Mach Learn (2011) 83: 355–419
DOI 10.1007/s10994-010-5230-7

Non-homogeneous dynamic Bayesian networks for continuous data

Marco Grzegorczyk · Dirk Husmeier

Received: 15 March 2010 / Revised: 23 November 2010 / Accepted: 24 November 2010 /
Published online: 27 February 2011

Abstract Classical dynamic Bayesian networks (DBNs) are based on the homogeneous Markov assumption and cannot deal with non-homogeneous temporal processes. Various approaches to relax the homogeneity assumption have recently been proposed. The present paper presents a combination of a Bayesian network with conditional probabilities in the linear Gaussian family, and a Bayesian multiple changepoint process, where the number and location of the changepoints are sampled from the posterior distribution with MCMC. Our work improves four aspects of an earlier conference paper: it contains a comprehensive and self-contained exposition of the methodology; it discusses the problem of spurious feedback loops in network reconstruction; it contains a comprehensive comparative evaluation of the network reconstruction accuracy on a set of synthetic and real-world benchmark problems, based on a novel discrete changepoint process; and it suggests new and improved MCMC schemes for sampling both the network structures and the changepoint configurations from the posterior distribution. The latter study compares RJMCMC, based on changepoint birth and death moves, with two dynamic programming schemes that were originally devised for Bayesian mixture models. We demonstrate the modifications that have to be made to allow for changing network structures, and the critical impact that the prior distribution on changepoint configurations has on the overall computational complexity.

Keywords Dynamic Bayesian networks · Non-homogeneity · Multiple changepoint process · Reversible jump Markov chain Monte Carlo (RJMCMC) · Dynamic programming · Receiver operator characteristics (ROC) curve · Precision-recall (PR) curve · Gene regulatory network · Circadian regulation · Arabidopsis

Editor: Kevin P. Murphy.

M. Grzegorczyk (✉)
Department of Statistics, TU Dortmund University, 44221 Dortmund, Germany
e-mail: grzegorczyk@statistik.tu-dortmund.de

D. Husmeier
Biomathematics and Statistics Scotland (BioSS), JCMB, The King's Buildings, Edinburgh EH9 3JZ, UK
e-mail: dirk@bioss.sari.ac.uk

1 Introduction

There has recently been considerable interest in structure learning of Bayesian networks. Examples from the topical field of systems biology are the reconstruction of transcriptional regulatory networks from gene expression data (Friedman et al. 2000), the inference of signal transduction pathways from protein concentrations (Sachs et al. 2005), and the identification of neural information flow operating in the brains of songbirds (Smith et al. 2006). In particular, dynamic Bayesian networks (DBNs) have been applied, as they allow feedback loops and recurrent regulatory structures to be modelled while avoiding the ambiguity about edge directions common to static Bayesian networks. The standard assumption underlying DBNs is that time-series have been generated from a homogeneous Markov process. However, regulatory interactions and signal transduction processes in the cell are usually adaptive and change in response to external stimuli. Likewise, neural information flow slowly adapts via Hebbian learning to make the processing of sensory information more efficient. The assumption of homogeneity is therefore too restrictive in many circumstances, and can potentially lead to erroneous conclusions.

Following earlier approaches aiming to relax the homogeneity assumption for undirected graphical models (Talih and Hengartner 2005 and Xuan and Murphy 2007), various recent research efforts have addressed the homogeneity assumption for DBNs. Robinson and Hartemink (2009) proposed a discrete non-homogeneous 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 non-homogeneous DBN, in which the parameters are allowed to vary over time, with a common network structure providing information sharing among the time series segments. Lèbre (2007, 2010) proposed an alternative continuous non-homogeneous 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 akin to a non-homogeneous DBN where inference is based on sparse L1-regularized regression (LASSO) of the interaction parameters, and a second L1 regularization term penalizes differences between networks associated with different segments.

Parameter estimation in Ahmed and Xing (2009) and Kolar et al. (2009) is based on penalized maximum likelihood for fixed regularization parameters. The optimization of the latter is based on BIC or cross-validation, and a bootstrapping scheme is required to estimate inference uncertainty. In the present paper, we follow Robinson and Hartemink (2009), Grzegorczyk and Husmeier (2009) and Lèbre (2007) to infer the network structure, the interaction parameters, as well as the number and location of changepoints in a Bayesian framework by sampling them from the posterior distribution with a Markov chain Monte Carlo (MCMC) scheme.

Our work is an expansion of our earlier model (Grzegorczyk and Husmeier 2009), which was introduced to address two shortcomings of the alternative non-homogeneous DBNs of Robinson and Hartemink (2009) and Lèbre (2007). As opposed to Robinson and Hartemink (2009), the model in Grzegorczyk and Husmeier (2009) is continuous and thus avoids the information loss inherent in data discretization. A shortcoming of Lèbre (2007, 2010) is potential model over-flexibility: different network structures are associated with different time series segments, which for short time series will inevitably lead to over-fitting or inflated inference uncertainty. The approach in Grzegorczyk and Husmeier (2009) introduces information sharing among different time series segments via constraints on the network structure: the model is non-homogeneous with respect to the parameters, while the network structure is the same for all segments. While for certain scenarios, like morphogenesis, this model is too restrictive, we have argued that for most cellular processes on a shorter time scale it is not the structure but rather the strength of the regulatory interactions that changes with time. Put differently, and to paraphrase and recite Robinson and Hartemink (2009): it is not the road system (the network structure) that changes between off-peak and rush hour, but the intensity of the traffic flow (the strength of the interactions). In the same vein, it is not the ability of a transcription factor to potentially bind to the promoter of a gene and thereby initiate transcription (the interaction structure), but the extent to which this happens (the interaction strength).

The objective of the present paper is to expand and improve our earlier paper (Grzegorczyk and Husmeier 2009) in four important aspects. Firstly, due to a strict page limit, the presentation of the methodology in Grzegorczyk and Husmeier (2009) is very terse, and we here offer a more comprehensive and self-contained exposition. Secondly, we discuss the problem of spurious feedback loops. Feedback loops are essential to the regulation and stable control of complex biological systems, and the application of dynamic as opposed to static Bayesian networks has been motivated by the fact that feedback loops can, in principle, be learnt. In the present work, we demonstrate that a linear homogeneous DBN is susceptible to reconstructing spurious feedback loops, and we investigate how far this susceptibility is overcome when using the proposed non-homogeneous DBN. Thirdly, we have replaced the continuous changepoint process of Grzegorczyk and Husmeier (2009) by a simpler discrete changepoint process, and we have rerun all the simulations to ascertain that this modification does not noticeably affect the results. Fourthly and most importantly, we have invested considerable efforts into improving and assessing mixing and convergence of the MCMC sampling scheme. Like Robinson and Hartemink (2009) and Lèbre (2007, 2010), our earlier work pursued inference with reversible jump MCMC (Green 1995), based on birth and death moves for individual changepoints. In the present paper, we explore the application of the dynamic programming scheme of Fearnhead (2006), with which changepoint configurations are sampled from the proper conditional distribution within a Gibbs sampling scheme.[1] We compare two alternative approaches, based on different prior distributions for the changepoint processes, and we critically assess mixing, convergence and the computational complexity of these schemes.

2 Methodology

2.1 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 $\mathcal{G}$. Let $t = 1, \dots, m$ represent time points. 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 $X_n(t)$ of X_n at time point t is conditionally dependent on the realization $X_j(t-1)$ of X_j at time point $t-1$. See Fig. 1 for an example of a DBN consisting of two nodes X_1 and X_2. 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 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 $\mathcal{D}_{n,t}$

[1] Note that the dynamic programming scheme of Fearnhead (2006) has already been applied in the context of undirected graphical models (Xuan and Murphy 2007).

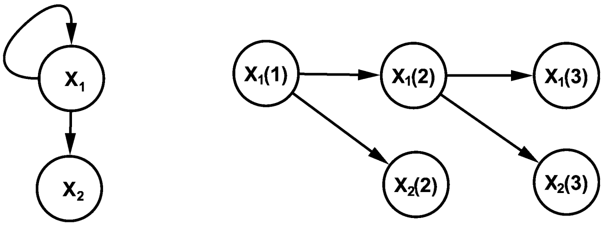


Fig. 1 State space graph and corresponding dynamic Bayesian network of order $\tau = 1$. The *left panel* shows a recurrent state space graph containing two nodes. Node X_1 has a recurrent feedback loop and acts as a regulator of node X_2. The *right panel* shows the same graph unfolded in time

and $\mathcal{D}_{\pi_n,t}$ are the t-th realizations $X_n(t)$ and $\pi_n(t)$ of X_n and π_n, respectively. 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 (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

$$P(\mathcal{D}|\mathcal{G}) = \int P(\mathcal{D}|\mathcal{G},\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathcal{G})d\boldsymbol{\theta} \ = \prod_{n=1}^{N} \Psi(\mathcal{D}_n^{\pi_n}) \tag{2}$$

$$\Psi(\mathcal{D}_n^{\pi_n}) = \int \prod_{t=2}^{m} P\big(X_n(t) = \mathcal{D}_{n,t}|\pi_n(t-1) = \mathcal{D}_{\pi_n,t-1}, \boldsymbol{\theta}_n\big) P(\boldsymbol{\theta}_n|\pi_n)d\boldsymbol{\theta}_n \tag{3}$$

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 the *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 (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 (3) has a closed form solution, given by (24) in Geiger and Heckerman (1994). The resulting expression is called the (local) BGe score. We note that the score equivalence aspect of the BGe model is not required for DBNs of order $\tau = 1$, because edge reversals are not permissible when all conditional dependencies $X_i \to X_j$ are modelled with a time lag: $X_i(t-1) \to X_j(t)$. Formulating our changepoint model in terms of the BGe score has an advantage with regard to potential generalizations in future work. The BGe score can also be employed (i) in DBNs with additional intra-slice interactions ($\tau = 0$), such as $X_i(t) \to X_j(t)$, and (ii) when aiming to adapt the proposed framework to nonlinear static Bayesian networks along the lines of Ko et al. (2007).

2.2 The non-homogeneous dynamic changepoint BGe model (cpBGe)

To obtain a non-homogeneous DBN, we generalize (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 k-th 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 (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 (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 (6) are given by:

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

Equation (7) is similar to (3), and can be interpreted as a local BGe score restricted to the data subset $\mathcal{D}_n^{\pi_n}[k,\mathbf{V}_n] := \{(\mathcal{D}_{n,t},\mathcal{D}_{\pi_n,t-1}) : \mathbf{V}_n(t)=k, 2\le t\le m\}$. The product $\Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathcal{K}_n,\mathbf{V}_n])$ in (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 (7) has a closed-form solution: it is given by (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 k-th mixture component (or k-th 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 $P(\mathcal{G}) = \prod_{n=1}^{N} P(\pi_n)$, 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 $|\pi_n| \leq 3$ to ensure sparsity of the inferred graph structures.[2] 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 ($n = 1, \ldots, N$) so that the joint probability distribution in (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 $\mathcal{D}_n^{1:N} := \{(\mathcal{D}_{n,t}, \mathcal{D}_{1,t-1}, \ldots, \mathcal{D}_{N,t-1}) : 2 \leq t \leq m\}$ contains the last $m-1$ observations $\mathcal{D}_{n,2}, \ldots, \mathcal{D}_{n,m}$ of X_n and the first $m-1$ observations $\mathcal{D}_{j,1}, \ldots, \mathcal{D}_{j,m-1}$ of all potential parent nodes X_j ($j = 1, \ldots, N$) of X_n. We note that each factor $P(\pi_n, \mathbf{V}_n, \mathcal{K}_n|\mathcal{D}_n^{1:N})$ in (10) can be inferred independently.

As prior probability distributions on the node-specific numbers of mixture components $\mathcal{K}_n$, $P(\mathcal{K}_n)$, we take i.i.d. 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). As in our earlier work (Grzegorczyk and Husmeier 2009), the prior distribution on the node-specific latent variable vectors, $P(\mathbf{V}_n|\mathcal{K}_n)$, is implicitly defined via a changepoint process. Different from our earlier work we employ the discrete counterpart of the prior of Green (1995) and identify $\mathcal{K}_n$ components with $\mathcal{K}_n - 1$ changepoints $\mathbf{b}_n = (b_{n,1}, \ldots, b_{n,\mathcal{K}_n-1})$ on the *discrete* set $\{2, \ldots, m-1\}$. With this modification it is possible to employ a dynamic programming scheme for sampling changepoints from the posterior distribution, as discussed in more detail at the end of this section and in Sects. 2.7.1 and 2.7.2. For node X_n the observation at time point t is assigned to the k-th component $\mathbf{V}_n(t) = k$ if and only if $b_{n,k-1} < t \leq b_{n,k}$, where $b_{n,k}$ is the k-th changepoint implied by $\mathbf{V}_n$, and $b_{n,0} = 1$ and $b_{n,\mathcal{K}_n} = m$ are two pseudo changepoints, which have been introduced for notational convenience. Different from the continuous changepoint process, the discrete version avoids empty components and gives a one-to-one mapping between allocation vectors and changepoints: For $t = 2, \ldots, m$ and $k = 1, \ldots, \mathcal{K}_n$: $b_{n,k-1} < t \leq b_{n,k} \Leftrightarrow \mathbf{V}_n(t) = k$. To make that more specific, we henceforth use the notation $\mathbf{b}_{\mathbf{V}_n} = (b_{\mathbf{V}_n,1}, \ldots, b_{\mathbf{V}_n,\mathcal{K}_n-1})$ for the changepoint vector implied by $\mathbf{V}_n$. Following Green (1995) and our own earlier work (Grzegorczyk and Husmeier 2009) we assume that the changepoints are distributed as the even-numbered order statistics of $\mathcal{L} := 2(\mathcal{K}_n - 1) + 1$ points $u_1, \ldots, u_{\mathcal{L}}$ uniformly and independently distributed on the set

[2] Given the homogeneous DBN model from Sect. 2.1 and a "sufficient" fan-in restriction, inference by full model-averaging is often more efficient than MCMC sampling of graph structures. We note that full model-averaging is generally *unfeasible* for the non-homogeneous cpBGe model considered here. We return to this point in Sects. 2.4 and 2.5.

$\{2,\dots,m-1\}$. Different from a uniform distribution, this distribution encourages *a priori* an equal spacing between the changepoints. That is, we want to discourage mixture components (i.e. segments) that contain only a few observations. 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 different from the continuous changepoint model, the even-numbered order statistics prior on the discrete changepoints 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 Sects. 2.5 and 2.8 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 even for the continuous changepoint model. For the more sophisticated 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. Since the continuous changepoint model counterpart of (11) cannot be computed in closed form, we decided to modify our original model (Grzegorczyk and Husmeier 2009) correspondingly.

2.3 MCMC based model inference

2.3.1 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 (10). Our MH samplers combine the structure MCMC algorithm for Bayesian networks (Giudici and Castelo 2003 and 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 Sect. 2.2. 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 $\mathbf{V}^{i+1} = \mathbf{V}^i$ 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 (6). The graph is left unchanged $\mathcal{G}^{i+1} := \mathcal{G}^i$ 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 $(\mathcal{K}_n^i, \mathbf{V}_n^i) \to (\mathcal{K}_n^{i+1}, \mathbf{V}_n^{i+1})$ 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 (6)–(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 $\{b_{\mathbf{V}_n^i,j-1}+2, \ldots, b_{\mathbf{V}_n^i,j+1}-2\}$ where $b_{V_n^i,0} = 1$ and $b_{\mathbf{V}_n^i,\mathcal{K}_n} = m$. The inverse proposal probability ratio for reallocation moves (r) is equal to 1 ($B_{(r)} = 1$) and the prior probabilities $P(\mathcal{K}_n^{i+1}) = P(\mathcal{K}_n^i)$ in the prior probability ratio $A_{(r)}$ cancel out. From (11) it can be seen that the remaining prior probability ratio $P(\mathbf{V}_n^{i+1}|\mathcal{K}_n^{i+1})/P(\mathbf{V}_n^i|\mathcal{K}_n^i)$ 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:

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

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 $(\mathcal{K}_n^{i+1}, \mathbf{V}_n^{i+1})$. For this birth move ($b$) the prior probability ratio $A_{(b)}$ can be computed with (11):

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

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:

$$\begin{aligned} A_{(b)}B_{(b)} = {} & \frac{(2\mathcal{K}_n^i+1)(2\mathcal{K}_n^i)}{(m-2\mathcal{K}_n^i-1)(m-2\mathcal{K}_n^i-2)} \frac{(b_{\mathbf{V}_n^i,j+1} - b^\dagger - 1)(b^\dagger - b_{\mathbf{V}_n^i,j} - 1)}{(b_{\mathbf{V}_n^i,j+1} - b_{\mathbf{V}_n^i,j} - 1)} \\ & \times \frac{|B^\dagger(\mathbf{V}_n^i)|}{\mathcal{K}_n^i} \end{aligned} \tag{17}$$

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 $b^\dagger$. For the death move (d) we obtain for the product of the prior probability ratio $A_{(d)}$ and the inverse proposal probability ratio $B_{(d)}$:

$$\begin{aligned} A_{(d)}B_{(d)} = {} & \frac{(m-2\mathcal{K}_n^i-3)(m-2\mathcal{K}_n^i-4)}{(2\mathcal{K}_n^i-1)(2\mathcal{K}_n^i-2)} \frac{(b_{\mathbf{V}_n^i,j+1} - b_{\mathbf{V}_n^i,j-1} - 1)}{(b_{\mathbf{V}_n^i,j+1} - b^\dagger - 1)(b^\dagger - b_{\mathbf{V}_n^i,j-1} - 1)} \\ & \times \frac{\mathcal{K}_n^i - 1}{|B^\dagger(\mathbf{V}_n^{i+1})|} \end{aligned} \tag{18}$$

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.

2.4 Problems with mixing and convergence of the structure MCMC sampler

For dynamic Bayesian networks (DBNs) the standard structure MCMC sampler for Bayesian networks is usually based on two single-edge operations, namely edge additions and edge deletions, as described in Sect. 2.3.1.[3] Edge reversal operations are often excluded since the time lag of interactions renders interactions, such as $X_i(t-1) \to X_j(t)$ and $X_j(t-1) \to X_i(t)$ independent; especially these two oppositely oriented edges do not exclude each other. That is, in principle, the parent node set π_n of each variable X_n can be inferred independently if restricting on edge additions and edge deletions, while edge reversals can effectively be seen as a combination of two independent edge operations that change two parent node sets π_i and π_j simultaneously. This generates unnecessary dependencies in the inference of π_i and π_j and would render a parallel computing approach impossible.

Several studies, e.g. Friedman and Koller (2003) or Grzegorczyk and Husmeier (2008), have shown that the proposal scheme of the structure MCMC sampler leads to poor convergence and mixing, as simulations tend to get stuck in local optima. The proposed cpBGe model infers graphs with the structure MCMC sampler and thus it is likely that the graph inference is suboptimal in terms of convergence. In the following Sects. 2.5–2.8, we will therefore look into a methodologically consistent way of improving the convergence and mixing of the MCMC chains by designing improved proposal mechanisms that exploit the intrinsic modularity of the system.

2.5 Sampling parent node sets from the "Boltzmann" distribution

The Metropolis-Hastings (MH) sampler presented in Sect. 2.3.1 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 (3) and the sum is over all valid parent node sets π_n subject to a fan-in restriction $\mathcal{F}$. If one draws a parallel of the negative logarithm of the score $\Psi(D_n^{\pi_n})$ to a configurational energy of a fictitious physical system, then the distribution in (19) is the "Boltzmann" distribution—a standard distribution in statistical physics—and we hence use the same name to refer to it. Equation (19) is similar to (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. (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

[3] Note that the structure MCMC algorithm for static Bayesian networks (Giudici and Castelo 2003 and Madigan and York 1995) is usually based on *three* types of single edge operations, namely: edge additions, edge deletions, and edge reversals.

allocation vector $\mathbf{V}_n$:

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

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 (6)–(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-computation is required.

In Sects. 2.7.1 and 2.7.2 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: $P(\mathbf{V}_n, \mathcal{K}_n|\pi_n, \mathcal{D}_n^{\pi_n})$. Employing this dynamic programming scheme allows for large steps in the configuration space of the allocation vector so that the stepwise re-computation of the "Boltzmann" distributions becomes computationally more efficient. We will show that 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 sampling scheme. See Sect. 2.8 for details.

2.6 New variant of the structure MCMC sampler: The FLIP move

As an alternative to the sampling scheme discussed in Sect. 2.5 we will also try to improve the convergence of the structure MCMC sampler by a new single edge operation. The *parent-node FLIP move* exchanges one single parent node $X_i \in \pi_n$ from the current parent node set π_n for another novel node $X_j \notin \pi_n$. Since the parent-node flip is effectively the simultaneous performance of an edge deletion $X_i \rightarrow X_n$ and an edge addition $X_j \rightarrow X_n$ it is similar to the edge reversal move; but it is different from the edge reversal in that the flip operation affects only one single parent set π_n. From that perspective the flip move can be seen as the dynamic Bayesian network alternative to single edge reversals in static Bayesian networks. Incorporating the flip operator move into the structure MCMC sampler improves the flexibility of the proposal scheme, as it allows for moves that could otherwise only be accomplished by two successive moves. To demonstrate that this can be advantageous we consider a simple example: Let there be three potential parent nodes A, B, and C for a node X_n, and only two parent sets $\pi_{n,1} = \{A, B\}$ and $\pi_{n,2} = \{A, C\}$ with a high local score while all others parent node sets have low local scores. If we restrict on edge additions and edge deletions, then, after having reached $\pi_{n,1}$, the structure MCMC sampler can propose only three neighbouring parent node sets $\{A\}$, $\{B\}$ and $\{A, B, C\}$. The acceptance probability for these three moves will be low. But for every move between $\pi_{n,1}$ and $\pi_{n,2}$ one of these intermediate parent node sets has to be accepted first. Consequently the structure MCMC sampler will not mix well between the two optimal parent node sets $\pi_{n,1}$ and $\pi_{n,2}$. And when moves to the three intermediate parent node sets have a low acceptance probability, the simulation is susceptible to getting stuck either in $\pi_{n,1}$ or $\pi_{n,2}$. With the novel FLIP operator the problem can be avoided, as $\pi_{n,2}$ can be reached from $\pi_{n,1}$ in one single step and vice-versa so that substantially better mixing can be expected.

2.7 Sampling changepoints by dynamic programming

In (20) we have introduced the distribution $P(\pi_n^\star|\mathcal{K}_n, \mathbf{V}_n, \mathcal{D}_n^{1:N})$. This is half a Gibbs step: given the changepoints, we can sample the parent configuration from the proper conditional distribution. However, as sampling from (20) is computationally expensive, owing to the normalization, the application of this scheme makes only sense if the changepoints can be sampled from the complementary conditional distribution $P(\mathcal{K}_n, \mathbf{V}_n|\pi_n, \mathcal{D}_n^{1:N}) = P(\mathcal{K}_n, \mathbf{V}_n|\pi_n, \mathcal{D}_n^{\pi_n})$ so as to complete the Gibbs step. In the present section, we will discuss how this can be accomplished with a dynamic programming scheme. The method is based on Fearnhead (2006), which was developed for Bayesian mixture models, and we adapt this framework to non-homogeneous dynamic Bayesian networks. As discussed in Fearnhead (2006), one can consider two classes of prior distribution for the changepoint process. The first, which we have considered so far, involves a prior on the number of changepoints, and then a conditional prior on their positions. An alternative prior is based on modelling the changepoint process by a point process, which indirectly specifies a joint prior on the number and positions of the changepoints. As it turns out, the dynamic programming scheme becomes conceptually and computationally simpler when adopting the second prior, and we will describe it first, in Sect. 2.7.1. We will then, in Sect. 2.7.2, present a dynamic programming scheme for the original prior. Several mathematical symbols for referring to particular subsets of the data have been defined, and three more symbols will be required in the following two subsections. For clarity, a summary of all those symbols is given in Table 1.

2.7.1 *Dynamic programming for a point process prior*

As mentioned above, we will slightly modify the prior distribution for $(\mathcal{K}_n, \mathbf{V}_n)$. Instead of modelling $P(\mathcal{K}_n)$ explicitly, and 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 $b_{n,1}, \ldots, b_{n,\mathcal{K}_n-1}$

Table 1 Overview of various symbols referring to subsets of the data $\mathcal{D}$

Symbol	Definition	First appearance
$\mathcal{D}$	the complete data set	2.1, above (1)
$\mathcal{D}_{n,t}$	realization of node X_n at time point t	2.1, above (1)
$\mathcal{D}_{\pi_n,t}$	realizations of the parent nodes of X_n at time point t	2.1, above (1)
$\mathcal{D}_n^{\pi_n}$	$\{(\mathcal{D}_{n,t}, \mathcal{D}_{\pi_n,t-1}) : 2 \le t \le m\}$	2.1, below (3)
$\mathcal{D}_n^{\pi_n}[k, \mathbf{V}_n]$	$\{(\mathcal{D}_{n,t}, \mathcal{D}_{\pi_n,t-1}) : \mathbf{V}_n(t) = k, 2 \le t \le m\}$	2.2, below (7)
$\mathcal{D}_n^{1:N}$	$\{(\mathcal{D}_{n,t}, \mathcal{D}_{1,t-1}, \ldots, \mathcal{D}_{N,t-1}) : 2 \le t \le m\}$	2.2, below (10)
$\mathcal{D}_{n,s:t}$	$\{\mathcal{D}_{n,i} : s \le i \le t\}$	2.7.1, above (25)
$\mathcal{D}_{\pi_n,s:t}$	$\{\mathcal{D}_{\pi_n,i} : s \le i \le t\}$	2.7.1, above (25)
$\mathcal{D}_n^{\pi_n}[s:t]$	$\{(\mathcal{D}_{n,i}, \mathcal{D}_{\pi_n,i-1}) : s \le i \le t\}$	2.7.1, above (25)

The symbols $\mathcal{D}$, $\mathcal{D}_{n,t}$, $\mathcal{D}_{\pi_n,t}$, $\mathcal{D}_n^{\pi_n}$, and $\mathcal{D}_n^{1:N}$ have been introduced in Sects. 2.1 and 2.2. The symbols $\mathcal{D}_{n,s:t}$, $\mathcal{D}_{\pi_n,s:t}$, and $\mathcal{D}_n^{\pi_n}[s:t]$ will be introduced and used in Sects. 2.7.1 and 2.7.2

is:

$$P(b_{n,1},\ldots,b_{n,\mathcal{K}_n-1}) = g_0(b_{n,1})\left(\prod_{j=2}^{\mathcal{K}_n-1} g(b_{n,j}-b_{n,j-1})\right)(1-G(b_{n,\mathcal{K}_n}-b_{n,\mathcal{K}_n-1})) \tag{21}$$

where $b_{n,0}=1$ and $b_{n,\mathcal{K}_n}=m$ are again pseudo changepoints, $g_0(.)$ is the prior distribution of the first changepoint $b_{n,1}$, and

$$G(t)=\sum_{s=1}^{t} g(t); \qquad G_0(t)=\sum_{s=1}^{t} g_0(t) \tag{22}$$

are the cumulative distribution functions corresponding to $g(.)$ and $g_0(.)$. For $g(.)$ the probability mass function of the negative binomial distribution[4] NBIN(p,k) with parameters p and k can be used:

$$g(t)=\binom{t-1}{k-1}p^k(1-p)^{t-k} \tag{23}$$

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:

$$g_0(b_{n,1})=\frac{1}{k}\sum_{i=1}^{k}\binom{(b_{n,1}-1)-1}{i-1}p^i(1-p)^{(b_{n,1}-1)-i} \tag{24}$$

Let $\mathcal{D}_n^{\pi_n}$ denote the set of observations $\{(\mathcal{D}_{n,i},\mathcal{D}_{\pi_n,i-1}):2\le i\le m\}$ pertaining to node X_n and its parent node set π_n, and accordingly, let $\mathcal{D}_n^{\pi_n}[s:t]$ denote the sub-segment $\{(\mathcal{D}_{n,i},\mathcal{D}_{\pi_n,i-1}):s\le i\le t\}$ of adjacent observations. We also set $\mathcal{D}_{n,s:t}=\{\mathcal{D}_{n,i}:s\le i\le t\}$ and $\mathcal{D}_{\pi_n,s:t}=\{\mathcal{D}_{\pi_n,i}:s\le i\le t\}$. For each node X_n we define $Q(t|n,\pi_n)$ as the probability of the observations for node X_n, $\mathcal{D}_{n,t:m}$, given the parental observations $\mathcal{D}_{\pi_n,(t-1):(m-1)}$ of π_n and a changepoint $b^\dagger$ at time point $t-1$ ($t=2,\ldots,m$):

$$Q(t|n,\pi_n)=P(\mathcal{D}_{n,t:m}|\mathcal{D}_{\pi_n,(t-1):(m-1)},b^\dagger=t-1) \tag{25}$$

$Q(m|n,\pi_n)$ is then equal to $\Psi(\mathcal{D}_n^{\pi_n}[m:m])$, and for $t=3,\ldots,m-1$ a recursion can be used:

$$\begin{aligned}Q(t|n,\pi_n)=&\left(\sum_{s=t}^{m-1}\Psi(\mathcal{D}_n^{\pi_n}[t:s])Q(s+1|n,\pi_n)g(s+1-t)\right)\\&+\Psi(\mathcal{D}_n^{\pi_n}[t:m])(1-G(m-t))\end{aligned} \tag{26}$$

and

$$\begin{aligned}Q(2|n,\pi_n)=&\left(\sum_{s=2}^{m-1}\Psi(\mathcal{D}_n^{\pi_n}[2:s])Q(s+1|n,\pi_n)g_0(s-1)\right)\\&+\Psi(\mathcal{D}_n^{\pi_n}[2:m])(1-G_0(m-2))\end{aligned} \tag{27}$$

[4] Note that the negative binomial distribution can be seen as a discrete version of the Gamma distribution.

where $G_0(t)=\sum_{s=1}^{t} g_0(s)$. For the proof, note that

$$P(\mathcal{D}_{n,t:m}|\mathcal{D}_{\pi_n,(t-1):(m-1)}, b^\dagger = t-1) \tag{28}$$

$$= \sum_{s=t}^{m-1} P(\mathcal{D}_{n,t:m}, \text{next changepoint at } b^\ddagger = s|\mathcal{D}_{\pi_n,(t-1):(m-1)}, b^\dagger = t-1)$$

$$+ P(\mathcal{D}_{n,t:m}, \text{no further changepoint } b^\ddagger > b^\dagger|\mathcal{D}_{\pi_n,(t-1):(m-1)}, b^\dagger = t-1) \tag{29}$$

We proceed by decomposing the first term. Note that for a new changepoint $b^\ddagger = s > b^\dagger = t-1$ that is not separated from $b^\dagger$ by any other changepoint we have:

$$P(\mathcal{D}_{n,t:m}, \text{next changepoint at } b^\ddagger = s|\mathcal{D}_{\pi_n,(t-1):(m-1)}, b^\dagger = t-1) \tag{30}$$

$$= P(\mathcal{D}_{n,t:m}|\text{next changepoint at } b^\ddagger = s, \mathcal{D}_{\pi_n,(t-1):(m-1)}, b^\dagger = t-1)$$

$$\times P(\text{next changepoint at } b^\ddagger = s|\mathcal{D}_{\pi_n,(t-1):(m-1)}, b^\dagger = t-1) \tag{31}$$

$$= P(\mathcal{D}_{n,t:s}, \mathcal{D}_{n,(s+1):m}|\text{next changepoint at } b^\ddagger = s, \mathcal{D}_{\pi_n,(t-1):(m-1)}, b^\dagger = t-1)$$

$$\times P(\text{next changepoint at } b^\ddagger = s|b^\dagger = t-1) \tag{32}$$

$$= P(\mathcal{D}_{n,t:s}|\mathcal{D}_{\pi_n,(t-1):(s-1)}, b^\ddagger = s, b^\dagger = t-1) P(\mathcal{D}_{n,s+1:m}|\mathcal{D}_{\pi_n,s:(m-1)}, b^\ddagger = s, b^\dagger = t-1)$$

$$\times P(\text{next changepoint at } b^\ddagger = s|b^\dagger = t-1) \tag{33}$$

There are two subsequent changepoints, at positions $b^\ddagger = s$ and $b^\dagger = t-1$. The segment $\mathcal{D}_{n,t:s}$ is thus homogeneous, i.e. not divided by any further changepoints, and we have: $P(\mathcal{D}_{n,t:s}|\mathcal{D}_{\pi_n,(t-1):(s-1)}, b^\ddagger = s, b^\dagger = t-1) = P(\mathcal{D}_{n,t:s}|\mathcal{D}_{\pi_n,(t-1):(s-1)}, \text{not divided by any changepoint}) = \Psi(\mathcal{D}_n^{\pi_n}[t:s])$. The expression $P(\mathcal{D}_{n,(s+1):m}|\mathcal{D}_{\pi_n,s:(m-1)}, b^\ddagger = s, b^\dagger = t-1)$ was defined in (25) and is given by $Q(s+1|n,\pi_n)$. The probability $P(\text{next changepoint at } b^\ddagger = s|b^\dagger = t-1)$ is given by the point process prior, which was defined in (23)–(24). Putting these terms together, we get:

$$P(\mathcal{D}_{n,t:m}, \text{next changepoint at } b^\ddagger = s|\mathcal{D}_{\pi_n,(t-1):(m-1)}, b^\dagger = t-1)$$

$$= \Psi(\mathcal{D}_n^{\pi_n}[t:s]) Q(s+1|n,\pi_n) g(s+1-t) \tag{34}$$

for $t > 2$. If $t = 2$, then $b^\ddagger = s$ is the first changepoint, and we have:

$$P(\mathcal{D}_{n,t:m}, \text{next changepoint at } b^\ddagger = s|\mathcal{D}_{\pi_n,(t-1):(m-1)}, b^\dagger = t-1)$$

$$= \Psi(\mathcal{D}_n^{\pi_n}[2:s]) Q(s+1|n,\pi_n) g_0(s-1) \tag{35}$$

In the same vein, we decompose the second term in (29):

$$P(\mathcal{D}_{n,t:m}, \text{no further changepoint } b^\ddagger > b^\dagger|\mathcal{D}_{\pi_n,(t-1):(m-1)}, b^\dagger = t-1)$$

$$= P(\mathcal{D}_{n,t:m}|\text{no further changepoint } b^\ddagger > b^\dagger, \mathcal{D}_{\pi_n,(t-1):(m-1)}, b^\dagger = t-1)$$

$$+ P(\text{no further changepoint } b^\ddagger > b^\dagger|b^\dagger = t-1) \tag{36}$$

As there are no further changepoints $b^\ddagger > b^\dagger = t-1$, $\mathcal{D}_{n,t:m}$ is a homogeneous segment of the time series, and we have $P(\mathcal{D}_{n,t:m}|\text{no further changepoint } b^\ddagger > b^\dagger, \mathcal{D}_{\pi_n,(t-1):(m-1)}, b^\dagger = t-1) = \Psi(\mathcal{D}_n^{\pi_n}[t:m])$. The second term is given by $P(\text{no further changepoint } b^\ddagger > b^\dagger|b^\dagger =$

$t-1) = 1 - P(\exists\,\text{further changepoint}\, b^{\ddagger} > b^{\dagger}|b^{\dagger} = t-1)$, where $\exists$ is the mathematical "exists" symbol. Recall that $P(\exists\,\text{further changepoint}\, b^{\ddagger} > b^{\dagger}|b^{\dagger} = t-1)$ is defined by the point process prior, which for $t > 2$ is equal to $G([m-1]-[t-1]) = G(m-t)$, while for $t = 2$ it is equal to $G_0(m-t) = G_0(m-2)$. Inserting these terms into (36), we get:

$$\begin{aligned}&P(\mathcal{D}_{n,t:m}, \text{no further changepoint}\, b^{\ddagger} > b^{\dagger}|\mathcal{D}_{\pi_n,(t-1):(m-1)}, b^{\dagger} = t-1)\\&\quad= \Psi(\mathcal{D}_n^{\pi_n}[t:m])[1 - G(m-t)]\end{aligned} \tag{37}$$

for $t > 2$, while for $t = 2$ we get:

$$\begin{aligned}&P(\mathcal{D}_{n,t:m}, \text{no further changepoint}\, b^{\ddagger} > b^{\dagger}|\mathcal{D}_{\pi_n,(t-1):(m-1)}, b^{\dagger} = t-1)\\&\quad= \Psi(\mathcal{D}_n^{\pi_n}[2:m])[1 - G_0(m-2)]\end{aligned} \tag{38}$$

Now, inserting (34) and (37) into (29) leads to (26), and inserting (35) and (38) into (29) gives (27). This completes the proof.

Having computed $Q(t|n,\pi_n)$ via the recursion (26)–(27), we can now set up another recursion to sample sequences of changepoints from the posterior distribution. The posterior distribution of the first changepoint $b_{n,1}$ given the parent set π_n is:

$$P(b_{n,1} = t|\mathcal{D}_n^{\pi_n}) = \frac{P(b_{n,1} = t, \mathcal{D}_n^{\pi_n})}{P(\mathcal{D}_n^{\pi_n})} \tag{39}$$

We expand the numerator in (39) as follows:

$$\begin{aligned}&P(b_{n,1} = t, \mathcal{D}_n^{\pi_n})\\&\quad= P(b_{n,1} = t, \mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,1:(m-1)})\\&\quad= P(b_{n,1} = t, \mathcal{D}_{n,2:t}, \mathcal{D}_{n,(t+1):m}|\mathcal{D}_{\pi_n,1:(m-1)})\\&\quad= P(\mathcal{D}_{n,2:t}, \mathcal{D}_{n,(t+1):m}|b_{n,1} = t, \mathcal{D}_{\pi_n,1:(m-1)})P(b_{n,1} = t|\mathcal{D}_{\pi_n,1:(m-1)})\\&\quad= P(\mathcal{D}_{n,(t+1):m}|b_{n,1} = t, \mathcal{D}_{\pi_n,t:(m-1)})P(\mathcal{D}_{n,2:t}|b_{n,1} = t, \mathcal{D}_{\pi_n,1:(t-1)})P(b_{n,1} = t)\end{aligned} \tag{40}$$

From definition (25), the first term is equal to $P(\mathcal{D}_{n,(t+1):m}|b_{n,1} = t, \mathcal{D}_{\pi_n,t:(m-1)}) = Q(t+1|n,\pi_n)$. Given that the first changepoint is at t, $\mathcal{D}_{n,2:t}$ is a homogeneous time series segment, and we get for the second term $P(\mathcal{D}_{n,2:t}|b_{n,1} = t, \mathcal{D}_{\pi_n,1:(t-1)}) = \Psi(\mathcal{D}_n^{\pi_n}[2:t])$. The third term is the prior probability of the first changepoint being at t, which is given by definition (24): $P(b_{n,1} = t) = g_0(t)$. Inserting these terms into (40), we get:

$$P(b_{n,1} = t, \mathcal{D}_n^{\pi_n}) = Q(t+1|n,\pi_n)\Psi(\mathcal{D}_n^{\pi_n}[2:t])g_0(t) \tag{41}$$

From definition (25) and recalling that $b_0 = 1$ is a pseudo changepoint, it is seen that the denominator in (39), $P(\mathcal{D}_n^{\pi_n}) = P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,1:(m-1)})$, is equal to $Q(2|n,\pi_n)$. Inserting this expression and (41) into (39), we get:

$$P(b_{n,1} = t|\mathcal{D}_n^{\pi_n}) = \frac{\Psi(\mathcal{D}_n^{\pi_n}[2:t])Q(t+1|n,\pi_n)g_0(t)}{Q(2|n,\pi_n)} \tag{42}$$

for $t = 2,\ldots,m-1$. The probability of no changepoint, $P(\mathcal{K}_n = 1)$, can easily be derived analogously and is given by:

$$P(\mathcal{K}_n = 1|\pi_n, \mathcal{D}_n^{\pi_n}) = \frac{\Psi(\mathcal{D}_n^{\pi_n}[2:m])[1 - G_0(m-2)]}{Q(2|n,\pi_n)} \tag{43}$$

where G_0 was defined in (22) and $[1 - G_0(m-2)]$ is the prior probability of the absence of any changepoint.[5]

Being able to sample the first changepoint from the posterior distribution $P(b_{n,1} = t|\mathcal{D}_n^{\pi_n})$, via (42), we next derive a recursion for the remaining changepoints. Assume that we have got a set of changepoints $\{b_{n,1}, \dots, b_{n,j-1}\}$ with $b_{n,1} < \cdots < b_{n,j-1}$. For the next changepoint $b_{n,j} > b_{n,j-1}$ we get:

$$\begin{aligned} P(b_{n,j} = t|b_{n,j-1}, \mathcal{D}_n^{\pi_n}) &= \frac{P(b_{n,j} = t, b_{n,j-1}, \mathcal{D}_n^{\pi_n})}{P(b_{n,j-1}, \mathcal{D}_n^{\pi_n})} \\ &= \frac{P(\mathcal{D}_n^{\pi_n}|b_{n,j} = t, b_{n,j-1}) P(b_{n,j} = t|b_{n,j-1})}{P(\mathcal{D}_n^{\pi_n}|b_{n,j-1})} \end{aligned} \tag{44}$$

We can expand the denominator as follows:

$$\begin{aligned} &P(\mathcal{D}_n^{\pi_n}|b_{n,j-1} = s) \\ &\quad = P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,1:(m-1)}, b_{n,j-1} = s) \\ &\quad = P(\mathcal{D}_{n,2:s}|\mathcal{D}_{\pi_n,1:(s-1)}, b_{n,j-1} = s) P(\mathcal{D}_{n,s+1:m}|\mathcal{D}_{\pi_n,s:(m-1)}, b_{n,j-1} = s) \\ &\quad = P(\mathcal{D}_{n,2:s}|\mathcal{D}_{\pi_n,1:(s-1)}, b_{n,j-1} = s) Q(s+1|n, \pi_n) \end{aligned} \tag{45}$$

where definition (25) has been used. For the numerator in (44), we get the following expansion:

$$\begin{aligned} P(\mathcal{D}_n^{\pi_n}|b_{n,j} = t, b_{n,j-1} = s) &= P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,1:(m-1)}, b_{n,j} = t, b_{n,j-1} = s) \\ &= P(\mathcal{D}_{n,2:s}|\mathcal{D}_{\pi_n,1:(s-1)}, b_{n,j-1} = s) \\ &\quad \times P(\mathcal{D}_{n,s+1:t}|\mathcal{D}_{\pi_n,s:(t-1)}, b_{n,j} = t, b_{n,j-1} = s) \\ &\quad \times P(\mathcal{D}_{n,t+1:m}|\mathcal{D}_{\pi_n,t:(m-1)}, b_{n,j} = t) \end{aligned} \tag{46}$$

The first term, $P(\mathcal{D}_{n,2:s}|\mathcal{D}_{\pi_n,1:(s-1)}, b_{n,j-1} = s)$, is also included in (45) and thus cancels out. For the second term, note that having two subsequent changepoints at positions $b_{n,j-1} = s$ and $b_{n,j} = t$ implies that the time series segment $\mathcal{D}_{n,(s+1:t)}$ is homogeneous, and hence $P(\mathcal{D}_{n,(s+1):t}|\mathcal{D}_{\pi_n,s:(t-1)}, b_{n,j} = t, b_{n,j-1} = s) = \Psi(\mathcal{D}_n^{\pi_n}[(s+1):t])$. From definition (25), the third term is given by $P(\mathcal{D}_{n,(t+1):m}|\mathcal{D}_{\pi_n,t:(m-1)}, b_{n,j} = t) = Q(t+1|n, \pi_n)$. Inserting these terms into (46) leads to:

$$\begin{aligned} &P(\mathcal{D}_n^{\pi_n}|b_{n,j} = t, b_{n,j-1} = s) \\ &\quad = P(\mathcal{D}_{n,2:s}|\mathcal{D}_{\pi_n,1:(s-1)}, b_{n,j-1} = s) \Psi(\mathcal{D}_n^{\pi_n}[(s+1):t]) Q(t+1|n, \pi_n) \end{aligned} \tag{47}$$

Inserting (45) and (47) into (44) and noting that $P(b_{n,j} = t|b_{n,j-1} = s) = g(t-s)$ with $g(.)$ defined in (23), we get for the posterior distribution of the j-th changepoint $b_{n,j} = t$ given the parent node set π_n and the previous changepoint $b_{n,j-1} = s$:

$$P(b_{n,j} = t|b_{n,j-1} = s, \mathcal{D}_n^{\pi_n}) = \frac{\Psi(\mathcal{D}_n^{\pi_n}[(s+1):t]) Q(t+1|n, \pi_n) g(t-s)}{Q(s+1|n, \pi_n)} \tag{48}$$

for $t = b_{n,j-1} + 1, \dots, m-1$.

[5] Recall that for a DBN and a time series of length m, there are $m-2$ possible changepoint locations, the first one being at position $t = 2$, and the last one at position $t = m-1$.

The probability of no further changepoint can be derived analogously and is given by

$$P_{\geq m} := \Psi(\mathcal{D}_n^{\pi_n}[(b_{n,j-1}+1):m])\frac{1-G_0(m-b_{n,j-1}-1)}{Q(b_{n,j-1}+1|n,\pi_n)} \tag{49}$$

where $G_0(.)$ was defined in (22).

Consequently, given a changepoint at $b_{n,j-1}=s$, the location of the next changepoint can be sampled from the discrete mass probability distribution $[P_{b_{n,j-1}+1},\dots,P_{m-1},P_{\geq m}]$ where $P_{\geq m}$ is the probability for no further changepoints. Having sampled changepoints $b_{n,1},\dots,b_{n,k-1}$ 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: The dynamic programming algorithm consists of two steps. In a first sweep through the data, the function $Q(t|n,\pi_n)$ is computed from (26)–(27). This function is then used in (48)–(49), where in a second sweep through the data a whole sequence of changepoints is sampled from the conditional distribution $P(.|\mathcal{D}_n^{\pi_n})$. The computational complexity is quadradic in the length of the time series, $\mathcal{O}(m^2)$. A sequence of changepoints uniquely determines the number of changepoints and the allocation vectors (these two representations are isomorphic). This allows us to unambiguously map the sampled changepoints onto a sample of $\{(\mathcal{K}_n,\mathbf{V}_n)\}$.

2.7.2 *Dynamic programming for a prior on the number of changepoints*

We will now revert to the prior that we have originally used, as defined below (10). The dynamic programming scheme remains essentially the same, with the difference that in the recursions of (26) and (48), the expression on the left-hand side will become explicitly dependent on the total number of changepoints.

As reminder of the notation, note that 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 $(n=1,\dots,N)$. $\mathcal{K}_n$ can be identified with $\mathcal{K}_n-1$ changepoints on the *discrete* set $\{2,\dots,m-1\}$ and there is a one-to-one mapping between $\mathbf{V}_n$ and the changepoint vector $\mathbf{b}_{\mathbf{V}_n}:=(b_{\mathbf{V}_n,0},\dots,b_{\mathbf{V}_n,\mathcal{K}_n})$ where $b_{\mathbf{V}_n,0}=1$ and $b_{\mathbf{V}_n,\mathcal{K}_n}=m$ are pseudo changepoints.

We now 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:

$$P(\mathcal{K}_n,\mathbf{V}_n|\pi_n,\mathcal{D}_n^{\pi_n}) = P(\mathcal{K}_n|\pi_n,\mathcal{D}_n^{\pi_n})P(\mathbf{V}_n|\mathcal{K}_n,\pi_n,\mathcal{D}_n^{\pi_n}) \tag{50}$$

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 $\mathcal{D}_{\pi_n,s:t}=\{\mathcal{D}_{\pi_n,i}:s\leq i\leq t\}$.

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 $\mathcal{D}_{\pi_n,1:(m-1)}$, $\mathcal{K}_n$ mixture components, and the allocation vector $\mathbf{V}_n$. The local score of X_n can be factorized using (6). Mapping the allocation vector $\mathbf{V}_n$ onto the changepoint vector $\mathbf{b}_{\mathbf{V}_n}$ we obtain as alternative representation:

$$\begin{aligned}&\Psi^\dagger(\mathcal{D}_n^{\pi_n}[\mathcal{K}_n,\mathbf{V}_n])\\ &\quad = P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,1:(m-1)},\mathcal{K}_n,\mathbf{b}_{\mathbf{V}_n}) = \prod_{k=0}^{\mathcal{K}_n-1}\Psi(\mathcal{D}_n^{\pi_n}[(b_{\mathbf{V}_n,k}+1):b_{\mathbf{V}_n,k+1}])\end{aligned} \tag{51}$$

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

$$P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,1:(m-1)},\mathcal{K}_n) = \sum_{\mathbf{b}_n \in \mathcal{B}(\mathcal{K}_n)} P(\mathbf{b}_n) \prod_{k=0}^{\mathcal{K}_n-1} \Psi(\mathcal{D}_n^{\pi_n}[(b_{n,k}+1):b_{n,k+1}]) \tag{52}$$

where $\mathcal{B}(\mathcal{K}_n)$ is the set of all valid changepoint vectors $\mathbf{b}_n = (b_{n,0},\ldots,b_{n,\mathcal{K}_n})$ 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 (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 $b_{n,j} = t-1$ and restrict on the data sub-segment $\mathcal{D}_n^{\pi_n}[(t-1):(m-1)]$:

$$\begin{aligned} &P(\mathcal{D}_{n,t:m}|\mathcal{D}_{\pi_n,(t-1):(m-1)},\mathcal{K}_n, b_{n,j} = t-1) \\ &\quad = \sum_{\mathbf{b}_n^j \in \mathcal{B}^j(\mathcal{K}_n|b_{n,j}=t-1)} P(\mathbf{b}_n^j) \prod_{k=j}^{\mathcal{K}_n-1} \Psi(\mathcal{D}_n^{\pi_n}[(b_{n,k}+1):b_{n,k+1}]) \end{aligned} \tag{53}$$

where $\mathcal{B}^j(\mathcal{K}_n|b_{n,j} = t-1)$ is the set of all valid changepoint vectors $\mathbf{b}_n^j = (b_{n,j+1},\ldots,b_{n,\mathcal{K}_n})$ on the discrete interval $\{t+1,\ldots,m-2\}$ with $b_{n,i+1} - b_{n,i} > 1$, $b_{n,j} = t-1$ and $b_{n,\mathcal{K}_n} = m$. Different from (11) and (52) 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 $j = 0,\ldots,\mathcal{K}_n - 1$ we set $Q_j^{\mathcal{K}_n}(t|n,\pi_n) = P(\mathcal{D}_{n,t:m}|\mathcal{D}_{\pi_n,(t-1):(m-1)}, \mathcal{K}_n, b_{n,j} = t-1)$ for $t = 2(j+1),\ldots,m-2(\mathcal{K}_n - j)+1$ and let $Q_j^{\mathcal{K}_n}(t|n,\pi_n)$ be zero otherwise, i.e. for $t < 2(j+1)$ and $t > m-2(\mathcal{K}_n - j)+1$. This definition corresponds to the one above (25).

It can be seen from (52) 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 $t = 2\mathcal{K}_n,\ldots,m-1$:

$$Q_{\mathcal{K}_n-1}^{\mathcal{K}_n}(t|n,\pi_n) = \Psi(\mathcal{D}_n^{\pi_n}[(t-1):(m-1)]) \tag{54}$$

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

In analogy to (26) and as a special case of the scheme described in Fearnhead (2006), we obtain the following recursion: For $\mathcal{K}_n > 1$, $j = 0,\ldots,\mathcal{K}_n - 2$ and $t = 2(j+1),\ldots, m-2(\mathcal{K}_n - j)+1$:

$$\begin{aligned} Q_j^{\mathcal{K}_n}(t|n,\pi_n) = &\sum_{s=t+1}^{m-2(\mathcal{K}_n-j-1)} \Psi(\mathcal{D}_n^{\pi_n}[t:s]) Q_{j+1}^{\mathcal{K}_n}(s+1|n,\pi_n) \\ &\times P(b_{n,j} = t-1|b_{n,j+1} = s,\mathcal{K}_n) \end{aligned} \tag{55}$$

where the bounds on t as well as the upper summation index allow for the changepoints that still need to be included.[6]

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

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:

$$P(b_{n,j}=t-1|b_{n,j+1}=s,\mathcal{K}_n)=P(m,\mathcal{K}_n,s,t):=\frac{s-t}{\binom{m-2}{2(\mathcal{K}_n-1)+1}} \tag{56}$$

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: $P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,1:(m-1)},\mathcal{K}_n)$ 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 (6)) is equal to the local BGe score of X_n (see (3)):

$$P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,1:(m-1)},\mathcal{K}_n=1)=\Psi(\mathcal{D}_n^{\pi_n}) \tag{57}$$

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

$$P(\mathcal{K}_n=k^\star|\mathcal{D}_{n,2:m},\mathcal{D}_{\pi_n,1:(m-1)})=\frac{P(\mathcal{K}_n=k^\star)P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,1:(m-1)},\mathcal{K}_n=k^\star)}{\sum_{k=1}^{\mathcal{K}_{MAX}}P(\mathcal{K}_n=k)P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,1:(m-1)},\mathcal{K}_n=k)} \tag{58}$$

where $P(\mathcal{K}_n)$ is a Poisson distribution with $\lambda=1$ truncated to $1\le\mathcal{K}_n\le\mathcal{K}_{MAX}$ in our cpBGe model.

After having sampled $\mathcal{K}_n=k$ from $P(\mathcal{K}_n|\mathcal{D}_{n,2:m},\mathcal{D}_{\pi_n,1:(m-1)})$, we can sample an allocation vector $\mathbf{V}_n$ from $P(\mathbf{V}_n|\mathcal{K}_n=k,\mathcal{D}_{n,2:m},\mathcal{D}_{\pi_n,1:(m-1)})$ 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 $j=1,\dots,k-1$ from the following distribution:

$$\begin{aligned}&P(b_{\mathbf{V}_n,j}=s|b_{\mathbf{V}_n,j-1},\mathcal{D}_n^{\pi_n},\mathcal{K}_n=k)\\&\quad=\frac{\Psi(\mathcal{D}_n^{\pi_n}[(b_{\mathbf{V}_n,j-1}+1):s])Q_j^k(s+1|n,\pi_n)P(m,k,s,b_{\mathbf{V}_n,j-1}+1)}{Q_{j-1}^k(b_{\mathbf{V}_n,j-1}+1|n,\pi_n)}\end{aligned} \tag{59}$$

as shown in Fearnhead (2006). Note that this recursion is analogous to (48) from the previous subsection. The dynamic programming scheme works as follows: (i) We sample $\mathcal{K}_n=k$ from (58). (ii) For $k=1$ we have no changepoints and for $k>1$ we can subsequently employ (59) to sample the locations of the $k-1$ changepoints. Because of the one-to-one mapping between changepoints and allocation vectors, the sampled changepoints $b_{\mathbf{V}_n,1},\dots,b_{\mathbf{V}_n,k-1}$ give a unique allocation vector $\mathbf{V}_n$ which can be seen as directly sampled from $P(\mathbf{V}_n|\mathcal{K}_n=k,\mathcal{D}_{n,2:m},\mathcal{D}_{\pi_n,1:(m-1)})$.

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

2.8 A Gibbs MCMC sampling scheme for the cpBGe model

In Sects. 2.7.1 and 2.7.2 we have described dynamic programming schemes that can be used for sampling for each domain node X_n from the conditional posterior distribution $P(\mathcal{K}_n,\mathbf{V}_n|\pi_n,\mathcal{D}_n^{\pi_n})$. The scheme of Sect. 2.7.2 employs the same prior distribution as the

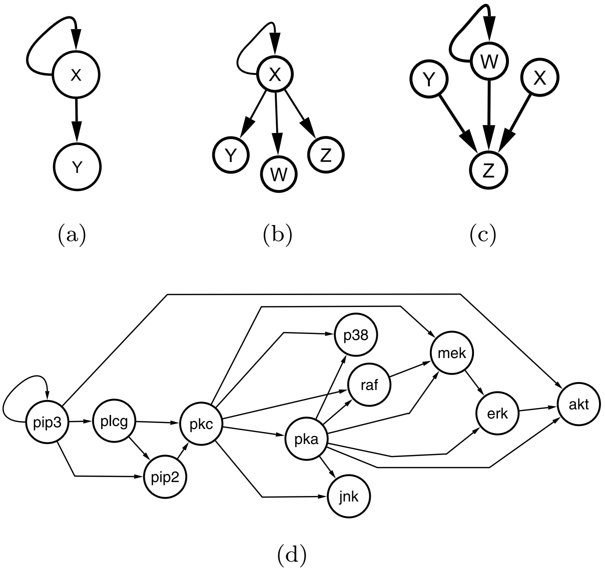
Metropolis-Hastings (MH) MCMC samplers. The scheme of Sect. 2.7.1 uses a modified prior distribution for $P(\mathcal{K}_n, \mathbf{V}_n)$ but can be computed more efficiently. Earlier in Sect. 2.5 we have shown that for each node X_n the parent node set π_n can be sampled from the posterior distribution $P(\pi_n|\mathcal{K}_n, \mathbf{V}_n, \mathcal{D}_n^{1:N})$.

Bringing these results together we can construct a Gibbs MCMC sampling scheme for sampling from the joint posterior distribution $P(\pi_n, \mathcal{K}_n, \mathbf{V}_n|\mathcal{D}_n^{1:N})$ by iteratively sampling from the two conditional distributions: $P(\pi_n|\mathcal{K}_n, \mathbf{V}_n, \mathcal{D}_n^{1:N})$ and $P(\mathcal{K}_n, \mathbf{V}_n|\pi_n, \mathcal{D}_n^{\pi_n})$. Using these Gibbs samplers independently for each node X_n ($n = 1, \ldots, N$) gives node-specific samples $\{(\pi_n^i, \mathcal{K}_n^i, \mathbf{V}_n^i) : i = 1, \ldots, I\}$ which can be merged into a sample $\{(\mathcal{G}^i, \mathbf{K}^i, \mathbf{V}^i) : i = 1, \ldots, I\}$ from $P(\mathcal{G}, \mathbf{K}, \mathbf{V}|\mathcal{D})$: $\mathbf{K}^i = (\mathcal{K}_1^i, \ldots, \mathcal{K}_N^i)$, $\mathbf{V}^i = (\mathbf{V}_1^i, \ldots, \mathbf{V}_N^i)$, and for $j, n \in \{1, \ldots, N\}$ we have that $\mathcal{G}^i$ possesses the edge $X_j \to X_n$ if and only if $X_j \in \pi_n^i$ for $j, n \in \{1, \ldots, N\}$. Note that the two sampling steps of the Gibbs samplers are computationally more expensive than the corresponding Metropolis-Hastings moves. On the other hand, the combined sampling scheme yields larger steps at acceptance probability 1 in both the parent node set and the allocation vector configuration spaces so that convergence may be reached in fewer MCMC steps. In our experiments we will therefore cross-compare the performance of the Metropolis-Hastings (MH) samplers and the Gibbs MCMC sampling schemes in terms of convergence and mixing.

3 Data

3.1 Synthetic network data: non-homogeneous data with node-specific changepoints

To assess the performance of the proposed cpBGe model, we apply it to synthetic data generated from the four different network structures shown in Fig. 2. For the synthetic network data we use a unique time series length of $m = 41$. Substantially shorter time series hardly leave enough data for posterior inference, whereas substantially longer time series in systems biology are rare due to the high laboratory costs.



(a) (b) (c)

(d)

Fig. 2 Networks for synthetic data generation. Panels (**a**–**c**) show elementary network motifs (Shen-Orr et al. 2002). Panel (**d**) shows a protein signal transduction network studied in Sachs et al. (2005), with an added feedback loop on the root node '*PIP3*'

Figure 2a shows the smallest synthetic network that we consider. It consists of two domain nodes X and Y, and there are two edges, namely a feedback-loop $X \to X$ so that there is autocorrelation in the time series $X(.)$, and a second edge from X to Y, which is modelled by a piecewise linear process with changing (time-dependent) coefficient $\beta(t)$:

$$X(t+1) = \sqrt{1-\varepsilon^2} \cdot X(t) + \varepsilon \cdot \phi_X(t+1) \tag{60}$$

$$Y(t+1) = \beta(t) \cdot X(t) + c \cdot \phi_Y(t+1) \tag{61}$$

where $\varepsilon \in [0,1]$, and $\phi_X(1), \phi_X(2), \dots, \phi_Y(1), \phi_Y(2), \dots$ are i.i.d. normally distributed random variables.

Equation (60) describes the autoregressive process $X(.)$, and $\sqrt{1-\varepsilon^2} \in [0,1]$ is the (auto-)correlation between $X(t)$ and $X(t+1)$ for all time-points t. That is, the autocorrelation does not vary in time, and we can tune the autocorrelation straightforwardly by setting ε correspondingly. E.g. for $\varepsilon = 1$ we have a white noise process of i.i.d. standard normally distributed random variables $X(t+1) = \phi_X(t+1)$. For $\varepsilon = 0$ we obtain a process $X(.)$ which is constant in time $X(t+1) = X(t)$ for all t without any noise injections. Furthermore for each $\varepsilon \in [0,1]$ $X(.)$ is standard normally distributed at each time point t. Accordingly, we initialize $X(1)$ with a random realization from a standard normal variable. From (61) it can be seen that the relationship between X and Y is implemented by a piecewise linear function, whose coefficient $\beta(t)$ changes in time. For this 2-node domain we generate $m = 41$ observations, and for simplicity, we set $\beta(t) = 1$ for the first $(2 \le t \le 11)$ and the last $(32 \le t \le 41)$ ten observations and $\beta(t) = -1$ for the 20 time points in between $(12 \le t \le 31)$. We decided to specify the noise level in terms of signal-to-noise ratios (SNRs). That is, we set the coefficient c dependent on the average input signals. To this end, we estimate the standard deviation $\sigma(\beta(t)X(t))$ of the input signals $\beta(1)X(1), \beta(2)X(2), \dots$ before noise injections in advance by exhaustive data simulations. Having estimated $\sigma(\beta(t)X(t))$ by the empirical standard deviation $\sigma(\widehat{\beta(t)X(t)})$ from the pre-simulated data, we compute the coefficient c as follows:

$$c = \frac{\sigma(\widehat{\beta(t)X(t)})}{SNR} \tag{62}$$

where SNR is the specified signal-to-noise ratio.

The same idea can be used for generating data from the network shown in Fig. 2b. For this 4-node network domain we define:

$$\begin{aligned} X(t+1) &= \sqrt{1-\varepsilon^2} \cdot X(t) + \varepsilon \cdot \phi_X(t+1) \\ Y(t+1) &= \beta_Y(t) \cdot X(t) + c_Y \cdot \phi_Y(t+1) \\ W(t+1) &= \beta_W(t) \cdot X(t) + c_W \cdot \phi_W(t+1) \\ Z(t+1) &= \beta_Z(t) \cdot X(t) + c_Z \cdot \phi_Z(t+1) \end{aligned} \tag{63}$$

where all noise terms $\phi_{.}(.)$ are i.i.d. standard normally distributed random variables. We initialize all three β coefficients with $+1$ and for the three nodes Y, W, and Z that are regulated by X, we flip an unbiased coin to determine whether the corresponding coefficient $\beta_{.}(t)$ changes its sign once (from $+1$ to -1) or twice (that is, from $+1$ to -1 and later back to $+1$), and we randomly draw the changepoint locations afterwards. For each of the three variables we independently draw the changepoint location(s) from uniform distributions (i) over the discrete interval $\{6, \dots, 36\}$ to avoid changepoints during the first/last five time points, and (ii) under the constraint that there are at least 5 time points between the two changepoint locations when a coefficient changes its sign twice. As described for the smaller network the three coefficients c_X, c_Z, c_W can be computed from pre-simulated data to ensure that a pre-specified signal-to-noise ratio SNR is given, e.g.:

$$c_Y = \frac{\sigma(\widehat{\beta_Y(t)X}(t))}{SNR} \tag{64}$$

where SNR is the specified signal-to-noise ratio and $\sigma(\widehat{\beta_Y(t)X}(t))$ can be estimated from pre-simulated data. For these two networks with $N = 2$ and $N = 4$ nodes we consider $n_{pc} = 20$ different parameter combinations of $SNR \in \{100, 10, 3, 1, 0.5\}$ and $\varepsilon \in \{0.99, 0.5, 0.25, 0.1\}$, and we generate $n_{pc,i} = 25$ independent data instantiations for each combination (SNR, ε).

The same idea can be used to generate synthetic data for the (slightly-modified) RAF-pathway shown in Fig. 2d. Figure 2d was extracted from the systems biology literature (Sachs et al. 2005) and represents a well-studied protein signal transduction pathway. We added an extra feedback loop on the root node '*PIP3*' to allow the generation of a Markov chain with non-zero autocorrelation; note that this modification is not biologically implausible (Dougherty et al. 2005). Node '*PIP3*' has a recurrent feedback loop:

$$PIP3(t+1) = \sqrt{1-\varepsilon^2} \cdot PIP3(t) + \varepsilon \cdot \phi_{PIP3}(t+1) \tag{65}$$

The realizations of the other 10 domain nodes are linear combinations of the realizations of its parent nodes at the preceding time points plus realizations of i.i.d. standard normal distributions (noise injections). E.g. for '*PIP2*':

$$PIP2(t+1) = \beta_{PIP3}(t) \cdot PIP3(t) + \beta_{PLCG}(t) \cdot PLCG(t) + c_{PIP2} \cdot \phi_{PIP2}(t+1) \tag{66}$$

For each node we flip an unbiased coin to determine whether its coefficients change their values once or twice, and we randomly draw the changepoint locations independently for each domain node from discrete uniform distributions under the constraints (i) that there is no changepoint among the first/last 5 observations and (ii) that there are at least 5 time points between changepoints. Different from the regulatory mechanisms for the smaller domains in Fig. 2a–b, we sample new coefficients $\beta_.$ at each changepoint from continuous uniform distributions on the interval $[0.5, 2]$ and we flip an unbiased coin for each re-sampled coefficient to determine its (new) sign.[7] As before, the coefficients c can be computed from pre-simulated data to ensure that a pre-specified signal-to-noise ratio (SNR) is given, e.g.:

$$c_{PIP2} = \frac{\sigma(\beta_{PIP3}(t)PIP3\widehat{(t) + \beta}_{PLCG}(t)PLCG(t))}{SNR} \tag{67}$$

For the RAF network with $N = 11$ nodes we consider $n_{pc} = 15$ different parameter combinations (ε, SNR) with $\varepsilon \in \{0.5, 0.25, 0.1\}$ and $SNR \in \{10, 3, 1, 0.5, 0.1\}$, and for each combination (ε, SNR) we generate $n_{pc,i} = 5$ independent[8] data sets with $m = 41$ observations.

[7] Here changepoints do not necessarily imply changes of the signs of the coefficients.

[8] The changepoint locations and the coefficients are sampled independently for each of the $n_{pc,i} = 5$ data sets.

For the network structure shown in Fig. 2c we generated data using sinusoidal transfer functions. This leads to a stronger mismatch between the model and the data-generation mechanism. We set:

$$\begin{aligned} &X(t+1) = \phi_X(t); \qquad Y(t+1) = \phi_Y(t); \\ &W(t+1) = W(t) + c + c_W \cdot \phi_W(t) \\ &Z(t+1) = c_X \cdot X(t) + c_Y \cdot Y(t) + \sin(W(t)) + c_Z \cdot \phi_Z(t+1) \end{aligned} \tag{68}$$

where the $\phi_.(.)$ are i.i.d. standard normally distributed, and the drift term c was set to $c = \frac{2\pi}{m}$ to ensure that the complete period $[0, 2\pi]$ of the sinusoid is involved. We employed different values $c_X = c_Y \in \{0.25, 0.5\}$ and $c_Z, c_W \in \{0.25, 0.5, 1\}$ to vary the signal-to-noise ratio and the amount of autocorrelation in W. With $c_X = c_Y$ this yields $n_{pc} = 18$ parameter combinations (c_X, c_Z, c_W) and for each combination we generate $n_{pc,i} = 25$ independent data sets with $m = 41$ observations.

Finally, the latter idea of a sinusoidal transfer function was also employed to generate alternative data sets from the network shown in Fig. 2a. That is, to obtain a mismatch between our model and the data generation mechanism we generated data from the following state-space equations, which employ a sinusoidal transfer function to describe nonlinearity:

$$X(t+1) = X(t) + c + c_X \cdot \phi_X(t), \qquad Y(t+1) = \sin(X(t)) + c_Y \cdot \phi_Y(t) \tag{69}$$

where the $\phi_.(.)$ are i.i.d. standard normally distributed and $c = \frac{2\pi}{m}$ is the drift term. We employ different parameter values $c_X \in \{0.1, 0.25, 0.5, 1\}$ and $c_Y \in \{0.1, 0.25, 0.5, 1\}$ to vary the strength of the autocorrelation and the signal-to-noise ratio. For each of the $n_{pc} = 16$ parameter combinations (c_X, c_Y) we generate $n_{pc,i} = 25$ independent data sets of length $m = 41$.

An overview of all synthetic data sets that have been generated is given in Table 2.

Table 2 Overview of all generated synthetic network data sets

	NET1	NET2	NET3	NET4	NET5
Structure	Figure 2a	Figure 2b	Figure 2c	Figure 2d	Figure 2a
Nodes	$N = 2$	$N = 4$	$N = 4$	$N = 11$	$N = 2$
Equations	(60)–(62)	(63)–(64)	(68)	(65)–(67)	(69)
Observations	$m = 41$	$m = 41$	$m = 41$	$m = 41$	$m = 41$
Transfer functions	piecewise linear	piecewise linear	linear *and* sinusoidal	piecewise linear	sinusoidal
Parameter set...	(ε, SNR)	(ε, SNR)	$(c_X = c_Y, c_W, c_Z)$	(ε, SNR)	(c_X, c_Y)
...combinations	$n_{pc} = 20$	$n_{pc} = 20$	$n_{pc} = 18$	$n_{pc} = 15$	$n_{pc} = 16$
...replications	$n_{pc,i} = 25$	$n_{pc,i} = 25$	$n_{pc,i} = 25$	$n_{pc,i} = 5$	$n_{pc,i} = 25$

We denote by n_{pc} the number of considered parameter combinations, and by $n_{pc,i}$ the number of independent data instantiations for each parameter combination

3.2 Synthetic network data: homogeneous data and non-homogeneous data with changepoints that are common to all nodes

We also want to investigate how the proposed cpBGe Bayesian network model (see Sect. 2.2) compares with competing Bayesian network models on homogeneous network data[9] and on non-homogeneous data where all changepoints are tied together.[10] In Sect. 3.1 we have described how to generate synthetic network data with node-specific changepoints for various network topologies shown in Fig. 2. We now focus on the RAF-network topology shown in Fig. 2d with node '*PIP3*' again possessing a recurrent feedback loop:

$$PIP3(t+1) = \sqrt{1-\varepsilon^2} \cdot PIP3(t) + \varepsilon \cdot \phi_{PIP3}(t+1) \tag{70}$$

As before, the realizations of the other 10 nodes are linear combinations of the realizations of its parent nodes at the preceding time points plus realizations of i.i.d. standard normal distributions (noise injections). But different from the regulatory mechanisms with node specific changepoints described in Sect. 3.1, we now consider two different scenarios (S1) and (S2):

Scenario (S1): Homogeneous network data can be obtained by using regression coefficients that are constant in time, symbolically $\beta(t) = \beta$ for all coefficients β and all time points t. E.g. (66) is replaced by:

$$PIP2(t+1) = \beta_{PIP3} \cdot PIP3(t) + \beta_{PLCG} \cdot PLCG(t) + c_{PIP2} \cdot \phi_{PIP2}(t+1) \tag{71}$$

Scenario (S2): Non-homogeneous network data where changepoints are common to all nodes can be obtained as follows: For each data set one single set of changepoints is drawn, which then applies to all nodes (rather than drawing independent changepoints for each node). Thus, *all* regression coefficients are re-sampled at changepoints.[11] For our simulation study we assume that there is one single changepoint whose location is enforced to be located in the middle of the time series. That is, we want to avoid changepoints in the margins that would only bring about moderate degrees of non-homogeneity. As we generate time series of length $m = 41$, we randomly draw the location of the changepoint from the discrete set $\{16, \dots, 25\}$.

For both additional scenarios (S1) and (S2) we focus on the moderate autocorrelation parameter $\varepsilon = 0.25$ in (70) and on three signal-to-noise ratios $SNR \in \{10, 3, 1\}$. For both scenarios (S1) and (S2) this yields $n_{pc} = 3$ parameter combinations (ε, SNR), for which we generate $n_{pc,i} = 5$ independent data instantiations ($n_{pc,i} = 5$) with $m = 41$ observations each.

[9] Data that stem from the homogeneous Bayesian network model described in Sect. 2.1.

[10] Data that stem from the non-homogeneous Bayesian Gaussian Mixture (BGM) Bayesian network model (Grzegorczyk et al. 2010), with changepoints common to the whole network, rather than the cpBGe Bayesian network model proposed here; an overview is given in Table 4.

[11] The resulting data are non-homogeneous and therefore cannot be modelled with the standard BGe Bayesian network model described in Sect. 2.1. On the other hand, the full flexibility of the proposed cpBGe model (see Sect. 2.2) is not required. The changepoint variant of the BGM model, proposed in Grzegorczyk et al. (2010), appears to be ideal for this scenario.

Table 3 Gene expression time series segments for Arabidopsis

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

The table contains an overview of the experimental conditions under which each of the gene expression experiments was carried out

3.3 *Arabidopsis thaliana* gene expression time series

As an application to real data 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,[12] using GeneSpring© software (Agilent Technologies). We combine four time series, which differ with respect to the pre-experiment entrainment condition and the harvesting intervals: $T_e \in \{10, 12, 14\}$, and $\tau \in \{2, 4\}$. 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 3. We focus our analysis on 9 circadian genes[13] (i.e. genes involved in circadian regulation), and we merge all four time series into one single data set. The objective is to test whether the proposed cpBGe model detects the different segments (see Table 3). Since the gene expression values at the first time point of a 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 mathematically in Appendix A. 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.

4 Simulation and implementation details

4.1 Implementation of other approaches

In our cross-method comparison we have compared the proposed cpBGe model with four other Bayesian network models. The standard Gaussian Bayesian network model BGe was

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

[13] These 9 circadian genes are LHY, TOC1, CCA1, ELF4, ELF3, GI, PRR9, PRR5, and PRR3.

briefly described in Sect. 2.1. Another standard Bayesian network model, which we have included in our study, is the discrete multinomial BDe model with a Dirichlet distribution of the unknown parameters. Details on the parameter settings for these two models can be found in Sect. 4.2. We have also included a slightly modified version of the Bayesian Gaussian Mixture (BGM) Bayesian network model of Grzegorczyk et al. (2008). The BGM model differs from the proposed cpBGe model in two aspects. First, the latent variable allocation is common to the whole network, that is, the changepoints are not node-specific. Second, the assignment of data points to components is not affected by a changepoint process, but via a free allocation of the latent variables. The second aspect leads to a more flexible model, which could be useful for static Bayesian networks and i.i.d. data rather than time series. When combined with the node-specific allocations of the cpBGe model, it will lead to a nonlinear rather than non-stationary model. However, for time series, employing a free allocation model discards relevant information about the structure of the data. Namely, that under the assumption of a Markovian dependence, adjacent time points are *a priori* likely to be governed by the same process. Moreover, the free allocation model leads to a higher complexity of the latent variable configuration space, which is likely to adversely affect the mixing and convergence properties of the MCMC sampler. In order that the comparison between the two models is not dominated by (1) the different degrees of complexity of the MCMC simulations or (2) the presence versus absence of prior information about the data structure, we replace the free allocation model originally used for the Bayesian Gaussian Mixture (BGM) model in Grzegorczyk et al. (2008) by a changepoint process on the discrete time points. This yields the model presented in Grzegorczyk et al. (2010) except that the continuous changepoint process is substituted for a simpler discrete changepoint process, as in the proposed cpBGe model. Including the changepoint variant of the BGM model ensures that our comparison focuses on the aspect of employing node-specific rather than common changepoints, that is, it allows us to investigate to what extent this additional model flexibility leads to an improved network reconstruction accuracy. Some technical details for the new variant of the BGM model can be found in Appendix B. Another nonlinear Bayesian network model based on node-specific Gaussian mixture models has been proposed by Ko et al. (2007). In this approach, data are assigned *node-specifically* and *individually* to mixture components, resulting in high model flexibility. The authors resort to the Bayesian information criterion (BIC) of Schwarz (1978) for graph selection, which is only a good approximation to the marginal likelihood in the limit of large data sets. We refer to this Gaussian mixture model as the GM$_{BIC}$ model, and we relegate all technical details for the GM$_{BIC}$ model to Appendix C. We applied GM$_{BIC}$ 10 times independently with different initializations. In our study the initializations were outputs of the k-means cluster algorithm, whose initializations were sampled from an $N(\boldsymbol{\mu}, \mathbf{I})$ distribution, where $\mathbf{I}$ is the identity matrix and $\boldsymbol{\mu}$ is a random expectation vector with entries sampled independently from continuous uniform distributions on $[-1, 1]$. With this approach we obtain 10 estimates $\mathcal{G}^1, \ldots, \mathcal{G}^{10}$ of the underlying graph structure. We have used these estimates to compute individual edge scores. The score of an individual edge can be estimated by the fraction of graphs in $\{\mathcal{G}^1, \ldots, \mathcal{G}^{10}\}$ that obtain the edge of interest. For the evaluation of the network reconstruction accuracy (see Sect. 4.4) we have treated the individual edge scores of GM$_{BIC}$ analogously to the marginal edge posterior probabilities obtained from the Bayesian approaches (see Sect. 4.3). An overview of the five Bayesian network models included in our cross-method comparison (see Sect. 5.2) is given in Table 4.

Table 4 Overview of the five Bayesian network models included in our cross-method comparison in Sect. 5.2. See text for further details

Model	BDe	BGe	GM$_{BIC}$	BGM	cpBGe
From	Cooper & Herskovits	Geiger & Heckerman	Ko et al.	Grzegorczyk et al.	proposed here
Year	1992	1994	2008	2010	2010
Data format	Discrete	Continuous	Continuous	Continuous	Continuous
Score	Marginal Likelihood	Marginal Likelihood	BIC	Marginal Likelihood	Marginal Likelihood
Non-homogeneous modelling capacity	No	No	Yes	Yes	Yes
Latent variable format	–	–	Free allocation	changepoint process	changepoint process
Node-specific changepoints	–	–	Yes	No	Yes

4.2 Data pre-processing and hyperparameter settings

In all our simulations, synthetic data were standardized to zero mean and marginal variance of 1 for all dimensions. For BGe, BGM, and our cpBGe model, the prior distribution of the unknown parameters is assumed to be the conjugate Gaussian-Wishart distribution, and the hyperparameters were set as follows. The Wishart distribution has $\alpha = N + 3$ degrees of freedom, and its parameter matrix ($\mathbf{T}_0$ in the notation of Geiger and Heckerman 1994) was set to the identity matrix. The mean vector $\boldsymbol{\mu}$ of the Gaussian was set to the zero vector and the unknown covariance matrix $\boldsymbol{\Sigma}$ of the Gaussian was assumed to be equal to $(\nu\mathbf{W})^{-1}$, where $\mathbf{W}$ is the realization of the Wishart distribution and ν was set to 1. This setting reflects our prior belief that all domain variables are i.i.d. standard Gaussian distributed, where the hyperparameters α and ν (which correspond to equivalent prior sample sizes) are chosen as uninformative as possible subject to the regulatory conditions discussed in Geiger and Heckerman (1994). For the discrete BDe model the hyperparameters of the Dirichlet prior were also specified as uninformative as possible, as in Giudici and Castelo (2003).[14] The data discretization required for the BDe model was accomplished with the Information Bottleneck algorithm (IBA) (Hartemink 2001). First, we applied quantile discretization to discretize the values of each variable independently into 20 discrete levels. Afterwards, a dynamic version of IBA[15] was run until we had three discrete levels for each variable.

[14] The total prior precision α was set to 1, and we set $\alpha_{n,j,k} = \frac{\alpha}{r_n q_n}$ where r_n is the number of possible values for the n-th domain node and q_n is the number of possible different realizations of the parent node set π_n. The hyperparameters $\alpha_{n,j,k}$ determine the shape of the conjugate Dirichlet prior, as discussed in Heckerman and Geiger (1995).

[15] IBA merges for each variable neighbouring levels such that the pairwise information loss—in terms of the average mutual information between this variable and the others—is minimized. The standard algorithm for static data was modified to take into account (i) that the pairwise mutual information MI between two variables X and Y has to be computed with a time lag $\tau = 1$ and is given by the average of $MI(X(t), Y(t+1)$

Table 5 Overview of the MCMC schemes and numbers of MCMC iterations [in thousand (k)] in our comparative convergence study

Symbol	MH(−FLIP)	MH(+FLIP)	Gibbs($K = 10$)	Gibbs($K = 5$)	Gibbs-NBIN(p, k)
Sampling scheme	RJMCMC	RJMCMC	Gibbs	Gibbs	Gibbs
Sampling networks	structure MCMC without FLIP move	structure MCMC with FLIP move	sampling from the "Boltzmann" distribution	sampling from the "Boltzmann" distribution	sampling from the "Boltzmann" distribution
Sampling change-points	birth and death moves	birth and death moves	dynamic programming Sect. 2.7.2 $\mathcal{K}_{MAX} = 10$	dynamic programming Sect. 2.7.2 $\mathcal{K}_{MAX} = 5$	dynamic programming Sect. 2.7.1
Iterations	1100k	1100k	0.55k	1.1k	5.5k

Details on the dynamic programming (DP) scheme can be found in Sects. 2.7.1 and 2.7.2. It depends on the prior distributions over the changepoints. Three prior distributions were chosen. Gibbs($K = 10$): A Poisson prior on the number of components truncated at $\mathcal{K}_{MAX} = 10$, and an even-numbered order statistics prior on the changepoint locations. Gibbs($K = 5$): Idem, but truncated at $\mathcal{K}_{MAX} = 5$. Gibbs-NBIN(p, k): A point process prior on the distances between changepoints. The last row "Iteration" shows the total numbers of MCMC iterations that were performed. For each data set from the RAF network all MCMC runs could be accomplished in about 45 minutes using our Matlab© implementation on a SunFire X4100M2 machine with AMD Opteron 2224 SE dual-core processor. The simulations on the *Arabidopsis thaliana* data took approximately the same amount of time

4.3 MCMC convergence

We have compared five MCMC sampling schemes, described in Sect. 2.3. An overview is given in Table 5. Our Matlab© implementations are available upon request. We ran our simulations on a SunFire X4100M2 machine with AMD Opteron 2224 SE dual-core processor. Using our implementation we observed for several RAF-network data sets with $N = 11$ variables and $m = 41$ data points that the computational costs of 2000 MCMC iterations of the Metropolis-Hastings (MH) MCMC sampling schemes with (MH(+FLIP)) or without (MH(−FLIP)) the flip operator are comparable to the computational costs of approximately 1 Gibbs sampling step[16] when the same Poisson/changepoint process prior was used and the maximal number of components was set to $\mathcal{K}_{MAX} = 10$.[17] We refer to this Gibbs sampler as Gibbs($K = 10$). We tried two 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 were approximately as expensive as 2000 MH iterations. We refer to this version of the Gibbs sampler as Gibbs($K = 5$). (ii) We observed that replacing the Poisson/changepoint process

and $MI(Y(t), X(t+1)$, and (ii) that recurrent feedback loops are valid so that for each variable X the pairwise mutual information between $X(t)$ and $X(t+1)$ has to be included.

[16] Note that 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, on the other hand, always consists of two steps, i.e. a new parent node set π_n *and* a new allocation vector $\mathbf{V}_n$ are sampled.

[17] Note that the upper limit $\mathcal{K}_{MAX} = 10$ was never sampled by any model.

prior by the point process prior[18] described in Sect. 2.7.1 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 about 2,000 MH steps. See Table 5 for the total MCMC run lengths in our study and an overview of the computational costs.

After the burn-in phase of s_1 MCMC iterations, s_2 graphs from the posterior distribution are sampled with the four MCMC based models BDe, BGe, BGM, and cpBGe. Since this series of s_2 graphs (one for each iteration of the sampling phase) tends to be auto-correlated, it is usually thinned out. That is, only $I_{s_2} < s_2$ equally spaced graphs are kept and used for inference. Let $\mathcal{G}^1, \dots, \mathcal{G}^{I_{s_2}}$ be the graph subsample after thinning out. Marginal edge posterior probabilities can then be computed as follows: 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:

$$e_{n,j} = \frac{1}{I_{s_2}} \sum_{i=1}^{I_{s_2}} \mathcal{G}^i(n,j) \tag{72}$$

where $\mathcal{G}^i(n,j)$ is an indicator function which is 1 if the i-th graph in the sample $\mathcal{G}^1, \dots, \mathcal{G}^{I_{s_2}}$ contains the edge $X_n \to X_j$, and 0 otherwise ($n, j \in \{1, \dots, N\}$). A first impression of convergence can be obtained by a scatter plot of the individual edge posterior probabilities $e_{n,j}$ of two independent (differently seeded) MCMC runs on the same data set. Another standard diagnostic that we apply to evaluate convergence is based on potential scale reduction factors (PSRFs), which are usually monitored alongside 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 data set. Discarding the first $s_1 = s$ iterations as the burn-in phase, I_s graph samples are taken from the remaining $s_2 = s$ MCMC iterations. 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):

$$\mathcal{B}(n,j) = \frac{1}{H-1} \sum_{h=1}^{H} (e_{n,j,h} - \overline{e}_{n,j,.})^2 \tag{73}$$

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

$$\mathcal{W}(n,j) = \frac{1}{H(I_s - 1)} \sum_{h=1}^{H} \sum_{i=1}^{I_s} (G^{h,i}(n,j) - e_{n,j,h})^2 \tag{74}$$

where $G^{h,i}(n,j)$ is 1 if the i-th graph in the sample taken in the h-th simulation contains the edge $X_n \to X_j$, and 0 otherwise. Following Brooks and Gelman (1998) the $PSRF(n,j)$

[18] We performed a grid-search ($p \in \{0.01, 0.02, \dots, 0.20\}$ and $k \in \{= 1, \dots, 5\}$) to find the parameter combination (p,k) of the negative binomial distribution in the point process model that gives the best approximation to the Poisson prior of the original model. Purely prior-driven Gibbs-NBIN(p,k) simulations on a theoretical time series of length $m = 41$ revealed that the best approximation in terms of the Kulback Leibler divergence is obtained for $p = 0.05$ and $k = 2$.

of the individual edge $X_n \to X_j$ is then given by:

$$PSRF(n,j) = \frac{(1-\frac{1}{I_s})\mathcal{W}(n,j) + (1+\frac{1}{H})\mathcal{B}(n,j)}{\mathcal{W}(n,j)} \tag{75}$$

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 a PSRF-based convergence diagnostic the fraction of edges $\mathcal{C}(\xi)$ whose PSRF is lower than a pre-defined threshold value ξ:

$$\mathcal{C}(\xi) = \frac{1}{N^2}\sum_{n=1}^{N}\sum_{j=1}^{N} Z_{PSRF<\xi}(PSRF(n,j)) \tag{76}$$

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

For the cross-method comparison the MCMC inference for BDe, BGe, BGM, and cpBGe was done with the Metropolis-Hastings sampling scheme (improved by the FLIP operator). For the RAF pathway data (NET4 in Table 2) and for the *Arabidopsis thaliana* data $2s = 1000{,}000$ MCMC iterations were performed. From the last $s = 500{,}000$ iterations, we sampled $I_s = 500$ graphs by sampling every 1,000th iteration and checked whether sufficient convergence was reached.[19] For the small networks with $N \leq 4$ nodes in Table 2, $2s = 100{,}000$ MCMC iterations were performed and we sampled $I_s = 50$ graphs from the last $s = 50{,}000$ iterations, by sampling every 1000th iteration.[20]

In the second part of the study our focus is on the convergence of the five different MCMC sampling schemes for the cpBGe model. We focus our diagnostics on single data sets from the RAF-network. We perform $H = 10$ independent MCMC simulations and consider four different thresholds for ξ ($\xi = 1.2, 1.1, 1.05, 1.02$). When monitoring the $\mathcal{C}(\xi)$ diagnostic, we have to take into consideration that the computational costs of Gibbs moves are higher than those of the Metropolis-Hastings moves. See Appendix D for details.

4.4 Network reconstruction accuracy

For all our synthetic network data sets the true underlying graph structure $\mathcal{G}^\diamond$ is known. We can therefore objectively assess the network reconstruction accuracy for each model and/or inference scheme. We assume that $\mathcal{G}^\diamond(n,j) = 1$ indicates that the true graph possesses the edge $X_n \to X_j$, while $\mathcal{G}^\diamond(n,j) = 0$ indicates that there is no edge from X_n to X_j. Each method in our study outputs a marginal edge posterior probability $e_{n,j}$ for every edge $\mathcal{G}^\diamond(n,j)$, and for $\zeta \in [0,1]$ we define $E(\zeta) := \{\mathcal{G}(n,j) | e_{n,j} \geq \zeta\}$ as the set of all edges $\mathcal{G}(n,j)$ whose posterior probabilities exceed the threshold ζ. Since the true edges are known, for each $E(\zeta)$ the number of true positive $TP[\zeta]$, false positive $FP[\zeta]$, true negative $TN[\zeta]$, and false negative $FN[\zeta]$ edges can be counted. From this we can compute the true positive rate $TPR[\zeta] = TP[\zeta]/(TP[\zeta] + FN[\zeta])$ (also called *recall* or *sensitivity*), the false positive rate $FPR[\zeta] = FP[\zeta]/(TN[\zeta] + FP[\zeta])$, and the precision

[19] We randomly selected three synthetic RAF-network data sets and analyzed each of them $H = 5$ times independently with the four MCMC-based methods. From the $H = 5$ independent graph samples we then computed for each method the fraction of edges $\mathcal{C}(\xi)$ whose PSRF was lower than $\xi = 1.2$. For the three data sets we found for each method that the fraction of edges $\mathcal{C}(1.2)$ was always greater than 0.9.

[20] For the small network domains there were no convergence problems and for each individual edge the PSRF diagnostic was always lower than 1.2.

$PRE[\zeta] = TP[\zeta]/(TP[\zeta] + FP[\zeta])$. Plotting the $TPR[\zeta]$ values (y-axis) against the corresponding $FPR[\zeta]$ values (x-axis) and connecting neighbouring points by linear interpolation gives the receiver operator characteristic (ROC) curve. The area under the ROC curve (AUC-ROC) is a quantitative measure that can be obtained by integrating the ROC curve on the interval [0, 1]; larger AUC-ROC values indicate a better network reconstruction accuracy, whereby 1 indicates perfect prediction, whereas 0.5 corresponds to a random estimator. Although AUC-ROC diagnostics are commonly used, a more informative picture of the network reconstruction accuracy can be obtained by integrating the Precision-Recall (PR) curve. PR curves can be obtained as follows: (i) The $PRE[\zeta]$ values (y-axis) are plotted against the corresponding $TPR[\zeta]$ values (x-axis). (ii) Different from ROC curves, neighbouring points cannot be connected by straight lines and a nonlinear interpolation is required.[21] In our implementation we use the interpolation scheme described in Davis and Goadrich (2006). (iii) As the precision is not defined for $TP = 0$ and $FP = 0$ ($PRE = 0/0$), we integrate the PR curve on the interval $[(1/E), 1]$ where E is the number of edges of the true graph $\mathcal{G}^{\diamond}$; i.e. we restrict on the area where at least one of the true edges has been learnt.

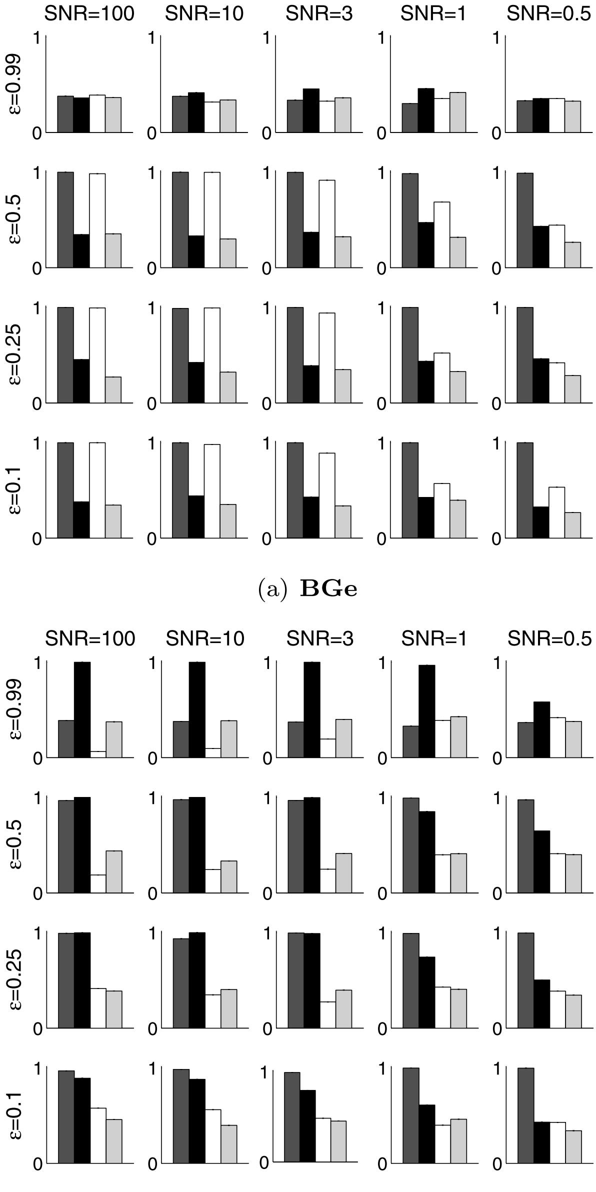
In our study we apply both criteria AUC-ROC and AUC-PR for assessing the network reconstruction accuracy; for a more detailed description and a theoretical comparison of both criteria we refer the reader to Davis and Goadrich (2006).

5 Results and discussion

5.1 Avoiding spurious feedback loops

Figures 3 and 4 show the marginal posterior probabilities of the four potentially possible edges in the 2-node network of Fig. 2a, predicted with the linear BGe model (top panel) and the proposed cpBGe model (bottom panel). The data were generated from the piecewise linear model of (60)–(61)—for Fig. 3—and the sinusoidal transfer function of (69)—for Fig. 4. In both cases, the linear BGe model shows a clear propensity for inferring the spurious self-loop $Y \to Y$. This systematic failure can be explained as follows. The functional dependence between nodes X and Y in Fig. 2a is nonlinear—either piecewise linear (see (61)) or of a sinusoidal form (see (69)). This nonlinear functional relationship cannot be adequately represented with a linear model, on which the BGe score is based. Consequently, the prediction of $Y(t+1)$ from $X(t)$ will tend to be poor. Note that for sufficiently small noise levels, the $Y(t)$'s exhibit a strong autocorrelation, by virtue of the autocorrelation of the $X(t)$'s, and the regulatory influence of $X(t)$ on $Y(t+1)$. As the latter regulatory influence cannot be learnt owing to the linear restriction of the model, the next best explanation is a direct modelling of the autocorrelation between the $Y(t)$'s themselves. This autocorrelation corresponds to an edge from $Y(t)$ to $Y(t+1)$ in the dynamic Bayesian network, which means, a feedback loop of Y acting back on itself in the state-space graph. The lack of nonlinear modelling flexibility hence explains why the BGe model systematically infers a spurious feedback loop, corresponding to the white bars in the histograms of Figs. 3 and 4. Compare this with the results for the proposed cpBGe model, shown in Figs. 3b and 4b. The general tendency is that the marginal posterior probabilities of the true edges (the two left bars in the histograms) clearly outweigh those of the spurious edges (the two right bars of the histograms). There are only two regimes where this tendency breaks down. In the top rows of

[21] The interpolation has to be done in terms of the precision PRE which corresponds to a nonlinear interpolation in data space.



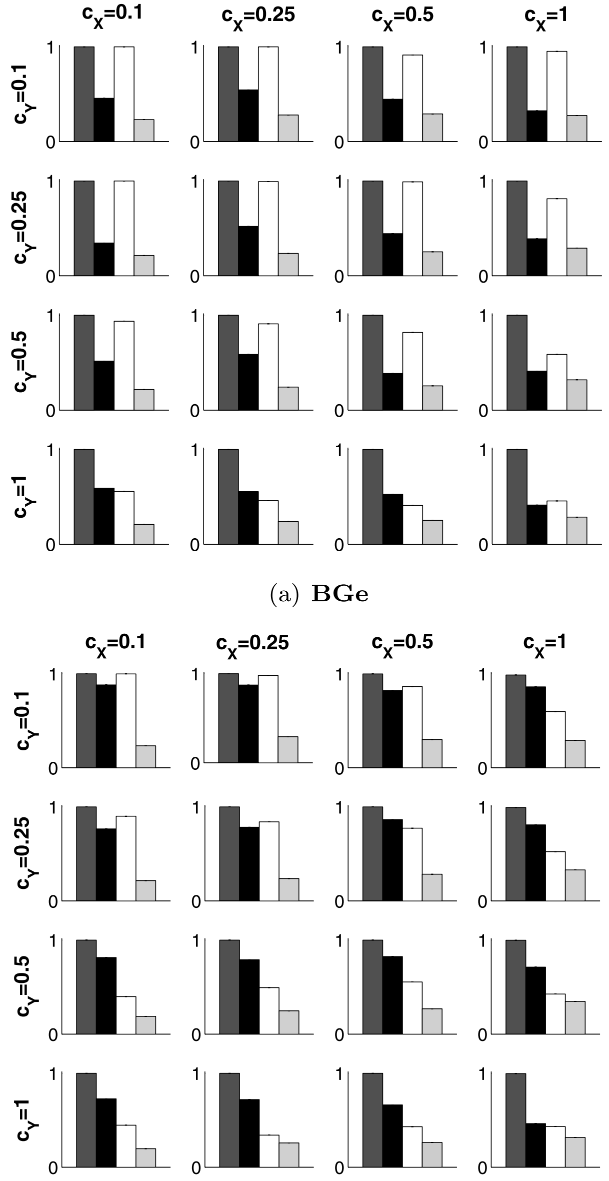
(a) **BGe**

(b) **cpBGe**

Fig. 3 NET 1: Histograms of average marginal edge posterior probabilities. Inference results for the synthetic network NET1 in Table 2. The network shown in Fig. 2a was modelled with (60)–(62), i.e. with a piece-wise linear relationship between X and Y. $\sqrt{1-\varepsilon^2}$ is the autocorrelation of the process $X(t) \to X(t+1)$ and SNR is the signal-to-noise ratio for the interaction $X(t) \to Y(t+1)$. For each parameter combination the average probabilities were obtained from $n_{pc,i} = 25$ independent data instantiations. *Left bar*: $X \to X$ (true self-loop), centre left bar: $X \to Y$ (true edge), centre *right bar*: $Y \to Y$ (spurious self-loop), and right bar: $X \leftarrow Y$ (spurious edge)

Fig. 3a and b, the marginal posterior probability of the true self-feedback loop on X, corresponding to the left-most bar in the histograms, is small, but this is a consequence of the small autocorrelation effect ($\varepsilon = 0.99$), which mean that the true edge strength is very weak (see (69) and Fig. 2a).

In Fig. 4, the posterior probability of the spurious self-feedback loop (white bars in the histogram) is higher than that of the true interaction between the two nodes (black bars in the histogram) when the noise levels are low (panels in the top left corner). This can



(a) **BGe**

(b) **cpBGe**

Fig. 4 NET 5: Histograms of average marginal edge posterior probabilities. Inference results for synthetic network NET5 in Table 2. The network shown in Fig. 2a was modelled with (69), i.e. with a sinusoidal transfer function from X to Y. The noise terms on $X \to X$ and $X \to Y$ increase with c_X and c_Y, respectively. For each parameter combination the average probabilities were obtained from $n_{pc,i} = 25$ independent data instantiations. *Left bar*: $X \to X$ (true self-loop), *centre left bar*: $X \to Y$ (true edge), *centre right bar*: $Y \to Y$ (spurious self-loop), and *right bar*: $X \leftarrow Y$ (spurious edge)

be explained from Figs. 1 and 2a. The dependence of $Y(t+1)$ on $Y(t)$ is indirect, via the interactions $X(t-1) \to Y(t)$, $X(t-1) \to X(t)$ and $X(t) \to Y(t+1)$, which means that it is subject to three noise injections. The relationship between $X(t)$ and $Y(t+1)$ is only subject to one noise injection. When the nonlinear relationship is piecewise linear and can hence be learnt exactly, as in Fig. 3, the spurious self-loop $Y(t) \to Y(t+1)$ will be explained away. When the true nonlinear relationship is sinusoidal, as in Fig. 4, then the functional relationship between $X(t)$ and $Y(t+1)$ can only be learnt approximately. For low noise levels, the effect of the approximation error might outweigh the effect of the noise, meaning that despite three noise injections, $Y(t)$ outperforms $X(t)$ as a predictor for $Y(t+1)$. However, Fig. 4 suggests that this scenario is quite rare, and that in the majority of noise scenarios, the marginal posterior probability of the true edge $X \rightarrow Y$ is significantly higher than that of the spurious self-loop $Y \rightarrow Y$. This suggests that the proposed cpBGe model is, overall, successful at suppressing spurious feedback loops.

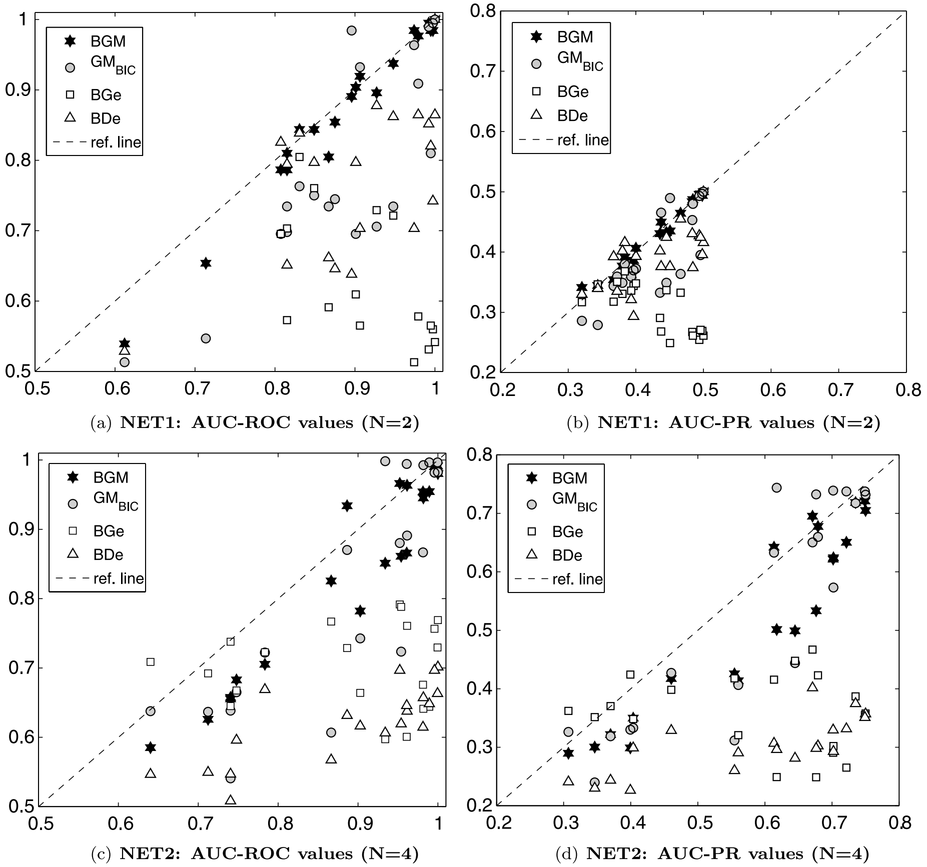
The reason for this reduced susceptibility to spurious feedback loops is improved nonlinear modelling capability. By partitioning the time series into segments, and learning separate parameters (or distributions of parameters) for the different segments, the proposed model is effectively a piecewise linear model. What distinguishes it from a proper piecewise linear model is the fact that the partitioning is carried out in the time domain, not in the domain of explanatory variables. Consider the interaction $X(t-1) \rightarrow Y(t)$. A proper piecewise linear model would partition the space of $X(t-1)$, whereas our model partitions the time domain, t. If the regulatory signal $X(t)$ is sufficiently smooth such that closeness in time implies closeness in $X(t)$ space, then the proposed non-homogeneous model is effectively a piecewise linear model, resulting in efficient nonlinear modelling capability.

5.2 Comparative network reconstruction accuracy

We have applied the proposed cpBGe model to the synthetic data described in Sect. 3.1 and Table 2, and we have compared it with four alternative models, as outlined in Sect. 4.1: the two classical homogeneous DBNs based on the BDe and BGe scores; the nonlinear GM_{BIC} model which constitutes the application of the EM algorithm (Dempster et al. 1977) to a node-specific mixture model subject to a BIC penalty term (Schwarz 1978); and the Bayesian Gaussian mixture (BGM) Bayesian network model (Grzegorczyk et al. 2010). For details of the implementation of these methods, see Sect. 4.1. An overview of these five Bayesian network models can be found in Table 4.

Figures 5, 6 and 7 show a comparative evaluation of the reconstruction accuracy on synthetic data generated from the four networks depicted in Fig. 2. Each figure contains two panels, corresponding to different scoring schemes. The left panel compares areas under the ROC curves; the right panel compares areas under the precision-recall curves. In each plot, the horizontal axis represents the scores of the proposed cpBGe model. The vertical axis represents the scores of the four alternative schemes, identified by different symbols. Symbols that lie above the diagonal dashed line indicate that the proposed cpBGe scheme performs poorer than the alternative method. Symbols that lie below the diagonal dashed line indicate that the proposed cpBGe scheme performs better than the alternative method. Hence, Figs. 5, 6 and 7 suggest that the proposed cpBGe model has a clear tendency to outperform the alternative methods. For a quantitative confirmation we have computed the p-values from a paired two-sided t-test, which are shown in Tables 6–9. For the RAF network with $SNR = 0.1$—corresponding to the leftmost clusters in the two panels of Fig. 7—there are no significant differences between the models. This would be expected, as for such a small signal-to-noise ratio, the signal is effectively buried in noise, and no patterns can be discerned. For the other data sets, cpBGe tends to outperform the other models significantly. A separation according to the alternative models reveals the following trend.

Comparison with BDe The proposed cpBGe model consistently outperforms the discrete BDe Bayesian network model. This can be explained by the fact that the BDe model gains nonlinear modelling capability at the price of information loss due to data discretization, whereas the proposed cpBGe model overcomes the restriction of a linear model without the need for data discretization.



(a) **NET1: AUC-ROC values (N=2)**

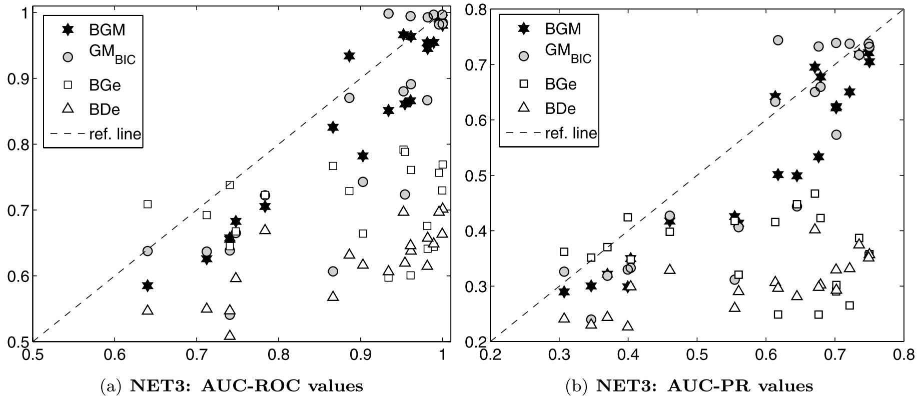
(b) **NET1: AUC-PR values (N=2)**

(c) **NET2: AUC-ROC values (N=4)**

(d) **NET2: AUC-PR values (N=4)**

Fig. 5 Network reconstruction accuracy for the synthetic networks NET1 and NET2 in Table 2. The structure of NET1 with $N = 2$ nodes is shown in Fig. 2a, and the structure of NET2 with $N = 4$ nodes is shown in Fig. 2b. For both domains we implemented an auto-correlated regulator node X that regulates the other node(s) by piece-wise linear functions. For NET1 see (60)–(62) and for NET2 see (63)–(64). $n_{pc} = 20$ parameter combinations $\varepsilon \in \{0.99, 0.5, 0.25, 0.1\}$ and $SNR \in \{100, 10, 3, 1, 0.5\}$ were used to vary the strength of the auto-correlation and the noise in the mutual interactions. To quantify the network reconstruction accuracy we computed the areas under the ROC curves (AUC-ROC) and the areas under the precision-recall curves (AUC-PR). For all $n_{pc} = 20$ parameter combinations $n_{pc,i} = 25$ independent data instantiations were analyzed and the average AUC scores were computed. As a summary of the cross-method comparison the average AUC scores of the 4 competing methods have been plotted against the AUC scores of the proposed cpBGe model. The *diagonal dashed line* indicates equal performance. *Symbols* that lie above this line indicate that the proposed cpBGe scheme performs poorer than the alternative method. *Symbols* that lie below the diagonal dashed line indicate that the proposed cpBGe scheme outperforms the alternative methods. Panels (**a**) and (**c**) show the AUC-ROC score scatter plots and panels (**b**) and (**d**) show the AUC-PR score scatter plots

Comparison with BGe For the RAF network with low signal-to-noise ratio, the linear Gaussian BGe Bayesian network model either outperforms the cpBGe model ($SNR = 0.5$), or shows no significant difference ($SNR = 0.1$, 1.0); see Fig. 7. This suggests that when the signal is buried in noise, a simple linear model shows greater robustness than a more complex one. However, for larger signal-to-noise ratios ($SNR = 3$, 10) and all data generated from the smaller networks—Figs. 5 and 6—the cpBGe model clearly outperforms BGe.



(a) **NET3: AUC-ROC values**

(b) **NET3: AUC-PR values**

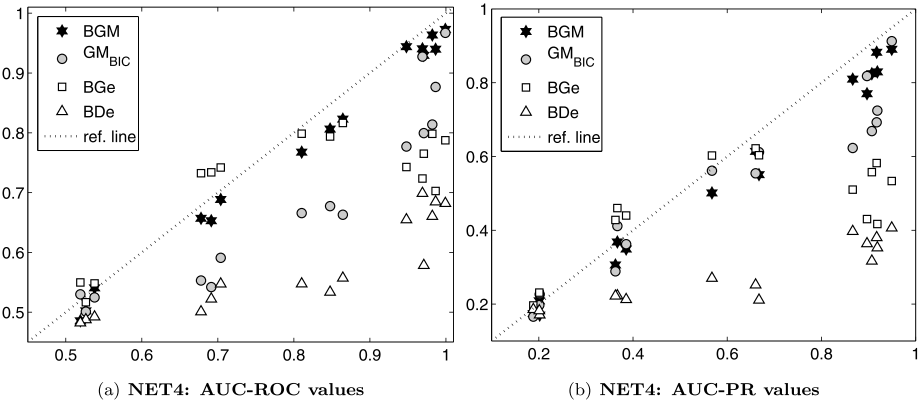
Fig. 6 Network reconstruction accuracy for network NET3 in Table 2. The structure of NET3 with $N = 4$ nodes is shown in Fig. 2c and was modelled with (68). Node Z is regulated by three other nodes X, Y, and W. The edges $X \to Z$ and $Y \to Z$ are implemented as linear functions. Node W is auto-correlated and a sinusoidal transfer function has been implemented for the interaction $X \to Z$. We considered $n_{pc} = 18$ different parameter settings $c_X = c_Y \in \{0.25, 0.5\}$, $c_W, c_Z \in \{0.25, 0.5, 1\}$ and generated $n_{pc,i} = 25$ independent data instantiations for each parameter set. To quantify the learning performance we computed the average areas under the ROC curves (AUC-ROC) and the average areas under the precision-recall curves (AUC-PR) for the $n_{pc} = 18$ parameter sets. In the panels the AUC scores of the 4 competing methods have been plotted against the AUC scores of the proposed cpBGe model. The *diagonal dashed line* indicates equal performance. *Symbols* that lie above this line indicate that the proposed cpBGe scheme performs poorer than the alternative method. *Symbols* that lie below the diagonal dashed line indicate that the proposed cpBGe scheme performs better than the alternative method. Panel (**a**) shows the scatter plot of the AUC-ROC scores and panel (**b**) shows the AUC-PR score scatter plots

Comparison with the GM$_{BIC}$ model The GM$_{BIC}$ model is a Gaussian mixture model with a BIC scoring scheme. The mixture model is more flexible than our changepoint process; the BIC score tends to lead to over-regularization. Our results indicate that the GM$_{BIC}$ model is consistently outperformed by the proposed cpBGe model, except for low signal-to-noise ratios on the RAF network.

Comparison with BGM The Bayesian Gaussian mixture (BGM) Bayesian network model is the closest to the proposed cpBGe model. The difference is that the changepoints are *not* node-specific, but apply to all the nodes in the network jointly. When the network only consists of two nodes, the difference in performance is hardly significant—see the top panels in Fig. 5. However, for networks with a larger number of nodes, the proposed cpBGe model significantly outperforms BGM, unless the signal-to-noise ratio is low.

5.3 Performance of the cpBGe model on data from homogeneous processes and processes where changepoints are common to all nodes

In this section we investigate how the proposed cpBGe Bayesian network model (see Sect. 2.2) compares with the competing models (see Table 4) on homogeneous network data (S1), and on non-homogeneous data where all changepoints are tied together (S2). Data were generated from the RAF-network shown in Fig. 2d as explained in Sect. 3.2. Figure 8 shows a comparative evaluation of the network reconstruction accuracy on synthetic data generated from the RAF-pathway. The figure contains four panels, corresponding to different types of



(a) **NET4: AUC-ROC values** (b) **NET4: AUC-PR values**

Fig. 7 Cross-method comparison of the network reconstruction accuracy for the RAF pathway (NET4) in Sect. 3.1. The RAF pathway with $N = 11$ nodes is shown in Fig. 2d and was modelled with (65)–(66). In our implementation PIP2 is auto-correlated and the other interactions are described by piece-wise linear functions with different signal-to-noise ratios (SNR). We considered $n_{pc} = 15$ parameter combinations (ε, SNR) with $\varepsilon \in \{0.5, 0.25, 0.1\}$ and $SNR \in \{10, 3, 1, 0.5, 0.1\}$ and generated $n_{pc,i} = 5$ independent data instantiations for each combination. To quantify the network reconstruction accuracy we computed the average areas under the ROC curves (AUC-ROC) and the average areas under the precision-recall curves (AUC-PR) for each of the $n_{pc} = 15$ parameter combinations. In the panels the AUC scores of the 4 competing methods have been plotted against the AUC scores of the proposed cpBGe model. The *diagonal dashed line* indicates equal performance. *Symbols* that lie above this line indicate that the proposed cpBGe scheme performs poorer than the alternative method. *Symbols* that lie below the diagonal dashed line indicate that the proposed cpBGe scheme outperforms the alternative method. Panels (**a**) shows the scatter plot of the AUC-ROC scores and panel (**b**) shows the AUC-PR score scatter plot

Table 6 Cross-method comparison of network reconstruction accuracy on the synthetic network data in terms of AUC-ROC values

cpBGe vs. ...	NET1	NET2	NET3	NET4
... vs. BGM	0.002	<0.001	<0.001	0.0394
... vs. GM$_{BIC}$	<0.001	<0.001	<0.001	<0.001
... vs. BGe	<0.001	<0.001	<0.001	0.021
... vs. BDe	<0.001	<0.001	<0.001	<0.001
sample size	$n_{pc} = 20$	$n_{pc} = 20$	$n_{pc} = 18$	$n_{pc} = 15$

For a summary of the network structures and regulatory relationships see Table 2. An overview of the five models is given in Table 4. For each network the average areas under the receiver operator characteristic curve (ROC) of the five different DBN models can be compared in terms of two-sided paired t-test p-values. We have tested for each network and each of the competing methods whether the average AUC-ROC scores for the n_{pc} parameter settings differ from the average AUC-ROC score of the proposed cpBGe model. That is, the p-values quantify for each competing method to what extent its n_{pc} average AUC-ROC points in Fig. 5a (NET1), Fig. 5c (NET2), Fig. 6a (NET3) or Fig. 7a (NET4) deviate from the diagonal reference line. Note that the signs of all t-statistics are in favour of the proposed cpBGe model

data (rows) and scoring schemes (columns). The left panels (a) and (c) compare the areas under the ROC curves; the right panels (b) and (d) compare areas under the precision-recall curves. In each plot, the horizontal axis represents the scores of the proposed cpBGe model. The vertical axis represents the scores of the four alternative schemes, identified by different symbols. Symbols that lie above (below) the diagonal dashed line indicate that the proposed

Table 7 Cross-method comparison of network reconstruction accuracy on the synthetic network data in terms of AUC-PR values

cpBGe vs. …	NET1	NET2	NET3	NET4
… vs. BGM	0.243	<0.001	<0.001	0.040
… vs. GM$_{BIC}$	<0.001	<0.001	<0.001	0.024
… vs. BGe	<0.001	<0.001	<0.001	0.029
… vs. BDe	<0.001	<0.001	<0.001	<0.001
sample size	$n_{pc} = 20$	$n_{pc} = 20$	$n_{pc} = 18$	$n_{pc} = 15$

The p-values quantify for each competing method to what extent its n_{pc} average AUC-PR points in Fig. 5b (NET1), Fig. 5d (NET2), Fig. 6b (NET3) or Fig. 7b (NET4) deviate from the diagonal reference line. Note that the signs of all t-statistics are in favour of the proposed cpBGe model. See caption of Table 6 for details

Table 8 Cross-method comparison of network reconstruction accuracy for the RAF pathway (NET4) in terms of AUC-ROC values

cpBGe vs. …	$SNR = 0.1$	$SNR = 0.5$	$SNR = 1$	$SNR = 3$	$SNR = 10$
… vs. BGM	0.152	0.096	0.032	0.013	0.003
… vs. GM$_{BIC}$	0.512	<0.001	<0.001	<0.001	0.004
… vs. BGe	*0.540*	*0.004*	0.159	<0.001	<0.001
… vs. BDe	0.048	<0.001	<0.001	<0.001	<0.001
sample size	15	15	15	15	15

For each SNR value there are $\sum_{\varepsilon \in \{0.1, 0.25, 0.5\}} n_{pc,SNR,\varepsilon} = 15$ AUC-ROC values for each model. These values can be compared in terms of two-sided paired t-test p-values. We have tested for the five SNRs and for each of the four competing methods (see Table 4) whether its average AUC-ROC value differs from the average AUC-ROC score of the proposed cpBGe model. P-values of t-statistics that were in favour of the competing method are indicated in *italics*

Table 9 Cross-method comparison of network reconstruction accuracy for the RAF pathway (NET4) in terms of AUC-PR values

cpBGe vs. …	$SNR = 0.1$	$SNR = 0.5$	$SNR = 1$	$SNR = 3$	$SNR = 10$
… vs. BGM	0.254	0.100	0.077	0.002	<0.001
… vs. GM$_{BIC}$	0.969	0.641	0.169	<0.001	0.004
… vs. BGe	*0.164*	*0.002*	0.800	<0.001	<0.001
… vs. BDe	0.216	<0.001	<0.001	<0.001	<0.001
sample size	15	15	15	15	15

For each SNR value there are $\sum_{\varepsilon \in \{0.1, 0.25, 0.5\}} n_{pc,SNR,\varepsilon} = 15$ AUC-PR values for each model. These values can be compared in terms of two-sided paired t-test p-values. We have tested for the five SNRs and for each of the four competing methods (see Table 4) whether its average AUC-PR value differs from the average AUC-PR score of the proposed cpBGe model. P-values of t-statistics that were in favour of the competing method are indicated in *italics*

cpBGe scheme performs poorer (better) than the alternative method. Overall, Fig. 8 suggests that the proposed cpBGe model and the BGM model perform approximately equally well and better than the alternative methods BDe, BGe and GM$_{BIC}$. For a quantitative con-

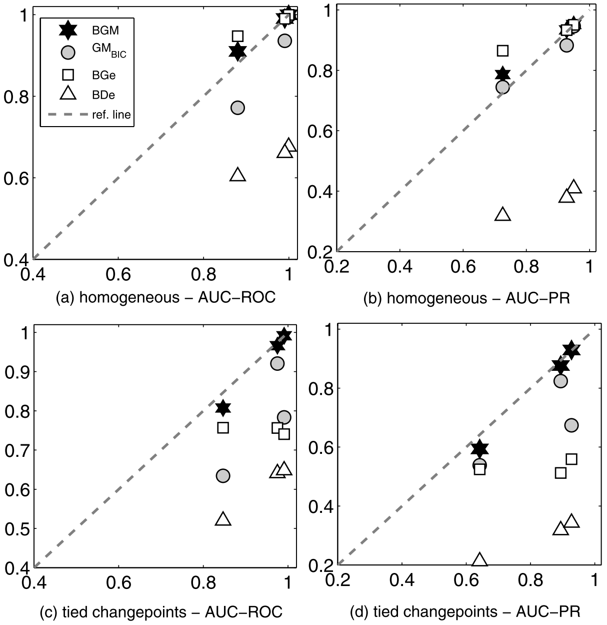


Fig. 8 Cross-method comparison of the network reconstruction accuracy for two different scenarios: (S1): homogeneous network data (*top row*) and (S2): non-homogeneous network data with tied changepoints (*bottom row*). The RAF pathway (NET4) is shown in Fig. 2d and was modelled as explained in Sect. 3.2. That is, the homogeneous data (S1) were generated using (65) and (71), and the tied changepoint scenario (S2) was generated using (65)–(66) *under the constraint* that each changepoint applies to all nodes. We fixed $\varepsilon = 0.25$ and considered $n_{pc} = 3$ signal-to-noise ratios, $SNR \in \{10, 3, 1\}$. For each SNR we generated $n_{pc,i} = 5$ independent data instantiations. To quantify the network reconstruction accuracy we computed the average areas under the ROC curves (AUC-ROC) and the average areas under the precision-recall curves (AUC-PR) for each of the $n_{pc} = 3$ parameter combinations. In the panels the AUC scores of the 4 competing methods have been plotted against the AUC scores of the proposed cpBGe model. The *diagonal dashed line* indicates equal performance. *Symbols* that lie above this line indicate that the proposed cpBGe scheme performs poorer than the alternative method. *Symbols* that lie below the *diagonal dashed line* indicate that the proposed cpBGe scheme outperforms the alternative method. The *left column* shows the scatter plots of the AUC-ROC scores and the *right column* shows the AUC-PR score scatter plots

firmation we have computed the p-values from a paired two-sided t-test, which are shown in Tables 10–11. A separation according to the alternative models reveals the following trends:

Comparison with BDe Similar to the results observed for the non-homogeneous data with node-specific changepoints (see Sect. 5.2), the proposed cpBGe model consistently outperforms the discrete BDe model for homogeneous data and non-homogeneous data with tied changepoints. It appears that the nonlinear modelling capability of the BDe model does *not* compensate the information loss associated with the data discretization. The BDe model performs consistently worse than the four other methods that analyse the continuous data.

Comparison with BGe For the homogeneous data there is no significant difference between the performance of the cpBGe model and the BGe model if the signal is stronger than noise ($SNR = 3$ and $SNR = 10$). Only when the signal is weak ($SNR = 1$), does the BGe model perform significantly better than the cpBGe model. This suggests that the cpBGe

Table 10 Cross-method comparison of network reconstruction accuracy for the RAF pathway (NET4) with homogeneous regulatory mechanisms in terms of AUC-ROC and AUC-PR values

Criterion	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR
SNR	10	10	3	3	1	1
cpBGe vs. …						
… vs. BGM	*0.237*	*0.227*	*0.700*	*0.098*	*0.218*	*0.167*
… vs. GM_{BIC}	0.459	0.589	<0.001	0.004	<0.001	*0.560*
… vs. BGe	*0.244*	*0.229*	0.816	*0.067*	*0.003*	*0.002*
… vs. BDe	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
Sample size	10	10	10	10	10	10

For each SNR value there are $n_{pc} = 10$ AUC-ROC and AUC-PR values for each model. These values can be compared in terms of two-sided paired t-test p-values. We have tested for the three SNRs and for each of the four competing methods (see Table 4) whether its average AUC-ROC (AUC-PR) value differs from the average AUC-ROC (AUC-PR) score of the proposed cpBGe model. P-values of t-statistics that were in favour of the competing method are indicated in *italics*

Table 11 Cross-method comparison of network reconstruction accuracy for the RAF pathway (NET4) with tied changepoints in terms of AUC-ROC and AUC-PR values

Criterion	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR
SNR	10	10	3	3	1	1
cpBGe vs. …						
… vs. BGM	0.503	0.467	*0.897*	*0.916*	0.039	0.145
… vs. GM_{BIC}	0.067	0.180	<0.001	<0.001	<0.001	0.021
… vs. BGe	<0.001	<0.001	<0.001	<0.001	<0.001	0.008
… vs. BDe	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
Sample size	10	10	10	10	10	10

For each SNR value there are $n_{pc} = 10$ AUC-ROC and AUC-PR values for each model. These values can be compared in terms of two-sided paired t-test p-values. We have tested for the three SNRs and for each of the four competing methods (see Table 4) whether its average AUC-ROC (AUC-PR) value differs from the average AUC-ROC (AUC-PR) score of the proposed cpBGe model. P-values of t-statistics that were in favour of the competing method are indicated in *italics*

model does not infer spurious changepoints for homogeneous data unless there is a high amount of noise in the data. For the non-homogeneous data with tied changepoints (see bottom row in Fig. 8) the results are similar to the results obtained in Sect. 5.2. That is, the proposed cpBGe model outperforms the BGe model, which is linear and therefore cannot deal with non-homogeneous data, independently of whether changepoints are node-specific or tied together.

Comparison with the GM_{BIC} model The GM_{BIC} model is consistently outperformed by the proposed cpBGe model, except for the low signal-to-noise ratio ($SNR = 1$) and the AUC-PR scoring scheme (see panel (b) in Fig. 8) where both methods perform equally well. This finding is also consistent with the finding obtained for non-homogeneous data with node specific changepoints (see Sect. 5.2). It appears that the BIC score, on which the GM_{BIC} model is based, leads to over-regularization for all types of data.

Comparison with BGM For homogeneous data (top row in Fig. 8) the BGM Bayesian network model and the proposed cpBGe model show a similar performance. It appears that the BGM model yields slightly higher scores than the cpBGe model, but there is no significant difference (see Table 10). For the non-homogeneous data with tied changepoints (bottom row in Fig. 8) there is no significant difference between the BGM model and the proposed cpBGe model either. On the contrary, it appears that there is a trend towards the cpBGe model for the small signal-to-noise ratio ($SNR = 1$). This finding is surprising, since the data have been generated in a way that is consistent with the BGM model; the additional flexibility (i.e. the node-specificity of changepoints) of the proposed cpBGe model is *not* required. We are therefore investigating that in more detail.

The results of the cross-method comparison between the proposed cpBGe model and the BGe, the BDe, and the GM_{BIC} model for homogeneous and non-homogeneous data with tied changepoints yields results that are comparable to those obtained for non-homogeneous data with node-specific changepoints. But different from our expectation, we did not observe a significant difference between BGM and cpBGe for non-homogeneous data with tied changepoints. We therefore had a closer look at the two models BGM and cpBGe. In the cpBGe (BGM) model we replaced the original truncated Poisson prior distribution on the number of changepoints $\mathcal{K}_n$ ($\mathcal{K}$) and the prior on the changepoint locations $(\mathbf{V}_n|\mathcal{K}_n)$ $((\mathbf{V}|\mathcal{K}))$ given in (11) by a point process prior on the distances between changepoints (see (21)–(24)). The point process prior is based on "waiting times" between changepoints, which are distributed according to a negative binomial distribution. The probability mass function of the negative binomial distribution $NBIN(p,k)$ is given in (23) and possesses two (hyper-)parameters p and k. After having replaced the prior distribution, we can vary the prior penalty that is associated with changepoints. We fixed $k = 2$ and varied $p \in \{10^{-i} : i = 1, \dots, 6\}$ where higher (lower) values of the hyperparameter p imply lower (higher) prior penalties for changepoints. With the modified models BGM and cpBGe (see Table 4) we re-analysed all RAF-pathway data sets. That is, the data with node-specific changepoints (from Sect. 3.1) as well as the data from the two additional scenarios (Sect. 3.2) were re-analysed with the BGM and the cpBGe model using six different hyperparameters p. Figure 9 summarizes the empirical results of this simulation study. The following trends can be observed for the three different types of data:

(1) *Homogeneous data*: From the top row in Fig. 9 it can be seen that the three Bayesian network models perform equally well for homogeneous data if the signal is stronger than noise ($SNR = 3$ and $SNR = 10$). For the low signal-to-noise ratio $SNR = 1$ the two non-homogeneous Bayesian networks models BGM and cpBGe are outperformed by the homogeneous BGe model for high settings of the hyperparameter p ($p \geq 10^{-2}$ (BGM) and $p \geq 10^{-1}$ (cpBGe)). Recalling that high parameters p imply low prior penalties for changepoints, this phenomenon can be explained by over-fitting. Low hyperparameters p yield an insufficient regularization of the model complexity of BGM and cpBGe.

(2) *Non-homogeneous data with tied changepoints*: The centre row in Fig. 9 suggests that the two non-homogeneous models perform consistently and substantially better than the BGe model. For $SNR = 3$ and $SNR = 10$ the BGM model outperforms the proposed cpBGe model for low hyperparameters p ($p \leq 10^{-4}$), but the network reconstruction accuracy becomes equal for higher hyperparameters p. This suggests that the BGM model is superior to the cpBGe model if a high prior penalty for changepoints is employed. An explanation is that the overall prior penalty is higher for the cpBGe model, as the changepoints have to be learnt individually and are individually penalized by the prior. Since the original prior distribution approximately corresponds to $p = 0.5 \times 10^{-2}$ (see Sect. 4.3 for details), these findings are consistent with those shown in Fig. 8. For the low signal-to-noise ratio (centre right panel) the trend is different: The proposed cpBGe model outperforms the BGM

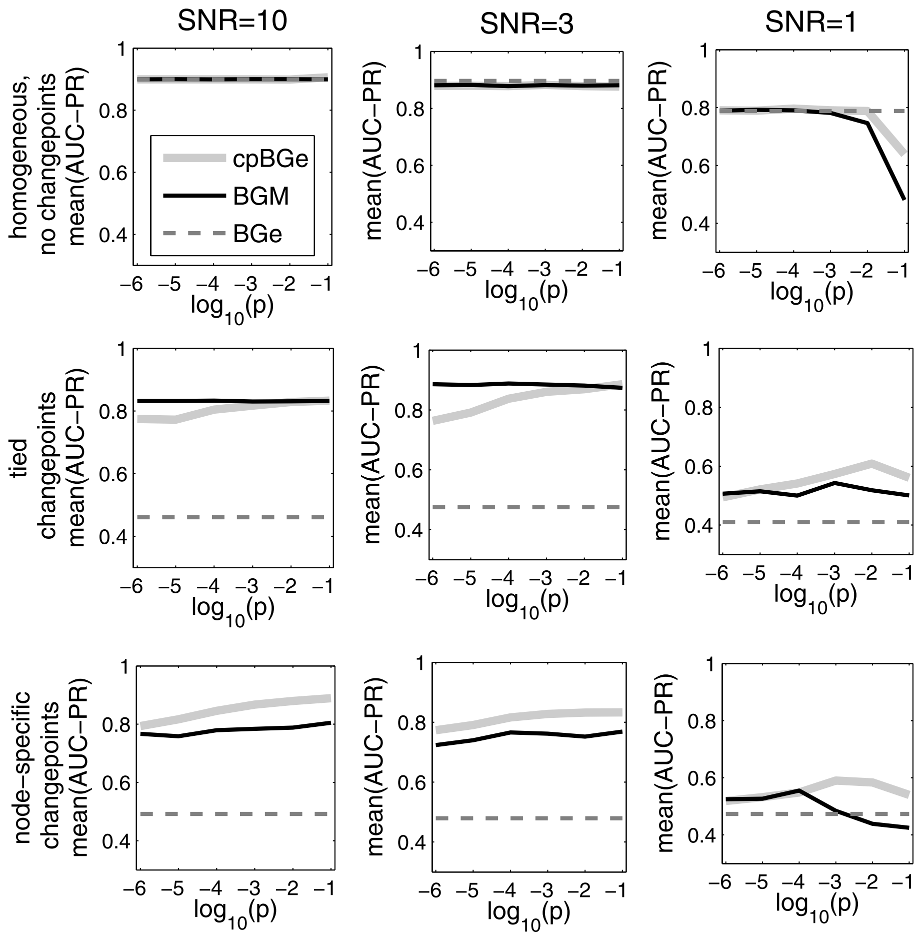


Fig. 9 Network reconstruction accuracy for different types of data. The figure shows the mean area under the precision-recall curves (AUC) in dependence on the hyperparameter p of the negative binomial point process prior, as defined in (21)–(24). Higher values of the hyperparameter p imply lower prior penalties for changepoints. For the RAF pathway in Fig. 2d we implemented three different regulatory mechanisms: homogeneous data without changepoints (*top row*), non-homogeneous data with tied changepoints (*centre row*), and non-homogeneous data with node-specific changepoints (*bottom row*). Details on these scenarios can be found in Sects. 3.1 and 3.2. For each scenario there are three panels for $SNR = 10$ (*left column*), $SNR = 3$ (*centre column*), and $SNR = 1$ (*right column*). The following models were applied to the data: (i) the BGe model as reference model, (ii) the BGM model, and (iii) the proposed cpBGe model. The BGe model has no changepoints and is therefore independent of p. The mean AUC-PR scores were computed from 5 independent data instantiations

model. This can be explained as follows: For $SNR = 1$ both models BGM and cpBGe tend to infer spurious changepoints. For the BGM model these spurious changepoints apply to all nodes, and the global network reconstruction accuracy is weakened. For the cpBGe model the (spurious) changepoints are node-specific, and thus have only a limited effect on the global network reconstruction accuracy.

(3) *Non-homogeneous data with node-specific changepoints*: From the bottom row in Fig. 9 it can be seen that the two non-homogeneous models perform consistently and sub-

stantially better than the BGe model. Moreover, the cpBGe model is consistently superior to the BGM model except for $SNR = 1$ and low hyperparameters p (see bottom right panel). This finding is in consistency with those results obtained for the RAF-pathway in Sect. 5.2.

Our findings can be summarized as follows. The proposed cpBGe model consistently outperforms BDe and the approach based on the BIC score (GM_{BIC}). When generating data that are consistent with the BGe model (homogeneous, no changepoints), cpBGe is only outperformed by the BGe model when the prior penalty for changepoints and the SNR are very low; this is the scenario where model overflexibility is most susceptible to overfitting. When the data are consistent with the BGM model (non-homogeneous, tied changepoints), the cpBGe and BGM models show a similar performance, except for the following extreme scenarios. When the prior penalty for changepoints is very high, BGM performs better; this is a consequence of the fact that for cpBGe a separate prior penalty for each node-specific changepoint has to paid, and the overall regularization effect of the prior becomes too strong. When both the SNR and the prior penalty are very low, cpBGe outperforms BGM. This is a consequence of the fact that low SNRs render more complex models more susceptible to overfitting, and spurious changepoints have a stronger effect for the BGM than the cpBGe model (because they simultaneously affect all nodes rather than specific target nodes). For data based on node-specific changepoint processes, cpBGe outperforms all other models. Hence, the overall conclusion from our study is that for the latter data, using the cpBGe model is an advantage, while for data consistent with less complex models, applying the cpBGe model is in general no disadvantage.

5.4 Convergence and mixing of the MCMC samplers

We have assessed the degree of convergence and mixing of the MCMC simulations by computing the potential scale reduction factor (PSRF) from the marginal posterior probabilities of the edges. Figures 10–12 show the proportion of edges for which a target convergence level has been reached, for four target levels of PSRF < 1.2, PSRF < 1.1, PSRF < 1.05 and PSRF < 1.02. Note that smaller values indicate a better degree of convergence, with a value of PSRF< 1.1 usually taken as an indication of "sufficient" convergence.

Figures 18–20 in Appendix E offer a complementary representation, which show scatter plots of the marginal posterior probabilities of the edges, as obtained from two different MCMC simulations. Here, a better agreement between these simulations, i.e. a location of the entries closer to the diagonal line, indicates a better convergence. We compared five MCMC schemes, as described in Sect. 2 and Table 5: RJMCMC with standard structure MCMC, RJMCMC with structure MCMC plus parent exchange (flip) move, and three Gibbs sampling schemes based on dynamic programming, in which both the parent configurations as well as the changepoint locations are sampled from the correct conditional distributions. Note that computing the conditional probabilities of the parent configurations according to (20) as well as sampling the changepoints via the dynamic programming schemes is computationally involved, and we have tried to approximately match the computational costs, as described in Sect. 4.2 and Table 5. We used three different dynamic programming schemes. As described in Sects. 2.7.1 and 2.7.2, the principal difference is in the choice of prior distribution for the changepoints. We used both a conditional distribution based on the number of changepoints, and a distribution based on a point process for the difference between change points. In the former case, we used two different cut-off values for the maximum number of components: $\mathcal{K}_{MAX} \leq 10$ and $\mathcal{K}_{MAX} \leq 5$.

We have applied our convergence analysis to three data sets. Figures 10 and 11 show the results obtained for synthetic data generated from the RAF network with different signal-to-noise ratios. Figure 12 shows the results on the circadian gene expression time series from

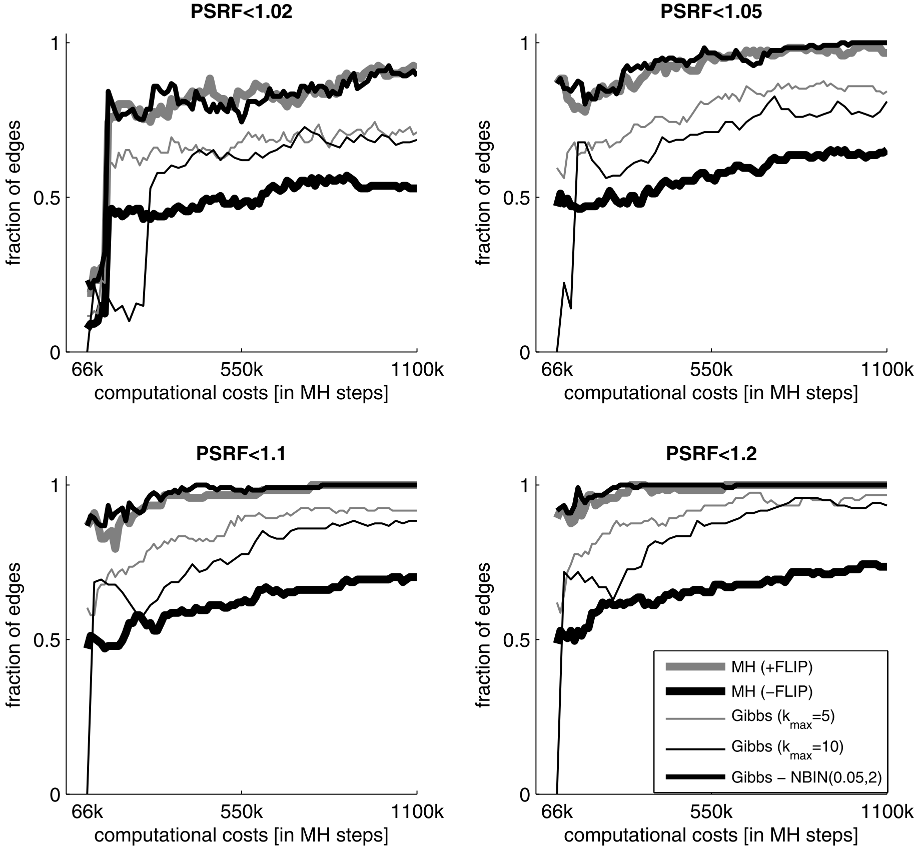


Fig. 10 Convergence diagnostics based on potential scale reduction factors (PSRFs) of individual network edges—RAF network with $SNR = 3$. We compare the five MCMC schemes of Table 5. (i) **MH(−FLIP)**: RJMCMC with standard structure MCMC; (ii) **MH(+FLIP)**: RJMCMC based on structure MCMC with the parent exchange (FLIP) move; (iii) **Gibbs(K = 10)**: Gibbs sampling with dynamic programming, using a prior on the number of components truncated at $\mathcal{K}_{MAX} = 10$; (iv) **Gibbs(K = 5)**: Idem, but truncated at $\mathcal{K}_{MAX} = 5$; (v) **Gibbs-NBIN**: Gibbs sampling with dynamic programming, using a point process prior on the distances between changepoints. All individual edge PSRFs have been computed for one single data instantiation of the RAF pathway (NET4 in Table 2) with $SNR = 3$ and $\varepsilon = 0.25$. For each of the five sampling schemes 10 independent MCMC simulations were performed and a PSRF was computed for each individual edge. Each panel shows overlaid trace plots of the fractions of individual edges whose PSRF was lower than the threshold (1.2, 1.1, 1.05, and 1.02). The computational costs on the horizontal axis are given in Metropolis-Hastings MCMC iterations. The numbers of iterations that can be performed for each of the three Gibbs samplers at the same computational costs are shown in Table 5. Details on how we defined a PSRF for an individual edge can be found in Sect. 4.3

Arabidopsis. A clear outcome of our study is that the conventional structure MCMC scheme leads to very poor convergence. On all data sets, there is considerable scope for improvement, with typically only about 50% of the edges satisfying the convergence criterion when using structure MCMC (without the flip-operator).

Using Gibbs sampling with dynamic programming and a prior distribution on the number of changepoints tends to give an improvement on structure MCMC. At least in two studies—Figs. 10 and 12—the proportion of edges satisfying the convergence criterion significantly increases. The improvement is more pronounced when using the more restrictive prior, with

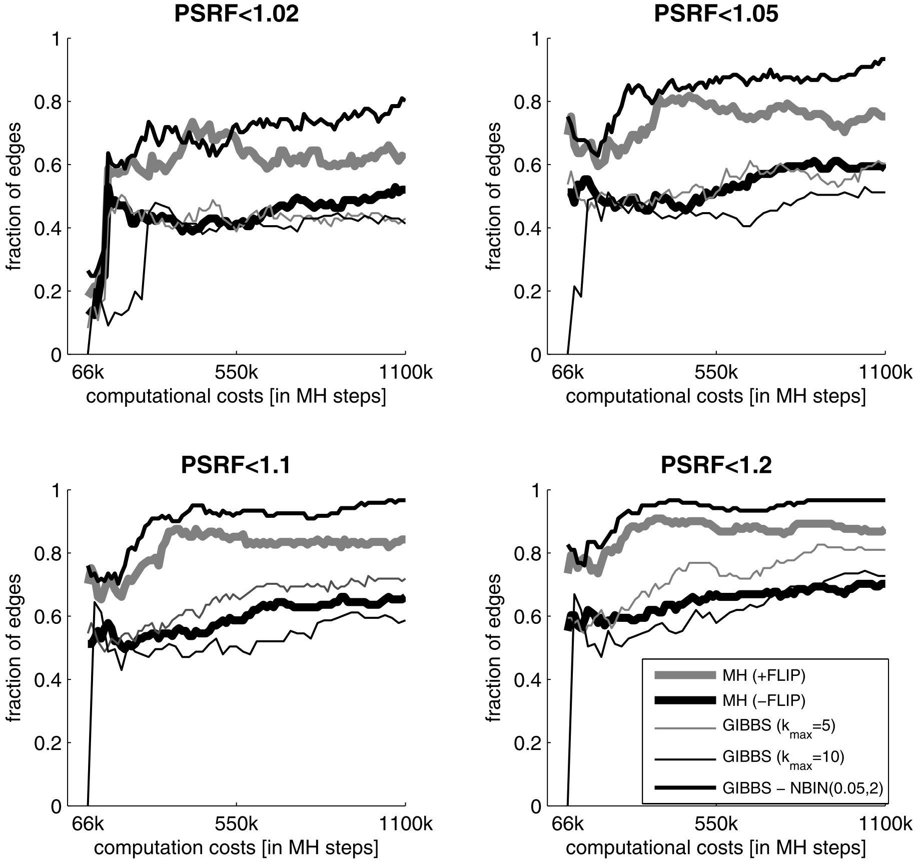


Fig. 11 Convergence diagnostics based on potential scale reduction factors (PSRFs) of individual network edges—RAF network with $SNR = 1$. We compare the five MCMC schemes of Table 5. (i) **MH(−FLIP)**: RJMCMC with standard structure MCMC; (ii) **MH(+FLIP)**: RJMCMC based on structure MCMC with the parent exchange (FLIP) move; (iii) **Gibbs(K = 10)**: Gibbs sampling with dynamic programming, using a prior on the number of components truncated at $\mathcal{K}_{MAX} = 10$; (iv) **Gibbs(K = 5)**: Idem, but truncated at $\mathcal{K}_{MAX} = 5$; (v) **Gibbs-NBIN**: Gibbs sampling with dynamic programming, using a point process prior on the distances between changepoints. All individual edge PSRFs were computed for one single data instantiation of the RAF pathway (NET4 in Table 2) with $SNR = 1$ and $\varepsilon = 0.25$. For each of the five sampling schemes 10 independent MCMC simulations have been performed and a PSRF was computed for each individual edge. Each panel shows overlaid trace plots of the fractions of individual edges whose PSRF was lower than the threshold (1.2, 1.1, 1.05, and 1.02). The computational costs on the horizontal axis are given in Metropolis-Hastings MCMC iterations. The numbers of iterations that can be performed for each of the three Gibbs samplers at the same computational costs are shown in Table 5. Details on how we defined a PSRF for an individual edge can be found in Sect. 4.3

a cut-off of $\mathcal{K}_{MAX} \leq 5$ rather than $\mathcal{K}_{MAX} \leq 10$ on the number of components. This is because a stricter restriction on the number of components/changepoints reduces the computational costs of the dynamic programming scheme, thereby allowing more Gibbs sampling steps to be performed at the same computational costs. Interestingly, structure MCMC with the new parent exchange move leads to a consistent improvement on this Gibbs sampling/dynamic programming scheme. Except for the top left panel in Fig. 12, the proportion of edges satisfying the convergence criterion is always higher with structure MCMC plus parent exchange

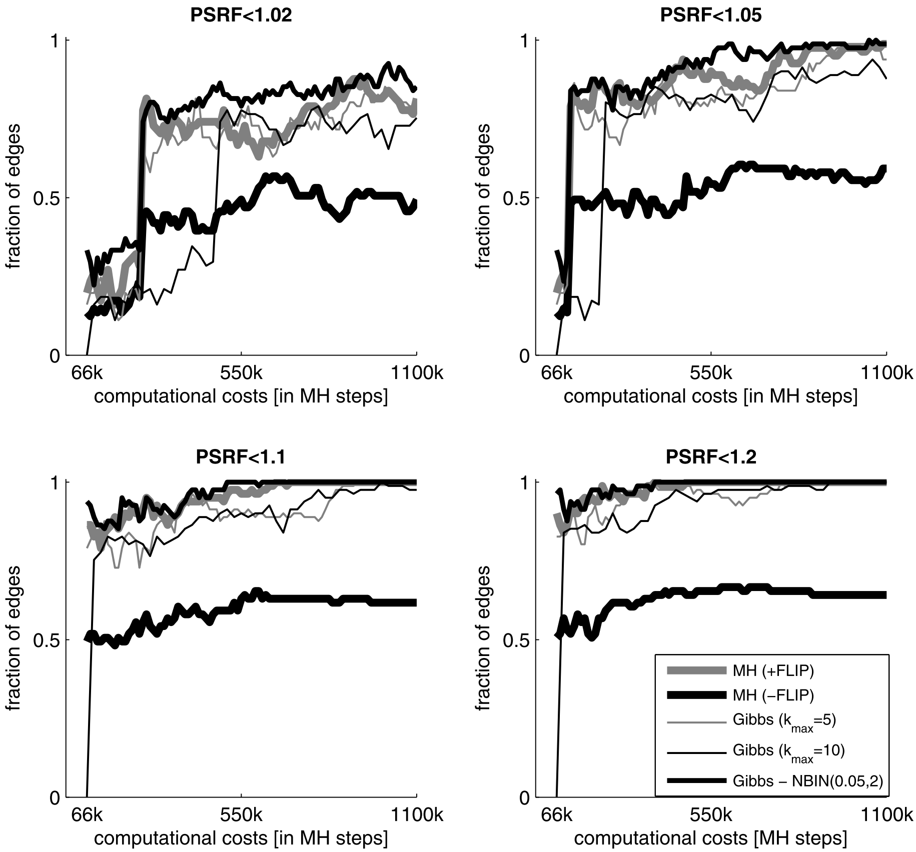


Fig. 12 Convergence diagnostics based on potential scale reduction factors (PSRFs) of individual network edges—*Arabidopsis thaliana* network. We compare the five MCMC schemes of Table 5. (i) **MH(−FLIP)**: RJMCMC with standard structure MCMC; (ii) **MH(+FLIP)**: RJMCMC with structure MCMC plus parent exchange (flip) move; (iii) **Gibbs(K = 10)**: Gibbs sampling with dynamic programming, using a prior on the number of components truncated at $\mathcal{K}_{MAX} = 10$; (iv) **Gibbs(K = 5)**: Idem, but truncated at $\mathcal{K}_{MAX} = 5$; (v) **Gibbs-NBIN**: Gibbs sampling with dynamic programming, using a point process prior on the distances between changepoints. For each sampling scheme 10 independent MCMC simulations were performed on the *Arabidopsis thaliana* data set and a PSRF was computed for each individual edge. Each panel shows overlaid trace plots of the fractions of individual edges whose PSRF was lower than the threshold (1.2, 1.1, 1.05, and 1.02). The computational costs on the horizontal axis are given in Metropolis-Hastings MCMC iterations. The numbers of iterations that can be performed for each of the three Gibbs samplers at the same computational costs are shown in Table 5. Details on how we defined a PSRF for an individual edge can be found in Sect. 4.3

move than when using dynamic programming with a prior on the number of changepoints. This might at first be surprising, but can be explained by the high computational costs of the dynamic programming scheme for sampling new changepoint positions. The solution to this counter-intuitive finding is to use the Gibbs sampling/dynamic programming scheme with a different prior distribution. Rather than imposing a prior on the number of components/changepoints, it is better to use a point process prior on the distances between changepoints. As already pointed out in Fearnhead (2006), the choice of this prior reduces the computational costs of the dynamic programming scheme, and it now turns out that dynamic

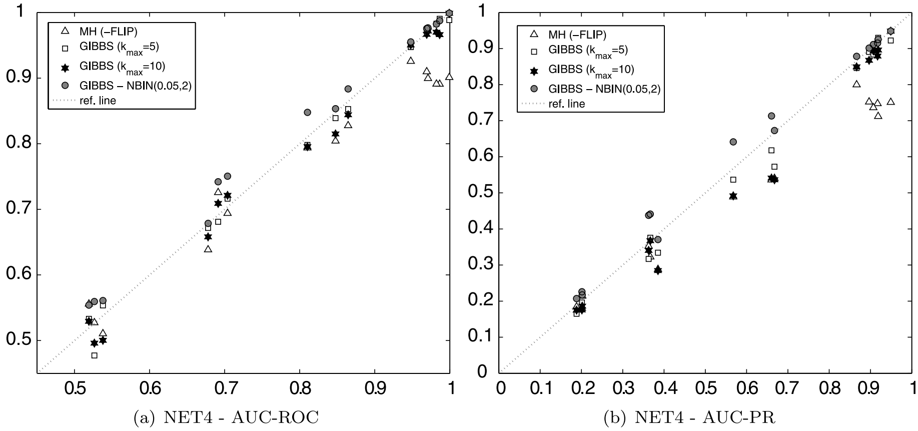


Fig. 13 MCMC sampling scheme comparison: Network reconstruction accuracy for the RAF pathway (NET4) in Table 2. The RAF pathway with $N = 11$ nodes is shown in Fig. 2d and was modelled with (65)–(66). We generated $n_{pc,i} = 5$ independent data instantiations for each of $n_{pc} = 15$ parameter combinations of $SNR \in \{10, 3, 1, 0.5, 0.1\}$ and $\varepsilon \in \{0.1, 0.25, 0.5\}$. For each of the $n_{pc} = 15$ combinations the average network reconstruction accuracy was quantified in terms of the average areas under the ROC curves (AUC-ROC) and the average areas under the precision-recall curves (AUC-PR). In the panels the AUC scores of four alternative MCMC sampling schemes (see Table 5) have been plotted against the AUC scores of the Metropolis-Hastings MCMC sampler improved by the FLIP-operator MH(+FLIP). The diagonal dashed line indicates equal performance. Symbols that lie above this line indicate that the MH(+FLIP) sampler performs poorer than the alternative sampler. Symbols that lie below the diagonal dashed line indicate that MH(+FLIP) performs better than the alternative sampler. The numbers of MCMC iterations that were performed with the different MCMC sampling schemes can be found in Table 5. An explanation of the notation in the legend can be found in Table 5, which contains an overview of the five MCMC schemes compared. Panel (**a**) shows the scatter plot of the AUC-ROC scores and Panel (**b**) shows the AUC-PR scores scatter plots

programming with Gibbs sampling does lead to a further improvement in convergence. This improvement on structure MCMC with parent exchange moves varies in its degree, though: negligible on the synthetic data with high signal-to-noise ratio (Fig. 10), marginal on the circadian gene expression data from Arabidopsis (Fig. 12), and noticeable on the synthetic data with low signal-to-noise ratio (Fig. 11). This suggests that the inclusion of the parent exchange move is attractive for practical applications, as it is easy and straightforward to implement, leads to a substantial convergence improvement on conventional structure MCMC, and often converges to a similar degree as a full-blown Gibbs sampling/dynamic programming scheme. As a complementary study, we have investigated to what extent the choice of MCMC scheme influences the network reconstruction accuracy. To this end, we have applied all five MCMC schemes at equal computational costs to all synthetic data generated from the RAF network and computed the areas under the ROC and precision-recall curves. The results are shown in Figure 13. The horizontal axis represents the scores obtained with the new structure MCMC scheme with parent exchange moves as a reference, whereas the vertical axis represents the scores obtained with the alternative MCMC schemes. We have carried out a quantitative significance estimation based on a paired two-sided t-test; the p-values from this test can be found in Table 12.

The results confirm the trends observed for the convergence plots. Including the parent exchange move leads to a significant improvement over conventional MCMC. The novel structure MCMC plus parent exchange move scheme tends to outperform Gibbs sampling with dynamic programming when a prior on the number of components/changepoints is

Table 12 Comparison of the network reconstruction accuracy among different MCMC sampling schemes for the cpBGe model on the RAF pathway (NET4)

MH(+FLIP) vs. ...	MH	Gibbs	Gibbs	Gibbs
	(−FLIP)	$K = 10$	$K = 5$	NBIN(0.05, 2)
AUC-ROC	0.046	0.258	0.608	*0.041*
AUC-PR	0.048	0.011	0.142	*0.065*

The $n_{pc} = 15$ average AUC-ROC and AUC-PR can be compared in terms of two-sided paired t-test p-values. The table gives the p-values for a comparison of the standard Metropolis-Hastings sampler improved by the flip operator MH(+FLIP) and the four competing sampling schemes. An explanation of the notation in the top row can be found in Table 5, which provides an overview of the five MCMC schemes compared. The p-values quantify to what extent the n_{pc} average AUC-ROC and AUC-PR points in Figs. 13a and b deviate from the diagonal reference line. P-values of t-statistics that were in favour of the competing method are indicated in *italics*

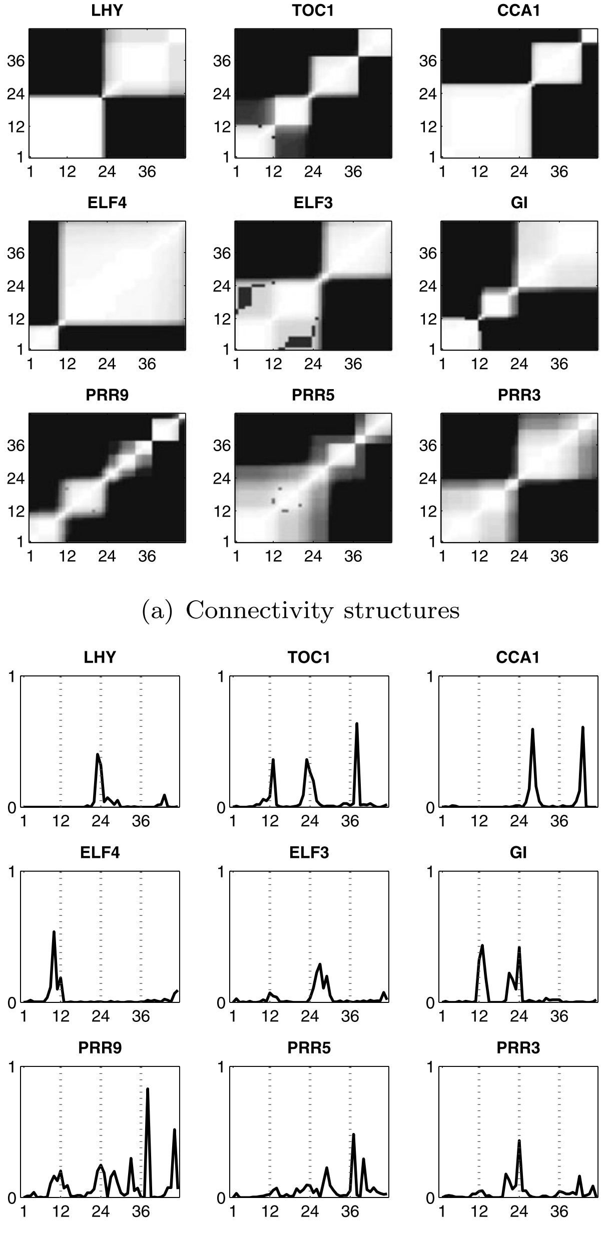
used, although the difference in performance tends to be not significant. However, when using Gibbs sampling/dynamic programming with a point process prior on the distances between changepoints, the network reconstruction accuracy further improves, and this improvement is significant.

We have carried out various additional simulations with hybrid samplers, which mix the Gibbs sampler and the RHJMCMC method with different mixing proportions. The results of our study are presented in Appendix F. Our findings suggest that the hybrid method does not lead to any improvement in mixing or convergence. This result does not seem to be surprising, as the hybrid approach combines a more effective sampler (Gibbs sampler) with a less effective one (RJMCMC). Compared to the former method, this does not reduce the autocorrelation between subsequent samples and hence does not improve the mixing/convergence of the Markov chain.

5.5 Application to circadian microarray gene expression data from Arabidopsis

We have applied our method to microarray gene expression time series related to the study of circadian regulation in plants. A description of the data is found in Sect. 3.3. We have focused our analysis on 9 circadian genes: LHY, TOC1, CCA1, ELF4, ELF3, GI, PRR9, PRR5, and PRR3. The aim is to integrate four gene expression time series, which differed with respect to the pre-experiment entrainment condition; see Table 3 for details. The ideal approach would be to use a supervised approach, as described in Werhli and Husmeier (2008), and use the knowledge we have about the experimental conditions for data segmentation. However, we elected to use these data as a test case for evaluating the efficiency of the proposed cpBGe model.

We therefore combined all four time series into a single set, and applied the proposed method to segment the resulting data in an unsupervised, node-specific manner. The objective was to test whether the proposed cpBGe model would detect the different experimental conditions. Since the gene expression values at the first time point of a time series segment have no relation with the expression values at the last time point of the preceding segment, the corresponding boundary time points were appropriately removed from the data; see Sect. 3.3 and Appendix A for a proper mathematical treatment. 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. Figure 14 shows the marginal posterior probability of the changepoint locations (panel a), and the posterior probability of



(a) Connectivity structures

(b) Transition probabilities

Fig. 14 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 3)

the co-allocation of two time points to the same component (panel b). 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 non-homogeneities beyond

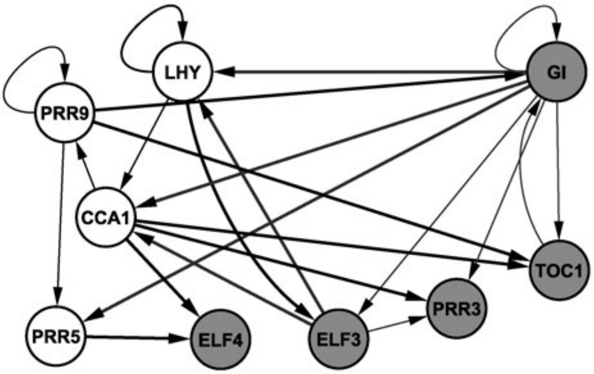


Fig. 15 Circadian gene regulatory network in Arabidopsis learnt from gene expression time series. Predicted regulatory network of nine circadian genes in *Arabidopsis thaliana*. Empty circles represent morning genes. *Shaded circles* represent evening genes. *Edges* indicate predicted interactions with a marginal posterior probability greater than 0.5

the four experiments. 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 3 with the locations of the peaks in Fig. 14 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 15 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) represses TOC1 and potentially other evening genes, and Kikis et al. (2005) report that CCA1 (and LHY) acts negatively on ELF4 expression. Our reconstructed network also contains edges pointing in 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 → CCA1 and ELF3 → LHY in Fig. 15 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 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 GI ↔ TOC1 between GI and 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 → 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.

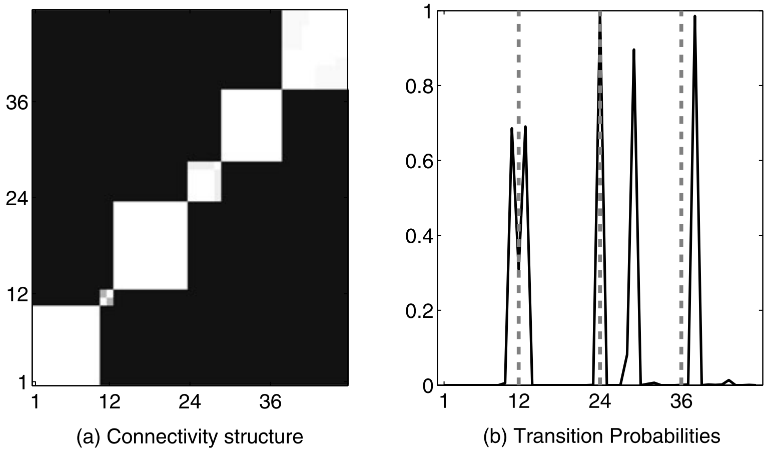


Fig. 16 Results on the Arabidopsis gene expression time series obtained with the BGM model. Panel (**a**): Co-allocation matrix for the nine circadian genes. The axes represent time. The grey shading indicates the posterior probability of two time points being assigned to the same mixture component, 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 3)

The Arabidopsis data have been obtained by merging four time series of gene expression data, which have been measured under different experimental conditions (see Table 3). Under the assumption that these external conditions affect the whole plant rather than specific genes, we have re-analyzed the Arabidopsis data with the BGM model, where changepoints are common to all nodes.

Figure 16 shows the marginal posterior probability of the changepoint locations and the posterior probability of the co-allocation of two time points to the same component for the BGM model. The three true segment boundaries are clearly detected. There is one additional changepoint subdividing the third time series segment, though. Interestingly, this changepoint is also detected with the cpBGe model: from Fig. 14 it is seen that genes CCA1, ELF3, and PRR5 show transitions that lag behind the change of the experimental conditions. A plausible explanation is that these external transitions may induce a delayed effect at the molecular level. Figure 14 suggests that different genes are affected by this retardation to different extent, with the aforementioned genes showing a delayed changepoint, while for other genes—LHY, TOC1, GI and PRR3—the changepoint coincides with the external transition. As opposed to the cpBGe model, the BGM model does not have the flexibility to allow for retarded node-dependent transitions. Instead, by tying changepoints together, it imposes both the retarded changepoint exhibited by genes CCA1, ELF3, and PRR5 as well as the unretarded changepoint found in genes LHY, TOC1, GI and PRR3 onto the whole network. In terms of the individual genes this leads to a spurious transition, with each gene now having two rather than one changepoint. This suggests that the investigated system profits from the extra flexibility inherent to the cpBGe model.

How does the BGM model differ from the cpBGe model in terms of the network reconstruction? Figure 17 shows scatter plots of the inferred marginal posterior probabilities of the edges: BGM against cpBGe. Most marginal posterior probability pairs fluctuate around the diagonal reference line, with a Pearson correlation coefficient of 0.82. This indicates, overall, a considerable agreement between both models. There are some deviations in the

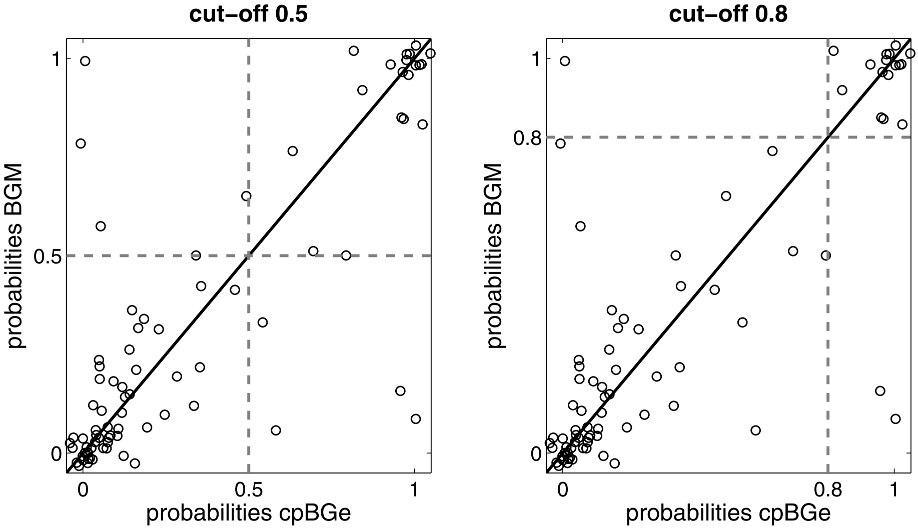


Fig. 17 Scatter plot of marginal edge posterior probabilities: cpBGe versus BGM. Both models BGM and cpBGe have been applied to the *Arabidopsis thaliana* data. The inferred marginal edge posterior probabilities can be plotted against each other. Both panels show the same scatter plot with a *diagonal reference line*. Additional *dotted grey horizontal* and *vertical reference lines* have been added to visualize those edges whose marginal posterior probabilities exceed a given cut-off (*left panel*: cut-off 0.5, *right panel*: cut-off 0.8). The coordinates of all points were randomly perturbed (by adding noise from a Gaussian distribution with mean $\mu = 0$ and standard deviation $\sigma = 0.01$ to each coordinate) to visualize clusters of points

predictions, though. Recall that a specific network prediction is obtained by imposing a cut-off threshold on the marginal edge posterior probabilities. Figure 17 shows that for a cut-off of 0.5, 4 edges are found with BGM but not with cpBGe and, conversely, 4 edges are found with cpBGe but not with BGM. With a total of 23 edges exceeding the threshold when using cpBGe, as shown in Fig. 15, the relative deviation is 17%. When increasing the threshold to 0.8, the number of deviating edges decreases to 2 edges recovered with cpBGe but not with BGM, and 1 edge detected with BGM but not with cpBGe. When using cpBGe, 14 edges pass the cut-off threshold; this corresponds to a relative deviation of 14%. The main difference (at the 0.5 threshold level) appears to be that three regulatees of PRR9 detected with cpBGe (LHY, CCA1, and PRR3) become regulatees of gene ELF3 when the BGM model is used. The replacement of another interaction related to LHY and CCA1—replacing CCA1 → ELF3 (inferred with cpBGe) by the edge LHY → ELF3 (inferred with BGM)—might be due to the fact that LHY and CCA1 are (partially redundant) homologues (Miwa et al. 2007). Unfortunately, a proper biological validation of the differences is not feasible at present owing to our limited insight into the nature of the molecular processes and the lack of a gold standard. For a quantitative assessment of the difference in the network reconstruction accuracy achieved with cpBGe versus BGM, we therefore refer the reader to the study discussed in Sects. 5.2 and 5.3.

6 Conclusions

We have proposed a continuous-valued non-homogeneous dynamic Bayesian network (DBN), which constitutes a non-homogeneous generalization of the BGe model. This complements the work of Robinson and Hartemink (2009), where a non-homogeneous BDe model was proposed. We have argued that a completely flexible network structure, as proposed by Lèbre (2007, 2010) can lead to over-fitting or inflated inference uncertainty, and we have therefore only allowed the parameters to vary with time. We have justified this approach with respect to the reconstruction of gene regulatory networks from short gene expression time series, where one would expect the strength of the interactions rather than their status of existence to evolve in time.

Our work expands and improves an earlier paper (Grzegorczyk and Husmeier 2009) in four important aspects: offering a comprehensive and self-contained exposition of the methodology, discussing the problem of spurious feedback loops, repeating our earlier simulations for a discrete rather than continuous changepoint process, and investigating how far mixing and convergence of the Markov chain can be improved with a dynamic programming scheme for sampling the changepoints.

We have demonstrated that when learning dynamic Bayesian networks from time series data, the presence of temporal autocorrelations and nonlinear regulatory functional relationships can render an approach based on the linear BGe score susceptible to spurious feedback loops. We have shown that the application of the proposed non-homogeneous DBN can substantially reduce the susceptibility to spurious feedback loops. This is a consequence of improved nonlinear modelling capability. When a regulator is driven by a feedback mechanism such that its associated signal is sufficiently smooth in time, then the proposed non-homogeneous model is effectively a piecewise linear model, which overcomes the linearity restriction of BGe.

We have replaced the continuous changepoint process of Grzegorczyk and Husmeier (2009) by a simpler discrete changepoint process, and we have rerun all the simulations of our earlier study. This provides a comprehensive comparative evaluation of the network reconstruction accuracy on synthetic data, which is missing from recent related studies on this topic, like Robinson and Hartemink (2009) and Lèbre (2007, 2010). Our findings suggest that the proposed non-homogeneous cpBGe model achieves a clear performance improvement over the classical homogeneous DBNs with BDe and BGe scores, as well as over the nonlinear/non-stationary models GM_{BIC} (Ko et al. 2007) and BGM (Grzegorczyk et al. 2010). The application of our model to gene expression time series from circadian clock-regulated genes in *Arabidopsis thaliana* has led to a plausible data segmentation, and the reconstructed network shows features that are consistent with the biological literature.

We have invested considerable efforts into improving and assessing mixing and convergence of the MCMC scheme, addressing both the sampling of network structures, and the sampling of changepoint configurations. We have shown that classical structure MCMC, which is based on single-edge operations, suffers from very poor convergence, and that the introduction of a novel single-parent exchange move leads to a substantial improvement. We have implemented and studied the effect of the dynamic programming schemes proposed by Fearnhead (2006) in the context of mixture models. These schemes allow the changepoints to be sampled from the correct conditional distribution. The essential difference between Fearnhead (2006) and our work is the conditioning part of these distributions. For the mixture models studied by Fearnhead (2006), the conditional distributions are dependent on the hyperparameters of the mixture model. These hyperparameters typically span a low-dimensional space, and even if they are slightly out of tune, the conditional distribution of the changepoints is usually not affected drastically. This allows the application of a computational trick based on sampling many changepoint configurations from the same conditional distribution at reduced computational costs (of additive rather than multiplicative complexity in the number of samples), and then correcting for the mismatch between the hyperparameters by the application of the Metropolis-Hastings acceptance criterion. In our work, the conditional distributions are dependent on the network structures associated with the different segments. Changing the network structures in the segments can have a considerable impact on the conditional distributions, and the computational trick referred to above is no longer applicable. The implication is that the dynamic programming scheme comes with substantial computational overheads, and it is therefore not clear from the outset whether it achieves any improvement over the RJMCMC schemes applied in Robinson and Hartemink (2009), Lèbre (2007, 2010), and our earlier work: Grzegorczyk and Husmeier (2009). Two different dynamic programming schemes were proposed in Fearnhead (2006), which differ with respect to the prior distribution on the changepoints. The natural choice appears to be a prior distribution on the number of changepoints, as in Robinson and Hartemink (2009), Lèbre (2007, 2010), and Grzegorczyk and Husmeier (2009). However, our findings suggest that the resulting computational costs of the dynamic programming scheme are so high that the improvement over RJMCMC with classical structure MCMC are modest, and no improvement over RJMCMC with parent-exchange structure MCMC can be achieved. As an alternative, we have therefore studied a point process prior on the times between two successive changepoints. As already discussed in Fearnhead (2006), the choice of this prior distribution reduces the computational costs of the dynamic programming scheme. A comparison to RJMCMC with classical structure MCMC indicates a substantial improvement in convergence. The improvement over RJMCMC with parent-exchange structure MCMC is less pronounced, but still tends to be significant.

The proposed model is based on a multiple changepoint process. A straightforward modification would be the replacement of the changepoint process by the allocation model of Nobile and Fearnside (2007) and Grzegorczyk et al. (2008). This modification would result in a fully-flexible mixture model, which would provide a more general approximation of nonlinear processes than with the proposed non-homogeneous DBN, and it could also be applied to static data. While the algorithmic implementation is in principle straightforward, the computational complexity of the latent variable configuration space would increase substantially. This would introduce new challenges for improving the mixing and convergence properties of the MCMC sampler, beyond those that have been discussed in the present work.

Acknowledgements Marco Grzegorczyk works for the research group of Jörg Rahnenführer at TU Dortmund University and is supported by the Graduate School "Statistische Modellbildung" of the Department of Statistics, TU Dortmund University. Dirk Husmeier is supported by the Scottish Government Rural and Environment Research and Analysis Directorate (RERAD). The authors are grateful to Paul Fearnhead for helpful discussions of some aspects of the dynamic programming scheme as well as to two anonymous referees for constructive feedback on an earlier version of this paper.

Appendix A: Merging independent data sets

We consider the scenario where d independent data sets $\mathcal{D}^1, \dots, \mathcal{D}^d$ are available. Let $\mathcal{D}^w$ be a N-by-m_w matrix consisting of m_w time-dependent realizations of the N variables ($w = 1, \dots, d$). Before we can apply a dynamic Bayesian network (DBN) model we have to merge the data sets appropriately into one single data set $\mathcal{D} = (\mathcal{D}^1, \dots, \mathcal{D}^d)$. It has to be taken into account that the gene expression values at the first time point of a time series segment $\mathcal{D}^w_{.,1}$ are independent of the expression values at the last time point of the preceding data segment

$\mathcal{D}^{w-1}_{\cdot,m_{(w-1)}}$. Consequently, since there are no realizations of potential parent nodes for the first time point of each data segment $\mathcal{D}^w$, the first time point of each data segment cannot be scored. The marginal likelihood in (2)–(3) of the BGe model has to be replaced by:

$$P(\mathcal{D}|\mathcal{G}) = \int P(\mathcal{D}|\mathcal{G},\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathcal{G})d\boldsymbol{\theta} = \prod_{n=1}^{N}\Psi(\mathcal{D}_n^{\pi_n},\mathcal{G})$$

$$\Psi(\mathcal{D}_n^{\pi_n},\mathcal{G}) = \int \prod_{w=1}^{d}\prod_{t=2}^{m_w} P\Big(X_n(t)=\mathcal{D}^w_{n,t}|\pi_n(t-1)=\mathcal{D}^w_{(\pi_n,t-1)},\boldsymbol{\theta}_n\Big)P(\boldsymbol{\theta}_n|\mathcal{G})d\boldsymbol{\theta}_n$$

where $\mathcal{D}_n^{\pi_n} := \{(\mathcal{D}^w_{n,t},\mathcal{D}^w_{\pi_n,t-1}) : 2 \le t \le m_w, 1 \le w \le d\}$ consists of the subsets of the d data segments pertaining to node X_n and parent set π_n. The marginal likelihood of the proposed cpBGe model (see (5)–(7)) can be modified accordingly. For the merged data set $\mathcal{D}$ we define the node-specific allocation vector $\mathbf{V}_n$ of the cpBGe model slightly differently to take into consideration that there are no realizations for the potential parent nodes of the first time points $\mathcal{D}^1_{\cdot,1},\dots,\mathcal{D}^d_{\cdot,1}$. We define for $t=2,\dots,1+(m_1-1)+\cdots+(m_w-1)$ that the realization $\mathbf{V}_n(t)$ corresponds to the s-th realization in data segment $\mathcal{D}^q$ where

$$q = 1 + \max\left\{u \in \{0,\dots,d\}\middle| t - \sum_{w=1}^{u}(m_w-1) > 0\right\} \tag{77}$$

and $s = t - \sum_{w=1}^{q}(m_w-1)$. With this definition the allocation vectors $\mathbf{V}_n$ can mathematically be treated as if they stemmed from one single time series. Practically, the vectors act as filters that sub-divide the merged data set $\mathcal{D}$ into subsets and the starting points of each segment are cut out.

Appendix B: The BGM model

In our implementation the Bayesian Gaussian mixture (BGM) model can be seen as a special case of the proposed cpBGe model. Instead of employing node-specific numbers of components $\mathcal{K}_n$ and allocation vectors $\mathbf{V}_n$ we restrict on one single vector $\mathbf{V}$ assigning $t=2,\dots,m$ to $\mathcal{K}$ components. $\mathbf{V}(t)=k$ indicates that the t-th realization has been generated by the k-th component. The marginal likelihood conditional on $\mathbf{V}$ is given by

$$P(\mathcal{D}|\mathcal{G},\mathbf{V},\mathbf{K}) = \int P(\mathcal{D}|\mathcal{G},\mathbf{V},\mathbf{K},\boldsymbol{\theta})P(\boldsymbol{\theta})d\boldsymbol{\theta} = \prod_{n=1}^{N}\Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathcal{K},\mathbf{V}]) \tag{78}$$

$$\Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathcal{K},\mathbf{V}]) = \prod_{k=1}^{\mathcal{K}}\Psi(\mathcal{D}_n^{\pi_n}[k,\mathbf{V}]) \tag{79}$$

where $\Psi(\mathcal{D}_n^{\pi_n}[k,\mathbf{V}])$ is the *local BGM score* and the factors in (79) are local BGe scores that have been defined in (3). The posterior distribution of the BGM model is:

$$P(\mathcal{G},\mathbf{V},\mathbf{K},\mathcal{D}) = P(\mathbf{V}|\mathcal{K})P(\mathcal{K})\prod_{n=1}^{N}P(\pi_n)\Psi^{\dagger}(\mathcal{D}_n^{\pi_n}[\mathcal{K},\mathbf{V}]) \tag{80}$$

where $\mathcal{K}$ is Poisson distributed with $\lambda=1$ and truncated to $1\le\mathcal{K}\le\mathcal{K}_{max}$ and the $P(\mathbf{V}|\mathcal{K})$ is implicitly defined via a changepoint process. We identify $\mathcal{K}$ with $\mathcal{K}-1$ changepoints on the set $\{2, \dots, m-1\}$ so that $\mathbf{V}(t) = k$, if and only if $b_{k-1} \leq t < b_k$, where b_k is the k-th changepoint. 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 $\{2, \dots, m-1\}$. To obtain a sample $\{\mathcal{G}^i, \mathbf{V}^i, \mathcal{K}^i\}_{i=1,\dots,I}$ from the posterior distribution of the BGM model we combine the structure MCMC algorithm (Giudici and Castelo 2003 and Madigan and York 1995) with the reversible jump MCMC sampling scheme for changepoints presented in Green (1995). With probability p_G we perform a single edge move on the graph $\mathcal{G}^i$ and leave $\mathbf{V}$ and $\mathcal{K}$ unchanged. The new candidate graph $\mathcal{G}^{i+1}$ is obtained by randomly selecting a node X_n and changing its parent set π_n^i to π_n^{i+1} by a single-edge operation as described in Sect. 2.3.1. The acceptance probability is given by:

$$A(\mathcal{G}^{i+1}|\mathcal{G}^i) = \min\left\{1, \frac{\Psi^\dagger(\mathcal{D}_n^{\pi_n^{i+1}}[\mathcal{K}, \mathbf{V}])}{\Psi^\dagger(\mathcal{D}_n^{\pi_n^{i}}[\mathcal{K}, \mathbf{V}])} \frac{P(\pi_n^{i+1})}{P(\pi_n^i)} \frac{|\mathcal{N}(\pi_n^i)|}{|\mathcal{N}(\pi_n^{i+1})|}\right\} \tag{81}$$

With probability $1 - p_{\mathcal{G}}$ we leave $\mathcal{G}$ unchanged and propose a move on $(\mathbf{V}^i, \mathcal{K}^i)$ along the lines of the changepoint birth, death and re-allocation moves described in Sect. 2.3.1. The new candidate $(\mathbf{V}^{i+1}, \mathcal{K}^{i+1})$ is accepted with probability $R = \min(1, A)$, where A is of the following form:

$$R = \frac{\prod_{n=1}^{N} \prod_{k=1}^{\mathcal{K}^{i+1}} \Psi(\mathcal{D}_n^{\pi_n}[k, \mathbf{V}^{i+1}])}{\prod_{n=1}^{N} \prod_{k=1}^{\mathcal{K}^{i}} \Psi(\mathcal{D}_n^{\pi_n}[k, \mathbf{V}^{i}])} \times A \times B \tag{82}$$

where $A = P(\mathbf{V}^{i+1}|\mathcal{K}^{i+1})P(\mathcal{K}^{i+1})/P(\mathbf{V}^i|\mathcal{K}^i)P(\mathcal{K}^i)$ is the prior probability ratio, and the inverse proposal probability ratio B depends on the move type.

Appendix C: The GM$_{BIC}$ model

The BIC score of a graph $\mathcal{G}$ is defined as follows:

$$Score(\mathcal{G}) = \log(P(\mathcal{D}|\mathcal{G}, \widehat{\boldsymbol{\theta}})) - \frac{m}{2} \cdot |\widehat{\boldsymbol{\theta}}| \tag{83}$$

where $\widehat{\boldsymbol{\theta}}$ is the maximum likelihood (ML) estimate of the unknown parameters, and $|\widehat{\boldsymbol{\theta}}|$ is the number of unknown parameters. The Gaussian mixture (GM) model of Ko et al. (2007) is a node-specific mixture model with node-specific mixture weight parameters $\alpha_{n,k}$. Conditional on the numbers of mixture components: $\mathbf{K} = (\mathcal{K}_1, \dots, \mathcal{K}_n)$ the likelihood of the GM model factorizes:

$$P(\mathcal{D}|\mathcal{G}, \mathbf{K}, \boldsymbol{\theta}) = \prod_{n=1}^{N} \prod_{t=2}^{m} \sum_{k=1}^{\mathcal{K}_n} \alpha_{n,k} P(X_n(t) = \mathcal{D}_{n,t}|\pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)}, \boldsymbol{\theta}_n^k) \tag{84}$$

There is no closed-form ML estimate $\widehat{\boldsymbol{\theta}}$ for the weights $\alpha_{n,k}$ and parameters $\boldsymbol{\theta}_n^k$, and Ko et al. (2007) apply the EM-algorithm to obtain estimates: $\widehat{\boldsymbol{\theta}_n^{k,\dagger}}$ and $\widehat{\alpha_{n,k,\dagger}}$ $(k = 1, \dots, \mathcal{K}_n)$ for the N joint probability distributions:

$$\prod_{t=2}^{m} \sum_{k=1}^{\mathcal{K}_n} \alpha_{n,k,\dagger} P(X_n(t) = \mathcal{D}_{n,t}, \pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)}, \boldsymbol{\theta}_n^{k,\dagger}) \tag{85}$$

and draw on the fact that the marginal probability distribution of the parent nodes in π_n is the same as the joint probability distribution in (85) with all the parameters corresponding to the child node X_n removed. That is, Ko et al. remove all ML estimates corresponding to the child node X_n from $\widehat{\boldsymbol{\theta}_n^{k,\dagger}}$ and plug the remaining parameters $\widehat{\boldsymbol{\theta}_n^{k,\ddagger}} \subset \widehat{\boldsymbol{\theta}_n^{k,\dagger}}$ and the estimated weights $\widehat{\alpha_{n,k,\ddagger}} := \widehat{\alpha_{n,k,\dagger}}$ $(k = 1, \dots, \mathcal{K}_n)$ into the (marginal) likelihood:

$$\prod_{t=2}^{m} \sum_{k=1}^{\mathcal{K}_n} \alpha_{n,k,\ddagger} P(\pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)}, \boldsymbol{\theta}_n^{k,\ddagger}) \tag{86}$$

to obtain an approximate[22] estimate for the ML value of the marginal probability distribution of the parent nodes in π_n. This is done independently for all N local distributions and from the definition of conditional probability distributions it follows:

$$P(\mathcal{D}|\mathcal{G}, \mathbf{K}, \widehat{\boldsymbol{\theta}}) = \prod_{n=1}^{N} \prod_{t=2}^{m} \frac{\sum_{k=1}^{\mathcal{K}_n} \widehat{\alpha_{n,k,\dagger}} P(X_n(t) = \mathcal{D}_{n,t}, \pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)}, \widehat{\boldsymbol{\theta}_n^{k,\dagger}})}{\sum_{k=1}^{\mathcal{K}_n} \widehat{\alpha_{n,k,\dagger}} P(\pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)}, \widehat{\boldsymbol{\theta}_n^{k,\ddagger}})} \tag{87}$$

For each of the N local (conditional) distributions in (84) the parameters of the joint posterior distributions of X_n and π_n, symbolically $\widehat{\alpha_{n,1,\dagger}}, \dots, \widehat{\alpha_{n,\mathcal{K}_n,\dagger}}, \widehat{\boldsymbol{\theta}_n^{1,\dagger}}, \dots, \widehat{\boldsymbol{\theta}_n^{\mathcal{K}_n,\dagger}}$ are maximized independently with the EM-algorithm on the data subset: $\mathcal{D}_n^{\pi_n} = \{(\mathcal{D}_{n,t}, \mathcal{D}_{\pi_n,t-1}) : 2 \le t \le m\}$. The ML estimates for the marginal likelihood of the parent nodes in π_n are approximated by removing all parameters corresponding to the child node X_n from $\widehat{\alpha_{n,k,\dagger}}$ and leaving the weights $\widehat{\alpha_{n,k,\dagger}}$ unchanged $(k = 1, \dots, \mathcal{K}_n)$. The number of estimated parameters is then given by:

$$|\widehat{\boldsymbol{\theta}}(\mathcal{G}, \mathbf{K})| = \sum_{n=1}^{N} \Big((|\pi_n| + 1) + (|\pi_n| + 2)(|\pi_n| + 1)/2)\Big)\mathcal{K}_n + (\mathcal{K}_n - 1) \tag{88}$$

where $\mathbf{K} = (\mathcal{K}_1, \dots, \mathcal{K}_n)$ are the numbers of components, and $|\pi_n|$ is the cardinality of the parent node set of X_n. For clarity, we note that $(|\pi_n| + 1)$ expectation parameters and $(|\pi_n| + 2) \cdot (|\pi_n| + 1)/2$ covariance parameters have to be estimated for each of the $\mathcal{K}_n$ mixture components and that there are $(\mathcal{K}_n - 1)$ (unknown) mixture weights. The *GM* score of a graph $\mathcal{G}$ is given by:

$$S(\mathcal{G}|GM) = \max\left\{\log(P(\mathcal{D}|\mathcal{G}, \mathbf{K}, \widehat{\theta})) - \frac{m}{2}|\widehat{\theta}(\mathcal{G}, \mathbf{K})| : \mathbf{K} = (\mathcal{K}_1, \dots, \mathcal{K}_n)\right\} \tag{89}$$

where the numbers of components $\mathcal{K}_n$, that is the elements in the vector $\mathbf{K}$, can be restricted: $1 \le \mathcal{K}_n \le \mathcal{K}_{MAX}$, and $P(\mathcal{D}|\mathcal{G}, \mathbf{K}, \widehat{\theta})$ was defined in (87). The *GM* estimator of the network structure is given by the graph $\mathcal{G}^\star$ with the highest score: $S(\mathcal{G}^\star|GM) \ge S(\mathcal{G}|GM)$ for all possible graphs $\mathcal{G}$.

[22] Note that this procedure is exact for a multivariate Gaussian distribution, but not for a mixture of multivariate Gaussians.

Appendix D: Matching the computational costs of the MCMC samplers

Recalling the computational costs shown in Table 5 we proceed as follows: For the Metropolis-Hastings samplers we compute the PSRF-based $\mathcal{C}(\xi)$ diagnostic after each of the following numbers of iterations: $2s = 66k, 88k, \dots, 1100k$ (where s is the burn-in phase length) and sample equidistantly with a distance of $11k$ steps $I = 3, 4, \dots, 50$ graphs from the last s iterations. For the Gibbs sampling scheme with $\mathcal{K}_{MAX} = 10$ ($\mathcal{K}_{MAX} = 5$) we perform $2s = 550$ ($2s = 1100$) iterations in total, and in the sampling phase a graph can be sampled only every 11th step, i.e. after each of the $N = 11$ nodes of the RAF network has (potentially) obtained a new parent set and a new node-specific allocation vector. With the burn-in phase length of s this gives a maximal sample size of $I = 25$ ($I = 50$), and we consider $2s = 66, 88, \dots, 550$ ($2s = 66, 88, \dots, 1100$) and sample equidistantly with a distance of 11 steps (22 steps) from the last s iterations $I = 3, 4, \dots, 25$ ($I = 3, 4, \dots, 50$) graphs. Finally, for the Gibbs sampling scheme with the point process prior we consider $2s = 660, 770, \dots, 5500$ and sample with a distance of 55 steps $I = 6, 7, \dots, 50$ graphs from the last s iterations. For each of the five sampling schemes $\mathcal{M} = 1, \dots, 5$ we then compute $\mathcal{C}(\xi)_{\mathcal{M},I}$ for each available sample size I. For the Metropolis-Hastings samplers ($\mathcal{M} = 1, 2$), and the Gibbs sampler with $\mathcal{K} = 5$ ($\mathcal{M} = 3$) this procedure yields 48 diagnostic values. For the Gibbs sampler with the point process prior ($\mathcal{M} = 4$) the first three values for $I = 3, 4, 5$ are missing. But for these four methods the values $\mathcal{C}(\xi)_{\mathcal{M},I}$ for $I = 6, \dots, 50$ correspond to the same computational costs and are immediately comparable. For the Gibbs samplers with $\mathcal{K} = 10$ ($\mathcal{M} = 5$) we have to map the 23 diagnostic values $\mathcal{C}(\xi)_{5,I}$ for $I = 3, \dots, 25$ onto $\mathcal{C}(\xi)_{\mathcal{M},2I}$ ($\mathcal{M} = 1, \dots, 4$) to take the mismatch in the computational costs into account.

Appendix E: Scatter plots of marginal edge posterior probabilities

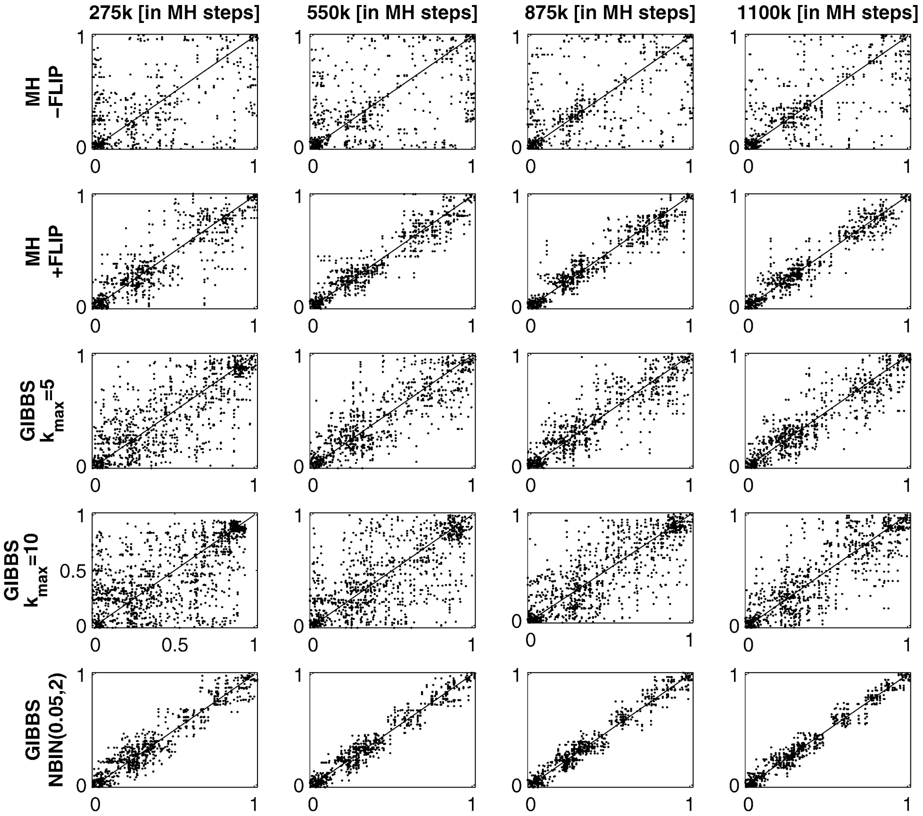


Fig. 18 Scatter plot of marginal edge posterior probabilities—data from RAF pathway with $SNR = 3$ and $\varepsilon = 0.25$. We compare the five MCMC schemes of Table 5. (i) **MH(−FLIP)**: RJMCMC with standard structure MCMC; (ii) **MH(+FLIP)**: RJMCMC based on structure MCMC with the parent exchange (FLIP) move; (iii) **Gibbs(K = 10)**: Gibbs sampling with dynamic programming, using a prior on the number of components truncated at $\mathcal{K}_{MAX} = 10$; (iv) **Gibbs(K = 5)**: Idem, but truncated at $\mathcal{K}_{MAX} = 5$; (v) **Gibbs-NBIN**: Gibbs sampling with dynamic programming, using a point process prior on the distances between changepoints. For each MCMC scheme the marginal edge posterior probabilities have been computed from 10 independent MCMC runs. This gives $\binom{10}{2} = 45$ ways of plotting the marginal edge posterior probabilities from one run against another, which have been superimposed in the panels

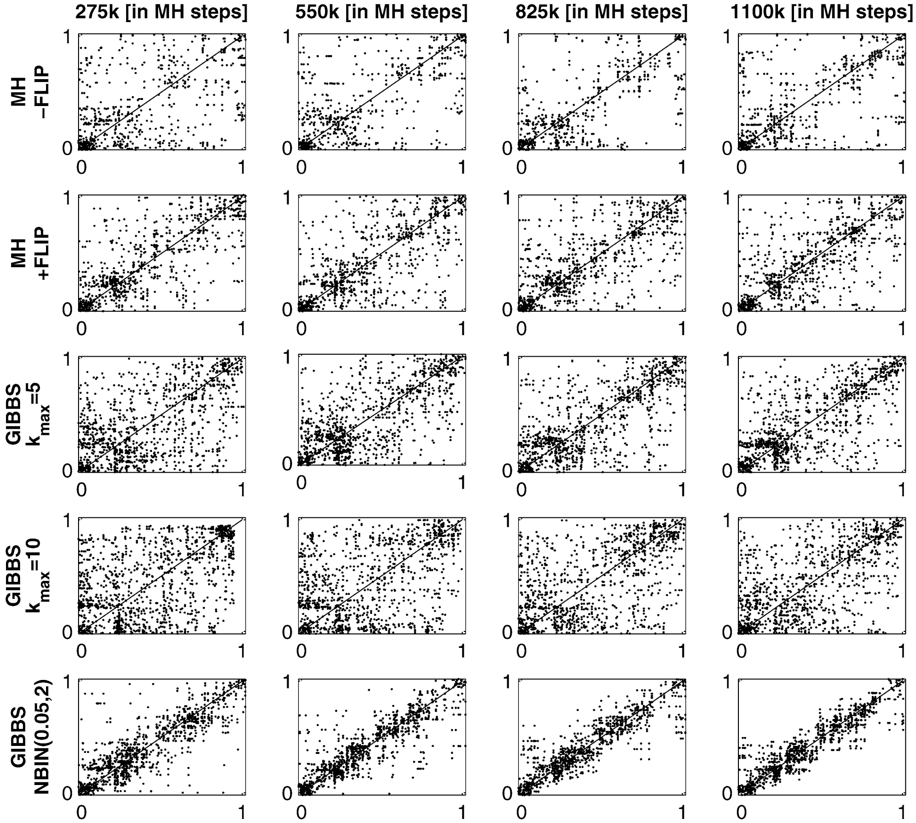


Fig. 19 Scatter plot of marginal edge posterior probabilities—data from RAF pathway with $SNR = 1$ and $\varepsilon = 0.25$. We compare the five MCMC schemes of Table 5. (i) **MH(−FLIP)**: RJMCMC with standard structure MCMC; (ii) **MH(+FLIP)**: RJMCMC with structure MCMC plus parent exchange (flip) move; (iii) **Gibbs(K = 10)**: Gibbs sampling with dynamic programming, using a prior on the number of components truncated at $\mathcal{K}_{MAX} = 10$; (iv) **Gibbs(K = 5)**: Idem, but truncated at $\mathcal{K}_{MAX} = 5$; (v) **Gibbs-NBIN**: Gibbs sampling with dynamic programming, using a point process prior on the distances between changepoints. For each MCMC scheme the marginal edge posterior probabilities have been computed from 10 independent MCMC runs. This gives $\binom{10}{2} = 45$ ways of plotting the marginal edge posterior probabilities from one run against another, which have been superimposed in the panels

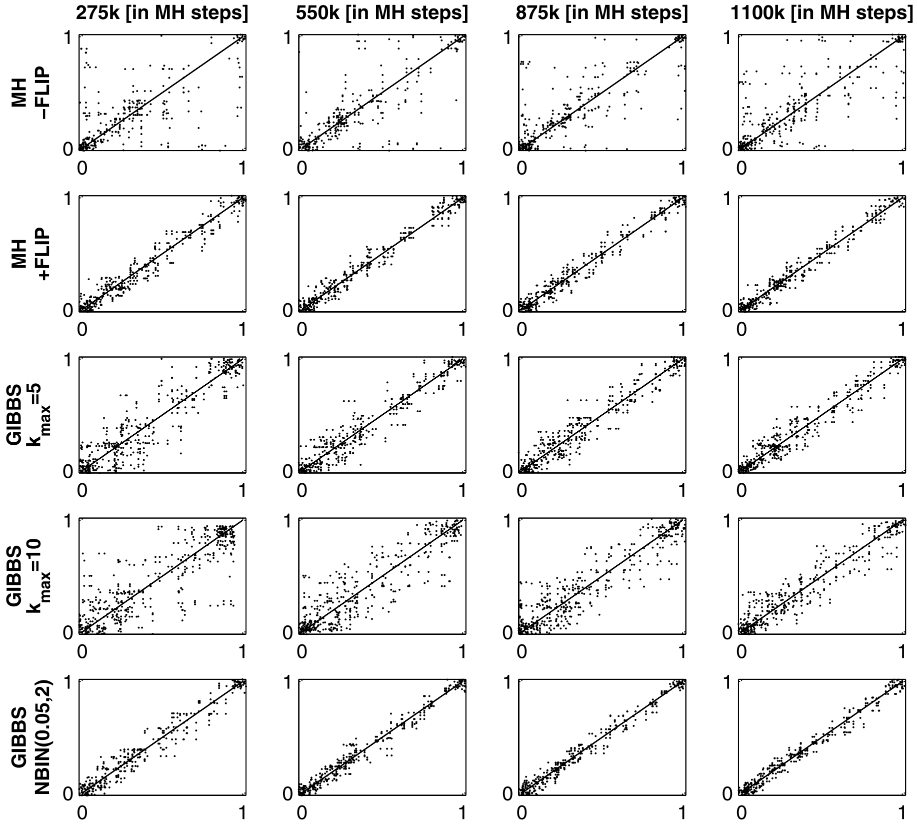


Fig. 20 Scatter plot of marginal edge posterior probabilities—*Arabidopsis thaliana*. We compare the five MCMC schemes of Table 5. (i) **MH(−FLIP)**: RJMCMC with standard structure MCMC; (ii) **MH(+FLIP)**: RJMCMC with structure MCMC improved by the parent exchange (FLIP) move; (iii) **Gibbs(K = 10)**: Gibbs sampling with dynamic programming, using a prior on the number of components truncated at $\mathcal{K}_{MAX} = 10$; (iv) **Gibbs(K = 5)**: Idem, but truncated at $\mathcal{K}_{MAX} = 5$; (v) **Gibbs-NBIN**: Gibbs sampling with dynamic programming, using a point process prior on the distances between changepoints. For each MCMC scheme the marginal edge posterior probabilities have been computed from 10 independent MCMC runs. This gives $\binom{10}{2} = 45$ ways of plotting the marginal edge posterior probabilities from one run against another, which have been superimposed in the panels

Appendix F: Hybrid Gibbs and RJMCMC sampling schemes

As proposed by an anonymous reviewer, we have also investigated whether a hybrid sampling scheme that combines Gibbs and RJMCMC sampling converges faster than the two original (pure) samplers. We have generated hybrid sampling schemes that randomly switch between RJMCMC and Gibbs sampling. To this end, we pre-define a probability p_{Gibbs}, with which the hybrid samplers perform a Gibbs sampling step, while a series of k_{MH} RJMCMC steps is performed with probability $p_{MH} = 1 - p_{Gibbs}$. Among the Gibbs sampling schemes, described in Sects. 2.7.1 and 2.7.2, Gibbs-NBIN (described in Sect. 2.7.1) is the most effective one (see Sect. 5.4), and the RJMCMC sampling scheme is improved by the proposed flip move (see Sect. 5.4). We therefore decided to combine the Gibbs-NBIN sampling scheme, which employs a point process prior on the distances between changepoints,

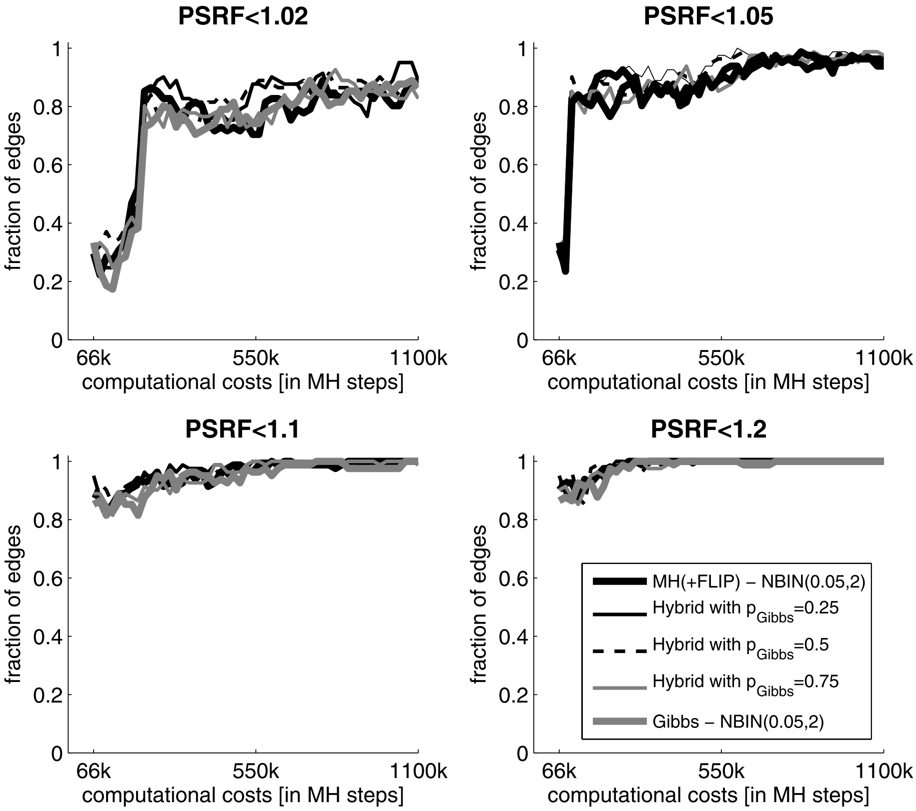


Fig. 21 Convergence diagnostics based on potential scale reduction factors (PSRFs) of individual network edges for *Arabidopsis thaliana* network—hybrid Gibbs/RJMCMC sampling schemes. We compared the best original sampling schemes with three hybrid sampling schemes. The two best original samplers are the RJMCMC sampler with structure MCMC plus parent exchange move (**MH(+FLIP)**) and the **Gibbs-NBIN** sampler with the point process prior on the distances between changepoints. To be consistent, the RJMCMC sampler has also been implemented with a point process prior on the distances between changepoints. During the MCMC simulation the hybrid sampling schemes either perform a **Gibbs-NBIN** sampling step with probability p_{Gibbs} or 200 MH(+FLIP) RJMCMC steps otherwise, where the parameter p_{Gibbs} was varied: $p_{Gibbs} = 0.25, 0.5, 0.75$. For each sampling scheme 10 independent MCMC simulations were performed on the *Arabidopsis thaliana* data set and a PSRF was computed for each individual edge. Each panel shows overlaid trace plots of the fractions of individual edges whose PSRF was lower than the threshold (1.2, 1.1, 1.05, and 1.02). The computational costs on the horizontal axis are given in Metropolis-Hastings MCMC iterations. Details on how we defined a PSRF for an individual edge can be found in Sect. 4.3

with the MH(+FLIP) RJMCMC sampling scheme. It has to be taken into account that these two samplers employ different prior distributions for the number of changepoints and the changepoint locations. As explained in Sect. 2.7.1, the effectiveness of the Gibbs-NBIN sampler requires a point process prior on the distances between changepoints. Thus, we have replaced the original prior of the MH(+FLIP) RJMCMC sampler from Sect. 2.2 by this point process prior on the distances between changepoints of the Gibbs-NBIN sampler (see (21)–(24)). Recalling from Sect. 4.3 that the computational costs of 200 MH(+FLIP) RJMCMC steps match the computational costs of one single Gibbs-NBIN step, we set $k_{MH} = 200$.

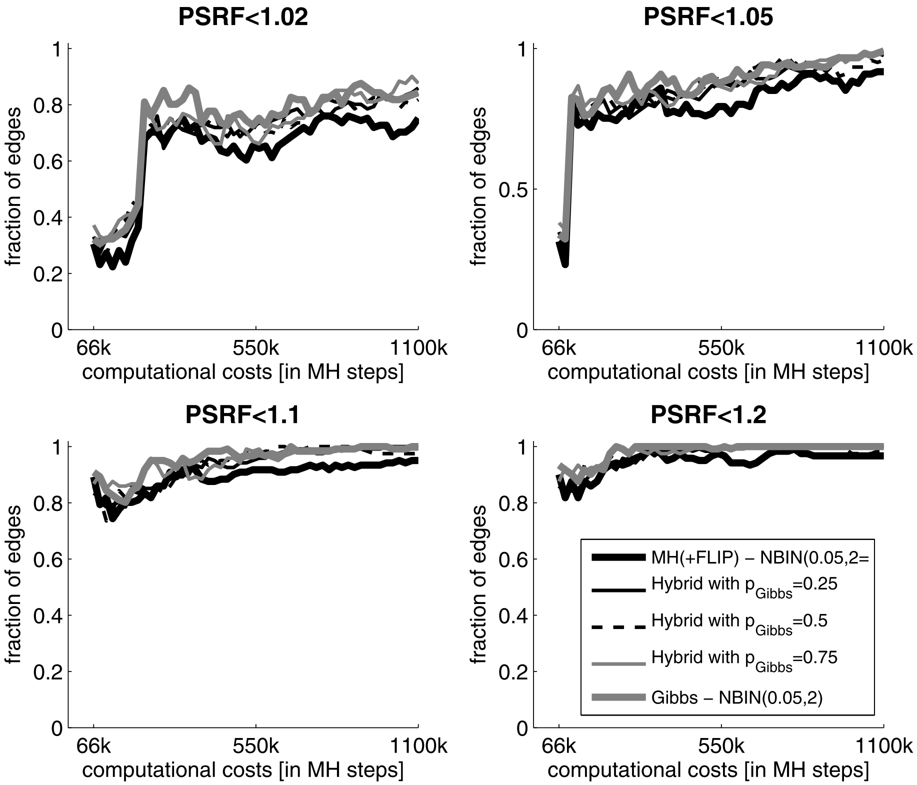


Fig. 22 Convergence diagnostics based on potential scale reduction factors (PSRFs) of individual network edges for the RAF network with $SNR = 1$—hybrid Gibbs/RJMCMC sampling schemes. For each sampling scheme 10 independent MCMC simulations were performed on the same synthetic RAF-network data set with $SNR = 1$ and $\varepsilon = 0.25$. A PSRF was computed for each individual edge. See caption of Fig. 21 for further explanations

A sufficient degree of convergence in terms of PSRF values was observed for the RAF-pathway data set with $SNR = 3$ and $\varepsilon = 0.25$ (see Fig. 10), while there is room for improvement for the RAF-pathway data with $SNR = 1$ and for the Arabidopsis data for lower PSRF thresholds (see upper panels of Figs. 10 and 12). Thus, we monitor mixing and convergence of three hybrid sampling schemes for the data set from the RAF-pathway with $SNR = 1$ and $\varepsilon = 0.25$ and the Arabidopsis data set. Figures 21 and 22 show the results for $p_{Gibbs} = 0.25, 0.5, 0.75$. It can be seen that the hybridization of Gibbs and RJMCMC sampling does not yield any significant improvement over the two original sampling schemes. From our perspective this finding is not surprising. Combining the effective Gibbs-NBIN sampling scheme with a less effective RJMCMC sampling scheme, which is based on smaller steps in the posterior landscape, yields a hybrid sampler that tends to produce a stronger autocorrelation between samples, which does not improve the mixing of the Markov chain.

References

Ahmed, A., & 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.

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

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

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. In *Proceedings of the twenty-third international conference on machine learning (ICML)* (pp. 233–240). New York: ACM.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, *B39*, 1–38.

Dougherty, M. K., Muller, J., Ritt, D. A., Zhou, M., Zhou, X. Z., Copeland, T. D., Conrads, T. P., Veenstra, T. D., Lu, K. P., & Morrison, D. K. (2005). Regulation of Raf-1 by direct feedback phosphorylation. *Molecular Cell*, *17*, 215–224.

Edwards, K. D., Anderson, P. E., Hall, A., Salathia, N. S., Locke, J. C., Lynn, J. R., Straume, M., Smith, J. Q., & 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., & Koller, D. (2003). Being Bayesian about network structure. *Machine Learning*, *50*, 95–126.

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, *7*, 601–620.

Geiger, D., & Heckerman, D. (1994). Learning Gaussian networks. In *Proceedings of the tenth conference on uncertainty in artificial intelligence* (pp. 235–243). San Francisco: Morgan Kaufmann.

Giudici, P., & 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., & 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., & Husmeier, D. (2009). Non-stationary continuous dynamic Bayesian networks. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, & A. Culotta (Eds.), *Advances in neural information processing systems (NIPS)* (Vol. 22, pp. 682–690).

Grzegorczyk, M., Husmeier, D., Edwards, K., Ghazal, P., & 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., & Husmeier, D. (2010). Modelling non-stationary dynamic gene regulatory processes with the BGM model. *Computational Statistics*. doi:10.1007/s00180-010-0201-9.

Hartemink, A. J. (2001) *Principled computational methods for the validation and discovery of genetic regulatory networks*. Ph.D. thesis, MIT.

Heckerman, D., & Geiger, D. (1995). Learning Bayesian networks: A unification for discrete and Gaussian domains. In *Proceedings of the 11th annual conference on uncertainty in artificial intelligence (UAI-95)* (pp. 274–82). San Francisco: Morgan Kaufmann.

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

Ko, Y., Zhai, C., & Rodriguez-Zas, S. (2007). Inference of gene pathways using Gaussian mixture models. In *BIBM International conference on bioinformatics and biomedicine*, Fremont, CA (pp. 362–367).

Kolar, M., Song, L., & Xing, E. (2009). Sparsistent learning of varying-coefficient models with structural changes. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, & A. Culotta (Eds.), *Advances in neural information processing systems (NIPS)* (pp. 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., Lelandais, G., & Stumpf, M. (2010). Statistical inference of the time-varying structure of gene-regulation networks. *BMC Systems Biology*, *4* (130).

Lim, W., Wang, K., Lefebvre, C., & 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., & Millar, A. (2005) Extension of a genetic network model by iterative experimentation and mathematical analysis. *Molecular Systems Biology*, *1* (online).

Madigan, D., & 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., Serikawa, M., Suzuki, S., Kondo, T., & 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.

Miwa, K., Ito, S., Nakamichi, N., Mizoguchi, T., Niinuma, K., Yamashino, T., & 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.

Mockler, T., Michael, T., Priest, H., Shen, R., Sullivan, C., Givan, S., McEntee, C., Kay, S., & 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.

Nobile, A., & Fearnside, A. (2007). Bayesian finite mixtures with an unknown number of components: The allocation sampler. *Statistics and Computing*, *17*, 147–162.

Robinson, J. W., & Hartemink, A. J. (2009). Non-stationary dynamic Bayesian networks. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems (NIPS)* (Vol. 21, pp. 1369–1376). San Mateo: Morgan Kaufmann.

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

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., & Nolan, G. P. (2005). Protein-signaling networks derived from multiparameter single-cell data. *Science*, *308*, 523–529.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461–464.

Shen-Orr, S. S., Milo, R., Mangan, S., & Alon, U. (2002). Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nature Genetics*, *31*, 64–68.

Smith, V. A., Yu, J., Smulders, T. V., Hartemink, A. J., & Jarvi, E. D. (2006). Computational inference of neural information flow networks. *PLoS Computational Biology*, *2*, 1436–1449.

Talih, M., & 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.

Werhli, A. V., & Husmeier, D. (2008). Gene regulatory network reconstruction by Bayesian integration of prior knowledge and/or different experimental conditions. *Journal of Bioinformatics and Computational Biology*, *6*, 543–572.

Xuan, X., & Murphy, K. (2007). Modeling changing dependency structure in multivariate time series. In Z. Ghahramani (Ed.), *Proceedings of the 24th annual international conference on machine learning (ICML 2007)* (pp. 1055–1062). New York: Omnipress.