turner, M. and Millar, A. (2005) Extension of a genetic network model by iterative experimentation and mathematical analysis. *Molecular Systems Biology*, **1**, (online).

Madigan, D. and York, J. (1995) Bayesian graphical models for discrete data. *International Statistical Review*, **63**, 215–232.

McClung, C. R. (2006) Plant circadian rhythms. *Plant Cell*, **18**, 792–803.

Miwa, K., Ito, S., Nakamichi, N., Mizoguchi, T., Niinuma, K., Yamashino, T. and Mizuno, T. (2007) Genetic linkages of the circadian clock-associated genes, TOC1, CCA1 and LHY, in the photoperiodic control of flowering time in Arabidopsis thaliana. *Plant and Cell Physiology*, **48**, 925–937.

Miwa, K., Serikawa, M., Suzuki, S., Kondo, T. and Oyama, T. (2006) Conserved expression profiles of circadian clock-related genes in two lemna species showing long-day and short-day photoperiodic flowering responses. *Plant and Cell Physiology*, **47**, 601–612.

Mockler, T., Michael, T., Priest, H., Shen, R., Sullivan, C., Givan, S., McEntee, C., Kay, S. and Chory, J. (2007) The diurnal project: Diurnal and circadian expression profiling, model-based pattern matching and promoter analysis. *Cold Spring Harbor Symposia on Quantitative Biology*, **72**, 353–363.

Robinson, J. W. and Hartemink, A. J. (2009) Non-stationary dynamic Bayesian networks. In Koller, D., Schuurmans, D., Bengio, Y. and Bottou, L. (eds.), *Advances in Neural Information Processing Systems (NIPS)*, volume 21, pp. 1369–1376. Morgan Kaufmann Publishers.

Rogers, S. and Girolami, M. (2005) A Bayesian regression approach to the inference of regulatory networks from gene expression data. *Bioinformatics*, **21**, 3131–3137.