Supplementary Material: Non-stationary continuous dynamic Bayesian networks (NIPS 2009)

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

Dirk Husmeier
Biomathematics & Statistics Scotland (BioSS)
JCMB, The King's Buildings, Edinburgh EH93JZ, United Kingdom
dirk@bioss.ac.uk

1 Introduction

This supplementary material provides additional information about our $cpBGe$ model, the MCMC simulations and the empirical results, which for space restrictions could not be included in the main paper. The most recent version of this supplementary material can be downloaded from the following website: http://www.bioss.ac.uk/∼dirk/papers/NIPS09/, and it might contain extra material or revised sections added after the NIPS submission deadline. The notation in the current version of the supplementary material follows [1], which deviates slightly from the main paper. The following seven sections 2 to 8 are organized as follows: In Section 2 we provide details about the BGe scoring metric for static Bayesian networks as developed by Geiger and Heckerman [1]. The BGe scoring metric for dynamic Bayesian networks is described in detail in Section 3. Section 4 is an extended version of the methodology section of our main paper. Section 5 provides some details on the four competing models: The focus is on the BGM model of Grzegorczyk et al. [5] and the Gaussian mixture model developed by Ko et al. [8]. In Section 6 we describe how we generated the synthetic network data for the comparative evaluation study presented in the main paper. In Section 7 we give all implementation details, such as choice of hyperparameters, MCMC simulation lengths, convergence diagnostics, etc.. Finally, in Section 8 we provide some additional figures and interpretations of the empirical results that could – due to space restrictions – not be included in the main paper.

2 The Gaussian BGe scoring metric for static Bayesian networks

This section describes the linear Gaussian BGe scoring metric (Bayesian metric for Gaussian networks having score equivalence) for static Bayesian networks as developed by Geiger and Heckerman [1]. Given a data set $\mathcal{D}$ with m observations of the variables $X_1, \ldots, X_N$:

$$\mathcal{D} = \begin{pmatrix} \mathcal{D}_{1,1} & \mathcal{D}_{1,2} & \ldots & \mathcal{D}_{1,m-1} & \mathcal{D}_{1,m} \\ \mathcal{D}_{2,1} & \mathcal{D}_{2,2} & \ldots & \mathcal{D}_{2,m-1} & \mathcal{D}_{2,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathcal{D}_{N,1} & \mathcal{D}_{N,2} & \ldots & \mathcal{D}_{N,m-1} & \mathcal{D}_{N,m} \end{pmatrix} \tag{1}$$

so that $\mathcal{D}_{n,j}$ denotes the jth realization of the nth node X_n, and the jth column of $\mathcal{D}$: $\mathcal{D}_{\cdot,j} = (\mathcal{D}_{1,j}, \ldots, \mathcal{D}_{N,j})^T$ is the jth realization vector of the variables. The Gaussian BGe model assumes that the observation vectors $\mathcal{D}_{\cdot,j}$ $(j = 1, \ldots, m)$ are a random sample from a multivariate Gaussian distribution $\mathcal{N}(\vec{\mu}, \Sigma)$ with an unknown mean vector $\vec{\mu}$ and an unknown covariance matrix Σ. The prior joint distribution of $\vec{\mu}$ and $W = \Sigma^{-1}$ is supposed to be the normal-Wishart distribution, that is, the conditional distribution of $\vec{\mu}$ given W is $\mathcal{N}(\vec{\mu}_0, (v \cdot W)^{-1})$ with $v > 0$, and the marginal distribution of W is a Wishart distribution with $\alpha > N + 1$ degrees of freedom and prior matrix T_0:

$$\mathcal{W}(\alpha, T_0) \;=\; c(n, \alpha)|T_0|^{\alpha/2}|\mathcal{W}|^{(\alpha-n-1)/2} \exp(-\frac{1}{2} tr(T_0 \mathcal{W})) \tag{2}$$

where $tr(T_0\mathcal{W})$ is the sum of the diagonal elements of $T_0\mathcal{W}$, and

$$c(n, \alpha) := \left\{ 2^{\alpha \cdot n/2} \cdot \pi^{n\cdot(n-1)/4} \cdot \prod_{i=1}^{n} \Gamma(\frac{\alpha + 1 - i}{2}) \right\}^{-1} \tag{3}$$

The condition $\alpha > N + 1$ ensures that the second moments of the posterior distribution are finite (see also Eq. (26) in [1]). Geiger and Heckerman show that the marginal likelihood $P(\mathcal{D}|\mathcal{G})$ of the data $\mathcal{D}$ given a graph $\mathcal{G}$ can then – under fairly weak conditions of parameter independence and parameter modularity – be computed in closed form. We define:

$$T_{\mathcal{D},m} := T_0 + S_{\mathcal{D},m} + \frac{v \cdot m}{v + m}(\vec{\mu}_0 - \overline{\mathcal{D}_m})(\vec{\mu}_0 - \overline{\mathcal{D}_m})^T \tag{4}$$

where

$$\overline{\mathcal{D}_m} := \frac{1}{m} \sum_{j=1}^{m} \mathcal{D}_{\cdot,j} \tag{5}$$

is the mean of the m realization vectors and

$$S_{\mathcal{D},m} := \sum_{j=1}^{m} (\mathcal{D}_{\cdot,j} - \overline{\mathcal{D}_m}) \cdot (\mathcal{D}_{\cdot,j} - \overline{\mathcal{D}_m})^T \tag{6}$$

T_0, μ_0, α, and v are hyperparameters of the normal-Wishart prior and have to be specified in advance. T_0 is an N-by-N matrix, μ_0 is an N-by-1 column vector, and v and α are 1-dimensional and usually referred to as total prior precision parameters.

The marginal likelihood can be computed as follows ([1]):

$$P(\mathcal{D}|\mathcal{G}) = \prod_{n=1}^{N} \Psi(\mathcal{D}_n^{\pi_n}, \mathcal{G}) = \prod_{n=1}^{N} \frac{P(\mathcal{D}^{\{X_n, \pi_n\}}|\mathcal{G}_F(\{X_n, \pi_n\})}{P(\mathcal{D}^{\{\pi_n\}}|\mathcal{G}_F(\pi_n))} \tag{7}$$

where X_n is the nth variable, π_n is the parent set of X_n in the graph $\mathcal{G}$, $\mathcal{D}^{\{X_n,\pi_n\}}$ and $\mathcal{D}^{\{\pi_n\}}$ are the data submatrices corresponding to the realizations of the variables in the sets $\{X_n, \pi_n\}$ and $\{\pi_n\}$ only, and $\mathcal{G}_F(\{X_n, \pi_n\})$ and $\mathcal{G}_F(\pi_n)$ correspond to so-called *full graphs* for the variable subsets $\{X_n, \pi_n\}$ and $\{\pi_n\}$, that is, to subgraphs with the maximal number of edges so that the subgraphs do not impose any independence restrictions on these subsets of variables.

The marginal likelihood of the data subset $\mathcal{D}^{\{S\}} \subset \mathcal{D}$ corresponding to the m realizations of the $N^\dagger$-dimensional subset $S \subset \{X_1, \ldots, X_N\}$ of the N variables given a full graph $\mathcal{G}_F(S)$ for the sub-domain S can be computed as follows ([1]):

$$\begin{aligned} P(\mathcal{D}^S|\mathcal{G}_F(S)) \quad &= \quad (2\pi)^{-\frac{N^\dagger \cdot m}{2}} \cdot \left\{ \frac{v}{v+m} \right\}^{N^\dagger/2} \cdot \frac{c(N^\dagger, \alpha)}{c(N^\dagger, \alpha + m)} \\ &\quad \cdot det(T_0^S)^{\frac{\alpha}{2}} \cdot det(T_{\mathcal{D},m}^S)^{-\frac{\alpha+m}{2}} \end{aligned} \tag{8}$$

where $det(T_0^S)$ and $det(T_{\mathcal{D},m}^S)$ denote the determinants of the submatrices T_0^S and $T_{\mathcal{D},m}^S$ consisting only of those $N^\dagger$ rows and columns that correspond to variables in the subset S. $T_\mathcal{D}$ was defined in Eq. (4), and $c(N^\dagger, \alpha)$ and $c(N^\dagger, \alpha + m)$ can be computed with Eq. (3).

3 The Gaussian BGe scoring metric for dynamic Bayesian networks

We now consider the case that instead of independent observations, time series data have been collected for the domain: $(X_1(t), \ldots X_N(t))_{t=1,\ldots,m}$, and that we have a (1st-order) Markovian dependence structure. In this case, dynamic Bayesian networks (DBNs) can be employed. In DBNs each edge corresponds to an interaction with a time delay τ; e.g. for $\tau = 1$ an edge pointing from X_i to X_j means that the realization $x_j(t)$ of X_j at time point t is influenced by the realization $x_i(t-1)$ of X_i at the previous time point $t-1$. This can be taken into consideration in the context of the Gaussian BGe model by building new data matrices – one for each domain variable – from the original data matrix of size N-by-m given in Eq. (1). For dynamic data the columns do not represent independent (steady-state) observations: the tth column of $\mathcal{D}$ is the realization of the variables at time point t ($t = 1, \ldots, m$). We note that the score equivalence aspect of the BGe model is not required for dynamic Bayesian networks, because edge reversals are not permissible. However, formulating the models in terms of the BGe score is advantageous in case one intends to adapt the framework proposed in the main paper to non-linear static Bayesian networks along the lines of [8].

In principle, there are two alternatives which can be used, and it depends on whether or not 'direct feedback-loops', that is edges having the same node as starting and end point, should be allowed in the network. Here, we allow for 'direct feedback-loops', and we build the following N matrices of size $(N+1)$-by-$(m-1)$ from the (time series) data matrix given in Eq. (1) :

$$\mathcal{D}(n) = \begin{pmatrix} \mathcal{D}_{1,1} & \mathcal{D}_{1,2} & \ldots & \mathcal{D}_{1,m-1} \\ \mathcal{D}_{2,1} & \mathcal{D}_{2,2} & \ldots & \mathcal{D}_{2,m-1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{D}_{N,1} & \mathcal{D}_{N,2} & \ldots & \mathcal{D}_{N,m-1} \\ \mathcal{D}_{n,2} & \mathcal{D}_{n,3} & \ldots & \mathcal{D}_{n,m} \end{pmatrix} \tag{9}$$

$n = 1, \ldots, N$. That is, we obtain $\mathcal{D}(n)$ by deleting the last column of $\mathcal{D}$ and adding a novel row $(\mathcal{D}_{n,2}, \ldots, \mathcal{D}_{n,m})$, i.e. the nth row of $\mathcal{D}$ shifted leftwards by 1, as the $(N+1)$-th row. We can identify the $(N+1)$-th row with a new domain variable X_{N+1}. This new variable is the nth domain variable with a time shift of size $\tau = 1$. We note that the novel data matrices $D(n)$ consist of observations for $N+1$ domain variables, i.e. the hyperparameters T_0 and μ_0 are of the form of an $(N+1)$-by-$(N+1)$ matrix and an $(N+1)$-by-1 column vector, respectively. As before we can compute the matrix $T_{\mathcal{D}(n)}$ for each data set $\mathcal{D}(n)$, and we replace Eq. (7) by:

$$P(\mathcal{D}|\mathcal{G}) = \prod_{n=1}^{N} \Psi(\mathcal{D}_n^{\pi_n}, \mathcal{G}) = \prod_{n=1}^{N} \frac{P(\mathcal{D}(n)^{\{X_{N+1}, \pi_n\}}|\mathcal{G}_F(\{X_{N+1}, \pi_n\})}{P(\mathcal{D}(n)^{\{\pi_n\}}|\mathcal{G}_F(\pi_n))} \tag{10}$$

and Eq. (8) by:

$$\begin{aligned} P(\mathcal{D}(n)^S|\mathcal{G}_F(S)) \quad &= \quad (2\pi)^{-\frac{N^\dagger \cdot (m-1)}{2}} \cdot \left\{ \frac{v}{v+(m-1)} \right\}^{N^\dagger/2} \cdot \frac{c(N^\dagger, \alpha)}{c(N^\dagger, \alpha+(m-1))} \\ &\quad \cdot det(T_0^S)^{\frac{\alpha}{2}} \cdot det(T^S_{\mathcal{D}(i),(m-1)})^{-\frac{\alpha+(m-1)}{2}} \end{aligned}$$

where $\mathcal{G}_F(S)$ is a full graph for the domain variable subset S of cardinality $N^\dagger$ and T_0^S and $T^S_{\mathcal{D}(n),(m-1)}$ are sub-matrices as explained in Section 2.

If we have d independent (time series) data sets $\mathcal{D}^1, \ldots, \mathcal{D}^d$ where $\mathcal{D}^w$ is an N-by-m_w matrix consisting of m_w time-dependent realizations of the N variables and $w = 1, \ldots, d$, then we can build the N matrices: $\mathcal{D}^w(i)$ of dimension $(N+1)$-by-(m_w-1), $i = 1, \ldots, N$, independently for each w, $w = 1, \ldots, d$, using Eq. (9). Afterwards we can merge the d data sets $\mathcal{D}^1(n), \ldots, \mathcal{D}^d(n)$ column-wise to one single data set: $D^{ALL}(n) = (\mathcal{D}^1(n), \ldots, \mathcal{D}^d(n))$ of dimension $(N+1)$-by-$\sum_{w=1}^{d}(m_w - 1)$ for each variable $n = 1, \ldots, N$. Using the combined data set $D^{ALL}(n)$ for

computing local scores of the variable X_n ensures that the realization of X_n at the first time point $t = 1$ of the wth data set segment, symbolically: $\mathcal{D}^w_{i,1}$, has no relation with the last realizations of its parent nodes π_n in the preceding data segment $\mathcal{D}^{w-1}$, symbolically: $\mathcal{D}^{w-1}_{\pi_n, m_{(w-1)}-1}$. That is, by adding shifted rows as the $(N+1)$th row to each data segment $\mathcal{D}^w$ *independently* with Eq. (9) before merging the resulting data sets it is taken into account that 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 data segment. Therefore, as there are no parent node realizations 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 Eqn. (2) and (3) of the main paper have to be replaced by:

$$P(\mathcal{D}|\mathcal{G}) \;=\; \int P(\mathcal{D}|\mathcal{G},\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathcal{G})d\boldsymbol{\theta} \;=\; \prod_{n=1}^{N} \Psi(\mathcal{D}^{\pi_n}_n, \mathcal{G}) \tag{11}$$

$$\Psi(\mathcal{D}^{\pi_n}_n, \mathcal{G}) \;=\; \int \prod_{w=1}^{d}\prod_{t=2}^{m_w} P\Big(X_n(t) = \mathcal{D}^w_{n,t} | \pi_n(t-1) = \mathcal{D}^w_{(\pi_n,t-1)}, \boldsymbol{\theta}_n\Big) P(\boldsymbol{\theta}_n|\mathcal{G}) d\boldsymbol{\theta}_n \tag{12}$$

where $\mathcal{D}^{\pi_n}_n := \{(\mathcal{D}^w_{n,t}, \mathcal{D}^w_{\pi_n,t-1}) : 2 \le t \le m_w, 1 \le w \le d\}$ consists of the subsets of the d data segments pertaining to node X_n and parent set π_n.

This framework can straightforwardly be applied to the cpBGe model, in that columns of the matrix $D^{ALL}(n)$ are allocated to different components of the underlying mixture model via a change-point process. As a very simple illustration, consider two time series $\{X(1), X(2), X(3), X(4)\}$ and $\{\tilde{X}(1), \tilde{X}(2), \tilde{X}(3)\}$, which we want to concatenate. We consider a simple DBN consisting of only one domain node X with a feedback loop back onto itself. Matrix $D^{ALL}(n)$ is given by

$$D^{ALL}(n) \;=\; \begin{pmatrix} X(1) & X(2) & X(3) & \tilde{X}(1) & \tilde{X}(2) \\ X(2) & X(3) & X(4) & \tilde{X}(2) & \tilde{X}(3) \end{pmatrix} \tag{13}$$

where we note that the column $\begin{pmatrix} X(4) \\ \tilde{X}(1) \end{pmatrix}$ has to be excluded, as explained above. For a two component mixture model, the columns of this matrix are assigned to one of two components via a change-point process. Hence, we get the following sub-matrices: $\left\{ \begin{pmatrix} X(1) \\ X(2) \end{pmatrix}, \begin{pmatrix} X(2) & X(3) & \tilde{X}(1) & \tilde{X}(2) \\ X(3) & X(4) & \tilde{X}(2) & \tilde{X}(3) \end{pmatrix} \right\}, \left\{ \begin{pmatrix} X(1) & X(2) \\ X(2) & X(3) \end{pmatrix}, \begin{pmatrix} X(3) & \tilde{X}(1) & \tilde{X}(2) \\ X(4) & \tilde{X}(2) & \tilde{X}(3) \end{pmatrix} \right\},$

$\ldots, \left\{ \begin{pmatrix} X(1) & X(2) & X(3) & \tilde{X}(1) \\ X(2) & X(3) & X(4) & \tilde{X}(2) \end{pmatrix}, \begin{pmatrix} \tilde{X}(2) \\ \tilde{X}(3) \end{pmatrix} \right\}$

More general and to be consistent with the mathematical notations that were used in the main paper we note that we can alternatively treat these merged data sets $\mathcal{D}^{ALL}(n)$ ($n = 1, \ldots, N$) as if they were extracted from one single time series $\mathcal{D}^{ALL} = (\mathcal{D}^1, \ldots, \mathcal{D}^d)$ with $\sum_{w=1}^{d} m_w$ time points. It has then to be taken into account that the boundary time points $\mathcal{D}^w_{\cdot,m_w}$ and $\mathcal{D}^{w+1}_{\cdot,1}$ of two neighbouring data sets in the sequence $\mathcal{D}^{ALL} = (\mathcal{D}^1, \ldots, \mathcal{D}^d)$ are unrelated.

More generally and so as to be consistent with the mathematical notation that was used in the main paper, we note that we can alternatively treat these merged data sets $\mathcal{D}^{ALL}(n)$ ($n = 1, \ldots, N$) as if they were extracted from one single time series $\mathcal{D}^{ALL} = (\mathcal{D}^1, \ldots, \mathcal{D}^d)$ with $\sum_{w=1}^{d} m_w$ time points. It has then to be taken into account that the boundary time points $\mathcal{D}^w_{\cdot,m_w}$ and $\mathcal{D}^{w+1}_{\cdot,1}$ of two neighbouring data sets in the sequence $\mathcal{D}^{ALL} = (\mathcal{D}^1, \ldots, \mathcal{D}^d)$ are unrelated.

In terms of our *cpBGe* model this means that we have an allocation matrix $\mathbf{V}^{ALL}$ of latent variables $V^{ALL}_n(t)$ for $\mathcal{D}^{ALL} = (\mathcal{D}^1, \ldots, \mathcal{D}^d)$ where $V^{ALL}_n(t) = k$ means that the tth realization $\mathcal{D}^{ALL}_{n,t}$ of

X_n ($2 \leq t \leq \sum_{w=1}^{d} m_w$) is allocated to the kth mixture component. In this context we note that the realization of the tth time point in $\mathcal{D}^{ALL}$ corresponds to the sth realization in data segment $\mathcal{D}^q$ where

$$q = 1 + max\{u \in \{0, \ldots, d\} | t - \sum_{w=1}^{u} m_w > 0\} \tag{14}$$

and $s = t - \sum_{w=1}^{q} m_w$.

There are no realizations for the potential parent nodes of the first time points: $\mathcal{D}^1_{\cdot,1}, \ldots \mathcal{D}^d_{\cdot,1}$. Therefore, the time points $t \in \{1, (m_1 + 1), (m_1 + m_2 + 1), \ldots, (m_1 + m_2 + \ldots + m_{d-1} + 1)\}$, which correspond to the first points of the time series, are redundant in the allocation matrix $\mathbf{V}^{ALL}$ (and the latent variables $V_n^{ALL}(t)$ $(n = 1, \ldots, N)$). We therefore left these d realizations out, which reduces the number of columns of $\mathbf{V}^{ALL}$ to $m_{ALL} = \sum_{w=1}^{d}(m_w - 1)$.

4 Methodology

4.1 The dynamic BGe network (duplicated from the main paper)

DBNs are flexible models for representing probabilistic relationships between interacting variables (nodes) $X_1, \ldots, X_N$ via a directed graph $\mathcal{G}$. An edge pointing from X_i to X_j indicates that the realization of X_j at time point t, symbolically: $X_j(t)$, is conditionally dependent on the realization of X_i at time point $t-1$, symbolically: $X_i(t-1)$. 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 $\mathcal{G}$. Given a data set $\mathcal{D}$, where $\mathcal{D}_{n,t}$ and $\mathcal{D}_{(\pi_n,t)}$ are the tth realizations $X_n(t)$ and $\pi_n(t)$ of X_n and π_n, respectively, and $1 \leq t \leq m$ represents time, DBNs are based on the following homogeneous Markov chain expansion:

$$P(\mathcal{D}|\mathcal{G}, \boldsymbol{\theta}) = \prod_{n=1}^{N} \prod_{t=2}^{m} P\Big(X_n(t) = \mathcal{D}_{n,t} | \pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)}, \boldsymbol{\theta}_n\Big) \tag{15}$$

where $\boldsymbol{\theta}$ is the total parameter vector, composed of node-specific subvectors $\boldsymbol{\theta}_n$, which specify the local conditional distributions in the factorization. From Eq. (15) and under the assumption of parameter independence, $P(\boldsymbol{\theta}|\mathcal{G}) = \prod_n P(\boldsymbol{\theta}_n|\mathcal{G})$, the marginal likelihood is given by

$$P(\mathcal{D}|\mathcal{G}) = \int P(\mathcal{D}|\mathcal{G}, \boldsymbol{\theta}) P(\boldsymbol{\theta}|\mathcal{G}) d\boldsymbol{\theta} = \prod_{n=1}^{N} \Psi(\mathcal{D}_n^{\pi_n}, \mathcal{G}) \tag{16}$$

$$\Psi(\mathcal{D}_n^{\pi_n}, \mathcal{G}) = \int \prod_{t=2}^{m} P\Big(X_n(t) = \mathcal{D}_{n,t} | \pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)}, \boldsymbol{\theta}_n\Big) P(\boldsymbol{\theta}_n|\mathcal{G}) d\boldsymbol{\theta}_n \tag{17}$$

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 choose a linear Gaussian distribution for the local conditional distribution $P(X_n|\pi_n, \boldsymbol{\theta}_n)$ in Eq.(15). Under fairly weak regularity conditions discussed in [1] (parameter modularity and conjugacy of the prior[1]), the integral in Eq. (17) has a closed form solution, given by Eq. (24) in [1]. The resulting expression is called the BGe score[2].

[1] The conjugate prior is a normal-Wishart distribution. For the present study, we chose the hyperparameters of this distribution maximally uninformative subject to the regularity conditions discussed in [1].

[2] The score equivalence aspect of the BGe model is not required for DBNs, because edge reversals are not permissible. However, formulating our method in terms of the BGe score is advantageous when adapting the proposed framework to non-linear static Bayesian networks along the line of [8].

4.2 The non-stationary dynamic change-point BGe model (cpBGe) (duplicated from the main paper)

To obtain a non-stationary DBN, we generalize Eq. (15) with a node-specific mixture model:

$$P(\mathcal{D}|\mathcal{G},\mathbf{V},\mathbf{K},\boldsymbol{\theta}) = \prod_{n=1}^{N}\prod_{t=2}^{m}\prod_{k=1}^{\mathcal{K}_n} P\Big(X_n(t)=\mathcal{D}_{n,t}|\pi_n(t-1)=\mathcal{D}_{(\pi_n,t-1)},\boldsymbol{\theta}_n^k\Big)^{\delta_{V_n(t),k}} \tag{18}$$

where $\delta_{V_n(t),k}$ is the Kronecker delta, $\mathbf{V}$ is a matrix of latent variables $V_n(t)$, $V_n(t)=k$ indicates that the realization of node X_n at time t, $X_n(t)$, has been generated by the kth component of a mixture with $\mathcal{K}_n$ components, and $\mathbf{K}=(\mathcal{K}_1,\ldots,\mathcal{K}_n)$. 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 break-points. The probability model defined in Eq.(18) 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. In the present work, we change the assignment of data points to mixture components from a free allocation to a change-point process[3]. This effectively reduces the complexity of the latent variable space and incorporates our prior belief that, in a time series, adjacent time points are likely to be assigned to the same component. From Eq. (18), the marginal likelihood conditional on the latent variables $\mathbf{V}$ is given by

$$P(\mathcal{D}|\mathcal{G},\mathbf{V},\mathbf{K}) = \int P(\mathcal{D}|\mathcal{G},\mathbf{V},\mathbf{K},\boldsymbol{\theta})P(\boldsymbol{\theta})d\boldsymbol{\theta} \;=\; \prod_{n=1}^{N}\prod_{k=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n}[k,\mathbf{V}_n],\mathcal{G}) \tag{19}$$

$$\Psi(\mathcal{D}_n^{\pi_n}[k,\mathbf{V}_n],\mathcal{G}) = \int \prod_{t=2}^{m} P\Big(X_n(t)=\mathcal{D}_{n,t}|\pi_n(t-1)=\mathcal{D}_{(\pi_n,t-1)},\boldsymbol{\theta}_n^k\Big)^{\delta_{V_n(t),k}} P(\boldsymbol{\theta}_n^k|\mathcal{G})d\boldsymbol{\theta}_n^k \tag{20}$$

Eq. (20) is similar to Eq. (17), except that it is restricted to the subset $\mathcal{D}_n^{\pi_n}[k,\mathbf{V}_n] := \{(\mathcal{D}_{n,t},\mathcal{D}_{\pi_n,t-1}) : V_n(t)=k, 2\le t\le m\}$. Hence when the regularity conditions defined in [1] are satisfied, then the expression in Eq.(20) has a closed-form solution: it is given by Eq. (24) in [1] restricted to the subset of the data that has been assigned to the kth mixture component (or kth segment). The joint probability distribution of the proposed cpBGe model is given by:

$$\begin{aligned} P(\mathcal{G},\mathbf{V},\mathbf{K},\mathcal{D}) &= P(\mathcal{D}|\mathcal{G},\mathbf{V},\mathbf{K})\cdot P(\mathcal{G})\cdot P(\mathbf{V}|\mathbf{K})\cdot P(\mathbf{K}) \\ &= P(\mathcal{G})\cdot\prod_{n=1}^{N}\left\{\{P(\mathbf{V}_n|\mathcal{K}_n)\cdot P(\mathcal{K}_n)\cdot\prod_{k=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n}[k,\mathbf{V}_n],\mathcal{G})\right\} \end{aligned} \tag{21}$$

In the absence of genuine prior knowledge about the regulatory network structure, we assume for $P(\mathcal{G})$ a uniform distribution on graphs, subject to a fan-in restriction of $|\pi_n|\le 3$. As prior probability distributions on the node-specific numbers of mixture components $\mathcal{K}_n$, $P(\mathcal{K}_n)$, we take iid truncated Poisson distributions with shape parameter $\lambda=1$, restricted to $1\le\mathcal{K}_n\le\mathcal{K}_{MAX}$ (we set $\mathcal{K}_{MAX}=10$ in our simulations). The prior distribution on the latent variable vectors, $P(\mathbf{V}|\mathbf{K})=\prod_{n=1}^{N}\{P(\mathbf{V}_n|\mathcal{K}_n)$, is implicitly defined via the change-point process as follows. We identify $\mathcal{K}_n$ with $\mathcal{K}_n-1$ change-points $\mathbf{b}_n=\{b_{n,1},\ldots,b_{n,\mathcal{K}_n-1}\}$ on the continuous interval $[2,m]$. For notational convenience we introduce the pseudo change-points $b_{n,0}=2$ and $b_{n,\mathcal{K}_n}=m$. For node X_n the observation at time point t is assigned to the kth component, symbolically $V_n(t)=k$, if $b_{n,k-1}\le t<b_{n,k}$. Following [4] we assume that the change-points 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 interval $[2,m]$. The motivation for this prior, instead of taking $\mathcal{K}_n$ uniformly distributed points, is to encourage *a priori* an equal spacing between the change-points, i.e. to discourage mixture components (i.e. segments) that contain only a few observations. The even-numbered order statistics prior on the change-point locations $\mathbf{b}_n$ induces a prior distribution on the node-specific allocation vectors $\mathbf{V}_n$. Deriving a closed-form expression is involved. However, the MCMC scheme we discuss in the next section does not sample $\mathbf{V}_n$ directly, but is based on local modifications of $\mathbf{V}_n$ based on birth, death and reallocation moves. All that is required for the acceptance probabilities of these moves are $P(\mathbf{V}_n|\mathcal{K}_n)$ ratios, which are straightforward to compute.

[3] This implies that we propose a non-stationary rather than a proper non-linear model.

4.3 MCMC inference (extended version of the main paper)

We now describe an MCMC algorithm to obtain a sample $\{\mathcal{G}^i, \mathbf{V}^i, \mathbf{K}^i\}_{i=1,\ldots,I}$ from the posterior distribution $P(\mathcal{G}, \mathbf{V}, \mathbf{K}|\mathcal{D}) \propto P(\mathcal{G}, \mathbf{V}, \mathbf{K}, \mathcal{D})$ of Eq. (21). We combine the structure MCMC algorithm[4] [3, 9] with the change-point model used in [4], and draw on the fact that conditional on the allocation vectors $\mathbf{V}$, the model parameters can be integrated out to obtain the marginal likelihood terms $\Psi(\mathcal{D}_n^{\pi_n}[k, \mathbf{V}_n], \mathcal{G})$ in closed form, as shown in the previous section. Note that this approach is equivalent to the idea underlying the allocation sampler proposed in [**?**]. The resulting algorithm is effectively an RJMCMC scheme [4] 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 structure MCMC move on the current graph $\mathcal{G}^i$ and leave the latent variable matrix and the numbers of mixture components unchanged, symbolically: $\mathbf{V}^{i+1} = \mathbf{V}^i$ and $\mathbf{K}^{i+1} = \mathbf{K}^i$. A new candidate graph $\mathcal{G}^{i+1}$ is randomly drawn out of the set of graphs $\mathcal{N}(\mathcal{G}^i)$ that can be reached from the current graph $\mathcal{G}^i$ by deletion or addition of a single edge. The proposed graph $\mathcal{G}^{i+1}$ is accepted with probability:

$$A(\mathcal{G}^{i+1}|\mathcal{G}^i) = min\left\{1, \frac{P(\mathcal{D}|\mathcal{G}^{i+1}, \mathbf{V}^i, \mathbf{K}^i)}{P(\mathcal{D}|\mathcal{G}^i, \mathbf{V}^i, \mathbf{K}^i)} \frac{P(\mathcal{G}^{i+1})}{P(\mathcal{G}^i)} \frac{|\mathcal{N}(\mathcal{G}^i)|}{|\mathcal{N}(\mathcal{G}^{i+1})|}\right\} \tag{22}$$

where $|.|$ is the cardinality, and the marginal likelihood terms have been specified in Eq. (19). The graph is left unchanged, symbolically $\mathcal{G}^{i+1} := \mathcal{G}^i$, if the move is not accepted.

With the complementary probability $1 - p_G$ we leave the graph $\mathcal{G}^i$ unchanged and perform a move on $(\mathbf{V}^i, \mathbf{K}^i)$, where $\mathbf{V}_n^i$ is the latent variable vector of X_n in $\mathbf{V}^i$, and $\mathbf{K}^i = (\mathcal{K}_1^i, \ldots, \mathcal{K}_N^i)$. We randomly select a node X_n and change its current number of components $\mathcal{K}_n^i$ via a change-point birth or death move, or its latent variable vector $\mathbf{V}_n^i$ by a change-point re-allocation move. The change-point birth (death) move increases (decreases) $\mathcal{K}_n^i$ by 1 and may also have an effect on $\mathbf{V}_n^i$. The change-point reallocation move leaves $\mathcal{K}_n^i$ unchanged and may have an effect on $\mathbf{V}_n^i$. Under fairly mild regularity conditions (ergodicity), the MCMC sampling scheme converges to the desired posterior distribution if the acceptance probabilities for the three change-point moves $(\mathcal{K}_n^i, \mathbf{V}_n^i) \rightarrow (\mathcal{K}_n^{i+1}, \mathbf{V}_n^{i+1})$ are chosen of the form $min(1, R)$, see [4], with

$$R = \frac{\prod_{k=1}^{\mathcal{K}_n^{i+1}} \Psi(\mathcal{D}_n^{\pi_n}[k, \mathbf{V}_n^{i+1}], \mathcal{G})}{\prod_{k=1}^{K_n^i} \Psi(\mathcal{D}_n^{\pi_n}[k, \mathbf{V}_n^i], \mathcal{G})} \times A \times B \tag{23}$$

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, and B is the inverse proposal probability ratio. The exact form of these factors depends on the move type and the formulae were not provided in the main paper:

(i) For a change-point reallocation[5] (r) we randomly select one of the existing change-points $b_{n,j} \in \{b_{n,1}, \ldots, b_{n,\mathcal{K}_n-1}\}$, and the replacement value $b_{n,j}^\dagger$ is drawn from a uniform distribution on $[b_{n,j-1}, b_{n,j+1}]$ where $b_{n,0} = 2$ and $b_{n,\mathcal{K}_n} = m$. Hence, the proposal probability ratio is one, the prior probabilities $P(\mathcal{K}_n^{i+1}) = P(\mathcal{K}_n^i)$ cancel out, and the remaining prior probability ratio $P(\mathbf{V}_n^{i+1}|\mathcal{K}_n^{i+1})/P(\mathbf{V}_n^i|\mathcal{K}_n^i)$ can be obtained from p.720 in [4]:

$$A_r = \frac{(b_{n,j+1} - b_{n,j}^\dagger)(b_{n,j}^\dagger - b_{n,j-1})}{(b_{n,j+1} - b_{n,j})(b_{n,j} - b_{n,j-1})}, \quad B_r = 1 \tag{24}$$

If there is no change-point ($\mathcal{K}_n^i = 1$) the move is rejected and the Markov chain is left unchanged. (ii) If a change-point birth move (b) on $\mathcal{K}_n^i$ is proposed, the location of the new change-point $b^\dagger$ is randomly drawn from a uniform distribution on the interval $[2, m]$; the proposal probability for this move is $b_{\mathcal{K}_n^i}/(m-2)$, where $b_{\mathcal{K}_n^i}$ is the ($\mathcal{K}_n^i$-dependent) probability of selecting a birth move. The reverse death move, which is selected with probability $d_{(\mathcal{K}_n^i+1)}$, consists in discarding randomly one

[4]The MCMC algorithm based on Eq.(10) in [**?**] is computationally less efficient than when applied to static DBNs, since the local scores have to be re-computed every time the positions of the change-points change.

[5]This move is chosen with probability $1 - b_{\mathcal{K}_n^i} - d_{\mathcal{K}_n^i}$, where $b_{\mathcal{K}_n^i}$ and $d_{\mathcal{K}_n^i}$ are defined below.

of the $(\mathcal{K}_n^i + 1) - 1 = \mathcal{K}_n^i$ change-points. The inverse proposal probability ratio is thus given by $B = d_{(\mathcal{K}_n^i+1)}(m-2)/b_{\mathcal{K}_n^i}\mathcal{K}_n^i$. The prior probability ratio is given by the first three factors in the expression at the bottom of p.720 in [4] (slightly modified to allow for the fact that $\mathcal{K}_n$ components correspond to $\mathcal{K}_n - 1$ change-points), and we get:

$$A_b = \frac{P(\mathcal{K}_n^i + 1)}{P(\mathcal{K}_n^i)} \frac{2\mathcal{K}_n^i(2\mathcal{K}_n^i + 1)}{(m-2)^2} \frac{(b_{n,j+1} - b^\dagger)(b^\dagger - b_{n,j})}{(b_{n,j+1} - b_{n,j})},\ B_b = \frac{d_{(\mathcal{K}_n^i+1)}(m-2)}{b_{\mathcal{K}_n^i}\mathcal{K}_n^i} \tag{25}$$

For $\mathcal{K}_n^i = \mathcal{K}_{MAX}$ the birth of a new change-point is invalid and the Markov chain is left unchanged. Note that the ratio of the proposal probabilities for birth versus death moves $d_{(\mathcal{K}_n^i+1)}/b_{\mathcal{K}_n^i}$ can be chosen such that it cancels out against the prior ratio $P(\mathcal{K}_n^i + 1)/P(\mathcal{K}_n^i)$, and the expression simplifies:

$$A_b B_b = \frac{2(2\mathcal{K}_n^i + 1)}{(m-2)} \frac{(b_{n,j+1} - b^\dagger)(b^\dagger - b_{n,j})}{(b_{n,j+1} - b_{n,j})} \tag{26}$$

(iii) A change-point death move (d) is the reverse of the birth move, and we get:

$$A_d B_d = \frac{(m-2)}{2(2\mathcal{K}_n^i - 1)} \frac{(b_{n,j+1} - b_{n,j})}{(b_{n,j+1} - b^\dagger)(b^\dagger - b_{n,j})} \tag{27}$$

5 Implementation of alternative Bayesian network methods included in our comparative benchmark study

The generalization of the BGe model of Geiger and Heckerman [1] to dynamic Bayesian networks has been described in Section 3. In analogy, the static BDe model of Heckerman et al. [7] can be generalized for dynamic Bayesian networks; e.g. it corresponds to the non-stationary model in Robinson and Hartemink [10]. We include a slightly modified version of the BGM model of Grzegorczyk et al. (see [5]) in our comparison. The BGM model differs from our $cpBGe$ model in two aspects. First, the latent variable allocation is common to the whole network, that is, the change-points are not node-specific. Second, the assignment of data points to components is not effected by a change-point 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 iid data rather than time series. When combined with the node-specific change-points of the $cpBGe$ model, it will lead to a non-linear rather than non-stationary model, as we have discussed in the main paper. 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 be 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 have replaced the free allocation model originally used for BGM by the change-point process of our own model. In this way our comparison focuses on the aspect of employing node-specific rather than common change-points, that is, it allows us to investigate to what extent this additional model flexibility leads to an improved network reconstruction accuracy.

We now briefly describe the modified BGM model: An allocation vector $\vec{\mathcal{V}}$ of size $m-1$ describes the allocation of the time points $t = 2, \ldots, m$ to the $\mathcal{K}$ components, and $\mathcal{D}^{(\vec{\mathcal{V}},k)}$ denotes all realizations that are allocated to component k. For the joint posterior probability we get:

$$P(\mathcal{G}, \vec{\mathcal{V}}, \mathcal{K}|\mathcal{D}) \propto P(\mathcal{K})P(\vec{\mathcal{V}}|\mathcal{K})P(\mathcal{G})P(\mathcal{D}|\mathcal{G}, \vec{\mathcal{V}}, \mathcal{K}) \tag{28}$$

where

$$P(\mathcal{D}|\mathcal{G}, \vec{\mathcal{V}}, \mathcal{K}) = \prod_{k=1}^{\mathcal{K}} P(\mathcal{D}^{(\vec{\mathcal{V}},k)}|\mathcal{G}) \tag{29}$$

and each factor $P(\mathcal{D}^{(\vec{\mathcal{V}},k)}|\mathcal{G})$ corresponds to a subset of the data $\mathcal{D}^{(\vec{\mathcal{V}},k)}$ for which an independent BGe score can be computed. For $P(\mathcal{K})$ we take a truncated Poisson distribution with $\lambda = 1$ restricted to $1 \leq \mathcal{K} \leq \mathcal{K}_{MAX}$ with $\mathcal{K}_{MAX} = 10$. As in the $cpBGe$ model, we identify $\mathcal{K}$ with $\mathcal{K}-1$ breakpoints on the continuous interval $[2, m]$ and we assume that these breakpoints are distributed as the even-numbered order statistics of $2(\mathcal{K}-1)+1$ points uniformly distributed on $[2, m]$.

As described for the $cpBGe$ model in Section 2 of the main paper, the structure MCMC algorithm of [9] is then combined with the change-point model of [4]. With probability $p_G = 0.5$ a structure MCMC move is performed on the graph $\mathcal{G}$, and $\vec{\mathcal{V}}$ and $\mathcal{K}$ are left unchanged. A new candidate graph $\mathcal{G}^\dagger$ is randomly drawn out of the set of graphs $\mathcal{N}(\mathcal{G})$ that can be reached from the current graph $\mathcal{G}$ by deletion or addition of a single edge. The acceptance probability for a move from $\mathcal{G}$ to $\mathcal{G}^\dagger$ is given by $A = min(1, R)$, where

$$R = \frac{P(\mathcal{D}|\mathcal{G}^\dagger, \vec{\mathcal{V}}, \mathcal{K})}{P(\mathcal{D}|\mathcal{G}, \vec{\mathcal{V}}, \mathcal{K})} \cdot \frac{P(\mathcal{G}^\dagger)}{P(\mathcal{G})} \cdot \frac{|\mathcal{N}(\mathcal{G})|}{|\mathcal{N}(\mathcal{G}^\dagger)|} \tag{30}$$

and $|.|$ is the cardinality. With the complementary probability $1 - p_G$ a breakpoint birth, death, or re-allocation move is performed on $(\vec{\mathcal{V}}, \mathcal{K})$ and $\mathcal{G}$ is left unchanged. The acceptance probabilities for these moves $(\mathcal{K}, \vec{\mathcal{V}}) \rightarrow (\mathcal{K}^\dagger, \vec{\mathcal{V}}^\dagger)$ are of the same functional form: $A = min(1, R)$ where

$$R = \frac{P(\mathcal{D}|\mathcal{G}, \vec{\mathcal{V}}^\dagger, \mathcal{K}^\dagger)}{P(\mathcal{D}|\mathcal{G}, \vec{\mathcal{V}}, \mathcal{K})} \times c_M \tag{31}$$

c_M depends on the move type, and can easily be derived as we did for the $cpBGe$ algorithm in Subsection 4.3. In essence, each $\mathcal{K}_n$ has to be replaced by $\mathcal{K}$ in the corresponding equations.

Another non-linear model based on node-specific Gaussian mixture models has been proposed by Ko et al. [8]. 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 [11] for graph selection, which is only a good approximation to the marginal likelihood in the limit of large data sets. The BIC score of a graph $\mathcal{G}$ is defined as follows:

$$Score(\mathcal{G}) = \log(P(\mathcal{D}|\mathcal{G}, \widehat{\theta})) - \frac{1}{2}|\widehat{\theta}| \log m \tag{32}$$

where $\widehat{\theta}$ is the maximum likelihood estimate of the unknown parameters, and $|\widehat{\theta}|$ is the number of unknown parameters that have been estimated.

The Gaussian mixture model of [8], which we henceforth refer to as the GM_{BIC} model, is a node-specific mixture model with node-specific mixture weight parameters $\alpha_{n,k}$. Conditional on a fixed numbers of mixture components: $\mathbf{K} = (\mathcal{K}_1, \ldots, \mathcal{K}_n)$ the likelihood of the GM_{BIC} model factorizes as follows:

$$P(\mathcal{D}|\mathcal{G}, \mathbf{K}, \boldsymbol{\theta}) = \prod_{n=1}^{N} \prod_{t=2}^{m} \sum_{k=1}^{\mathcal{K}_n} \alpha_{n,k} P(X_n(t) = \mathcal{D}_{n,t} | \pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)}, \boldsymbol{\theta}_n^k) \tag{33}$$

The maximum likelihood estimate $\widehat{\boldsymbol{\theta}}$ for the mixture weights $\alpha_{n,k}$ and parameters $\boldsymbol{\theta}_n^k$ in the model of Ko et al. (see Eq. (33)) has no closed-form solution. Therefore, Ko et al. [8] apply the EM-algorithm to obtain a (local) maximum likelihood estimate $\widehat{\boldsymbol{\theta}_n^{k,\dagger}}$ and $\widehat{\alpha_{n,k,\dagger}}$ $(k = 1, \ldots, \mathcal{K}_n)$ for the N joint probability distributions:

$$\prod_{t=2}^{m} \sum_{k=1}^{\mathcal{K}_n} \alpha_{n,k,\dagger} P(X_n(t) = \mathcal{D}_{n,t}, \pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)}, \boldsymbol{\theta}_n^{k,\dagger}) \tag{34}$$

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 Eq. (34) 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, symbolically: $\widehat{\boldsymbol{\theta}_n^{k,\ddagger}} \subset \widehat{\boldsymbol{\theta}_n^{k,\dagger}}$, and the estimated mixture weights $\widehat{\alpha_{n,k,\ddagger}} := \widehat{\alpha_{n,k,\dagger}}$ $(k = 1, \ldots, \mathcal{K}_n)$ into the (marginal) likelihood:

$$\prod_{t=2}^{m} \sum_{k=1}^{\mathcal{K}_n} \alpha_{n,k,\ddagger} P(\pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)}, \boldsymbol{\theta}_n^{k,\ddagger}) \tag{35}$$

to obtain an approximate[6] estimate for the maximum likelihood value of the marginal probability distribution of the parent nodes in π_n. This can be done independently for all N factors (local distributions) in Eq. (33). Finally, from the definition of conditional probability distributions, Ko et al. obtain:

$$P(\mathcal{D}|\mathcal{G}, \mathbf{K}, \widehat{\boldsymbol{\theta}}) \quad = \quad \prod_{n=1}^{N} \prod_{t=2}^{m} \frac{\sum_{k=1}^{\mathcal{K}_n} \widehat{\alpha_{n,k,\dagger}} P(X_n(t) = \mathcal{D}_{n,t}, \pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)}, \widehat{\boldsymbol{\theta}_n^{k,\dagger}})}{\sum_{k=1}^{\mathcal{K}_n} \widehat{\alpha_{n,k,\dagger}} P(\pi_n(t-1) = \mathcal{D}_{(\pi_n,t-1)}, \widehat{\boldsymbol{\theta}_n^{k,\ddagger}})} \tag{36}$$

In essence, for each of the N local (conditional) probability distributions in Eq. (33) the parameters of the joint posterior probability distributions of X_n and π_n, symbolically: $\widehat{\alpha_{n,1,\dagger}}, \ldots, \widehat{\alpha_{n,\mathcal{K}_n,\dagger}}, \widehat{\boldsymbol{\theta}_n^{1,\dagger}}, \ldots, \widehat{\boldsymbol{\theta}_n^{\mathcal{K}_n,\dagger}}$, are maximized independently as parameters of a Gaussian mixture distribution by applying the EM-algorithm on the data subset:

$$\mathcal{D}(X_n, \pi_n) = \{(\mathcal{D}_{n,t}, \mathcal{D}_{\pi_n,t-1}) : 2 \leq t \leq m\} \tag{37}$$

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 mixture weights $\widehat{\alpha_{n,k,\dagger}}$ unchanged $(k = 1, \ldots, \mathcal{K}_n)$.

The number of estimated parameters is given by:

$$|\widehat{\theta}(\mathcal{G}, \mathbf{K})| = \mathcal{K}_n - 1 + \sum_{n=1}^{N} ((|\pi_n| + 1) + (|\pi_n| + 2) \cdot (|\pi_n| + 1)/2)) \cdot \mathcal{K}_n \tag{38}$$

where $\mathbf{K} = (\mathcal{K}_1, \ldots, \mathcal{K}_n)$ are the numbers of mixture 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_{BIC} score of a graph $\mathcal{G}$ is then given by:

$$S(\mathcal{G}|GM_{BIC}) = max \left\{ \log(P(\mathcal{D}|\mathcal{G}, \mathbf{K}, \widehat{\theta})) - \frac{1}{2} |\widehat{\theta}(\mathcal{G}, \mathbf{K})| \log m : \mathbf{K} = (\mathcal{K}_1, \ldots, \mathcal{K}_n) \right\} \tag{39}$$

whereby the numbers of mixture components, that is the N elements in the vector $\mathbf{K}$, can be restricted: $1 \leq \mathcal{K}_n \leq \mathcal{K}_{MAX}$, and $P(\mathcal{D}|\mathcal{G}, \mathbf{K}, \widehat{\theta})$ was defined in Eq. (36). We set $\mathcal{K}_{MAX} = 10$, and the GM_{BIC} estimator of the network structure is given by the graph $\mathcal{G}^\star$ with the highest score:

$$S(\mathcal{G}^\star|GM_{BIC}) \geq S(\mathcal{G}|GM_{BIC}) \tag{40}$$

for all possible graphs $\mathcal{G}$. We note that the computational costs for inferring the GM_{BIC} estimator can be drastically reduced by determining the best parent node set π_n for each of the N domain variables X_n independently, and joining the resulting 'subnetworks' to generate a single comprehensive network for the whole domain. This holds true for dynamic Bayesian networks, where the acyclicity constraint is satisfied by construction, but was also made as a heuristic assumption in [8], where it can potentially violate the acyclicity assumption.

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

6 Synthetic data

To assess the performance of the proposed $cpBGe$ model, we applied it to synthetic data generated from four different network structures shown in Figure 1 of the main paper.

Figure 1a in the main paper shows the smallest synthetic network that we considered. It consists of two domain nodes X and Y, and there are two edges, namely a feedback-loop $X \rightarrow X$, leading to autocorrelation in the time series $X(.)$, and a second edge from X to Y, which was modelled by a piecewise linear process with changing (time-dependent) coefficient $\beta(t)$:

$$\begin{aligned} X(t+1) &= \sqrt{1-\varepsilon^2} \cdot X(t) + \varepsilon \cdot \phi_X(t+1) \quad (41) \\ Y(t+1) &= \beta(t) \cdot X(t) + c \cdot \phi_Y(t+1) \quad (42) \end{aligned}$$

where $\varepsilon \in [0,1]$, and $\phi_X(1), \phi_X(2), \ldots, \phi_Y(1), \phi_Y(2), \ldots$ are iid Normally distributed variables.

Eq. (41) 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 iid standard Normally distributed variables, symbolically: $X(t+1) = \phi_X(t+1)$. For $\varepsilon = 0$ we obtain a process $X(.)$ which is constant in time, symbolically: $X(t+1) = X(t)$ for all t without any noise injections. We initialize $X(1)$ with a random realization from a standard Normal variable. Then $X(.)$ is standard Normally distributed at each time point t, for each $\varepsilon \in [0,1]$.

From Eq. (42) 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 ($2 \leq t \leq 11$) and the last ($32 \leq t \leq 41$) ten observations and $\beta(t) = -1$ for the 20 time points in between ($12 \leq t \leq 31$).

Moreover, 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 $\beta(1)X(1), \beta(2)X(2), \ldots$ 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:

$$c = \frac{\sigma(\widehat{\beta(t)X(t)})}{SNR} \quad (43)$$

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

The same idea can be used for generating data from the network shown in Figure 1b of the main paper. For this 4-node network domain we define:

$$\begin{aligned} X(t+1) &= \sqrt{1-\varepsilon^2} \cdot X(t) + \varepsilon \cdot \phi_X(t+1) \\ Y(t+1) &= \beta_Y(t) \cdot X(t) + c_Y \cdot \phi_Y(t+1) \\ W(t+1) &= \beta_W(t) \cdot X(t) + c_W \cdot \phi_W(t+1) \\ Z(t+1) &= \beta_Z(t) \cdot X(t) + c_Z \cdot \phi_Z(t+1) \quad (44) \end{aligned}$$

where all noise terms $\phi_.(.)$ are iid standard Normally distributed 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 a 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 change-point locations afterwards. For each of the three variables we independently draw the change-point location(s) from uniform distributions (i) over the discrete interval $\{6, \ldots, 36\}$ to avoid change-points during the first/last five time points, and (ii) under the constraint that there are at least 5 time points between the two change-point 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.:

$$c_Y = \frac{\sigma(\beta_Y \widehat{(t)} X(t))}{SNR} \tag{45}$$

where SNR is the specified signal-to-noise ratio and $\sigma(\beta_Y \widehat{(t)} X(t))$ can be estimated from pre-simulated data.

The same idea can also be used to generate synthetic data for the (slightly-modified) RAF-pathway shown in Figure 1c of the main paper. Node 'PIP3' has a recurrent feedback loop:

$$PIP3(t+1) = \sqrt{1-\varepsilon^2} \cdot PIP3(t) + \varepsilon \cdot \phi_{PIP3}(t+1) \tag{46}$$

and 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 iid standard Normal distributions (noise injections). E.g. for 'PIP2':

$$PIP2(t+1) = \beta_{PIP3}(t) \cdot PIP3(t) + \beta_{PLCG}(t) \cdot PLCG(t) + c_{PIP2} \cdot \phi_{PIP2}(t+1) \tag{47}$$

For each node we flip a coin to determine whether its coefficients change their values once or twice, and we randomly draw the change-point locations independently for each domain node from discrete uniform distributions under the constraints (i) that there is no change-point among the first/last 5 observations and (ii) that there are at least 5 time points between change-points. Different from the regulatory mechanisms for the smaller domains in Figure 1a-b of the main paper, we sample new coefficients $\beta_{\cdot}$ at each change-point from continuous uniform distributions on the interval $[0.5, 2]$ and we flip a coin to determine the sign of the new coefficient (i.e. a change-point does not necessarily imply a change of sign of the coefficients.).

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:

$$c_{PIP2} = \frac{\sigma(\beta_{PIP3}(t) PIP3(t) \widehat{+ \beta_{PLCG}}(t) PLCG(t))}{SNR} \tag{48}$$

Finally, for the network structure shown in Figure 1c of the main paper we generated data using sinusoidal transfer functions. This leads to a stronger mismatch between the model and the data-generation mechanism. The details can be found in the main paper.

7 Simulations

In all our simulations, data were standardized to zero mean and marginal variance of 1 for all dimensions. For BGe, BGM, and our $cpBGe$ model the hyperparameters of the normal-Wishart prior were chosen as uninformative as possible subject to certain regulatory conditions discussed in Geiger and Heckerman [1]: $\vec{\mu_0} = (0, \ldots, 0)^T$ and $W = I_{N+1}$, where $\vec{\mu_0}$ is an $(N+1)$-dimensional column vector and I_{N+1} is the $(N+1)$-by-$(N+1)$ identity matrix. The total prior precision parameters were set to: $\alpha = 1$ and $v = N + 3$, where N is the number of domain variables (nodes). As described in Section 3 we have $(N+1)$-by-$(m-1)$ data matrices in a dynamic Bayesian networks in which 'direct feedback-loops' are allowed; hence the covariance matrices are of size $(N+1)$-by-$(N+1)$. The 'effective' number of nodes is $N+1$; see Section 3 for more details.

For the BDe model of Heckerman et al. [7] the hyperparameters of the Dirichlet prior were also specified as uninformative as possible, as in Giudici and Castelo [3]. That is the total prior precision α was set to 1, and we set $\alpha_{i,j,k} = \frac{\alpha}{r_i \cdot q_i}$, where r_i is the number of possible values for the ith domain

node and q_i is the number of possible discrete realizations that the parent nodes π_i of the ith node can take on.

For the smaller (bigger) network domains we set both the burn-in and the sampling-phase lengths of our MCMC simulations to 50,000 (500,000) each and sampled every 1,000 iterations during the sampling-phase. We note that even for the bigger network domains with $N = 11$ (synthetic RAF-pathway data) and $N = 9$ (real Arabidopsis data) nodes each single MCMC simulation for the proposed $cpBGe$ Bayesian network model was accomplished within few hours using Matlab© code on a SunFire X4100M2 machine with MAD Opteron 2224 SE dual-core processor. We applied the standard diagnostic based on trace plots (see [3]) and the potential scale reduction factor (see [2]) to assure that in this way a sufficient degree of convergence had been reached. That is, for several data sets from the RAF-pathway and for the *Arabidopsis thaliana* data set we started 5 independent MCMC simulations from different initializations on the same data set, and we computed the potential scale reduction factor (PSRF) based on the marginal edge posterior probabilities to monitor convergence. As we observed a sufficient degree of convergence for all these data sets ($PSFR < 1.2$), we reported only the results of the empty-seeded MCMC runs in the main paper.

For the evaluation of the results, we proceeded as follows. For the synthetic study based on the network domains shown in Figure 1 of the main paper, we computed the marginal posterior probabilities of the individual network edges. All MCMC schemes, which were applied to the conventional Bayesian network models (BDe and BGe), the non-homogeneous mixture Bayesian network model BGM, and the proposed $cpBGe$ model, output a sample of graphs from the posterior distribution. For each of these four methods, the marginal edge posterior probability can be estimated from the fraction of graphs in the MCMC sample that contain the edge of interest.

For a fair comparison, we applied the GM_{BIC} model of Ko et al. 10 times independently with different initializations. In essence, we initialized the k-means clustering algorithm by random realizations of $N(\mu, I_N)$ distributions, where I_N is the identity matrix and μ is a random expectation vector with entries sampled independently from continuous uniform distributions on $[-1, 1]$. The output of the k-means cluster algorithm was then used to initialize the EM-algorithm as described in Ko et al. [8]. Afterwards, we took for each individual edge the fraction of inferred GM_{BIC} graphs that contained the edge of interest as the score for this particular edge.

For all mixture models we restricted the maximal number of mixture components to $\mathcal{K}_{MAX} = 10$, a limit that was never reached in the simulations. The data discretization required for the multinomial BDe Bayesian network scoring metric was accomplished with the Information Bottleneck algorithm [6]. More precisely, we first applied quantile discretization to discretize each domain variable independently into 20 discrete levels. Afterwards the Information Bottleneck was run until each domain variable contained three discetization levels. We note that the Information Bottleneck algorithm merges, for each variable, neighboring discretization levels such that the pairwise information loss – in terms of the average mutual information between this variable and the others – is minimized. Therefore, 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 $MI(X(t), Y(t+1)$ and $MI(Y(t), X(t+1)$, and (ii) that recurrent feedback loops are valid in dynamic Bayesian networks so that for each domain variable X the pairwise mutual information between $X(t)$ and $X(t+1)$ has to be included, symbolically $MI(X(t), X(t+1)$.

We assessed the network reconstruction accuracy via the area under the ROC (receiver operator characteristic) curve: AUC; this is a standard criterion that has been applied in numerous related articles.

8 Additional figures of the empirical results

In this section we provide additional figures that could – due to space restrictions – not be included in the main paper. Figures 1 to 4 show AUC histograms for the synthetic data sets. For each network structure shown in Figure 1 of the main paper we chose various parameter settings, and in the main paper we summarized the results in terms of AUC scatter plots. In this supplementary paper, we present separate AUC histograms, for each parameter setting separately.

We further note that the conditions of the paired t-test applied in the main paper are not strictly satisfied, as the data have been generated under different conditions and are therefore not identically distributed. That is, for each of the four network structures and each of the five methods under comparison, we computed average AUC scores for various pre-specified parameter combinations. We then applied a two-sided paired t-test to test whether the average AUC scores over all h considered parameter combinations differed significantly. However, as different parameter combination give rise to different (average) AUC scores, we do not have a sample of identically distributed variables.

In this supplementary material we apply two-sided paired t-tests to each individual parameter combination. As the average AUC scores have been computed from 25 independent data instantiations for each individual parameter setting for the smaller networks in Figures 1a-c of the main paper, and from 5 independent data instantiations for each individual parameter setting in Figure 1d of the main paper, we can compute a p-value for each individual parameter setting. Figures 5 to 9 show the resulting p-values for each network structure, summarized as heatmatrices. As we considered $h = 20$ (networks in Figures 1a-b of the main paper), $h = 18$ (network in Figure 1c of the main paper) and $h = 15$ (network in Figure 1d of the main paper) different parameter combinations (hypotheses), we need to address the issue of multiple testing. To this end, we computed the overall p-value with a family-wise Bonferroni correction for each group of h pairwise tests. The resulting (two-sided paired t-test) p-values can be represented as heatmatrices, as shown in Figures 5-9. For each network structure there are four heatmatrices, in which the AUC scores of $cpBGe$ were compared with the AUC scores of the four competing models. The colors of the cells of the heatmatrices indicate whether $cpBGe$ performed better or worse than the corresponding competing method. In this context we distinguished between two significance levels: (1) significantly better or worse after Bonferroni correction for the number of parameter combinations, i.e. hypothesis to be tested: (corrected p-values: $p < 0.025/h$ or $p > (1 - 0.025/h)$ where $h \in \{15, 18, 20\}$), and (2) significantly better or worse without Bonferroni correction for multiple testing (uncorrected p-values: $p < 0.025$ or $p > 0.975$).

The heatmatrix representations in Figures 5-9 reveal a clear trend in favour of the proposed $cpBGe$ model. For example, in Figures 5 and 6 $cpBGe$ performs significantly better at the corrected level $p = 0.025/h$ than BGe and BDe for various parameter settings (e.g. see the block of black cells in the first three columns for $SNR = 100$, $SNR = 10$, and $SNR = 3$ in the corresponding four heatmatrices) while neither BGe nor BDe perform significantly better than $cpBGe$ for any parameter combination. The Gaussian mixture model GM_{BIC} of Ko et al. performs significantly worse than $cpBGe$ at the corrected level $p = 0.025/h$ 10 times for the network in Figure 1a of the main paper (see black cells in Figure 5) and 6 times for the network in Figure 1b of the main paper (see black cells in Figure 6) while it performs better than $cpBGe$ only once for these two domains with $N = 2$ and $N = 4$ nodes (see the white cell for $SNR = 100$ and $\varepsilon = 0.1$ in Figure 5). For the data from the network with a non-linear sinusoid process shown in Figure 1c of the main paper this trend becomes even more obvious: The GM_{BIC} model of Ko et al. performs significantly worse than $cpBGe$ at the corrected level $p = 0.025/h$ for 17 out of $h = 18$ considered parameter settings (see (b) panels in Figures 7 and 8). Moreover, we note that the proposed $cpBGe$ model is superior to the GM_{BIC} model at least at the uncorrected level $p = 0.025$ for the eighteenth parameter setting.

The BGM model of Grzegorczyk et al. and the proposed $cpBGe$ model both perform approximately equally well for the networks shown in Figures 1a and 1b of the main paper. That is, there are only 2 black cells and there is no white cell in the (a) panels of Figure 5 and Figure 6. However,

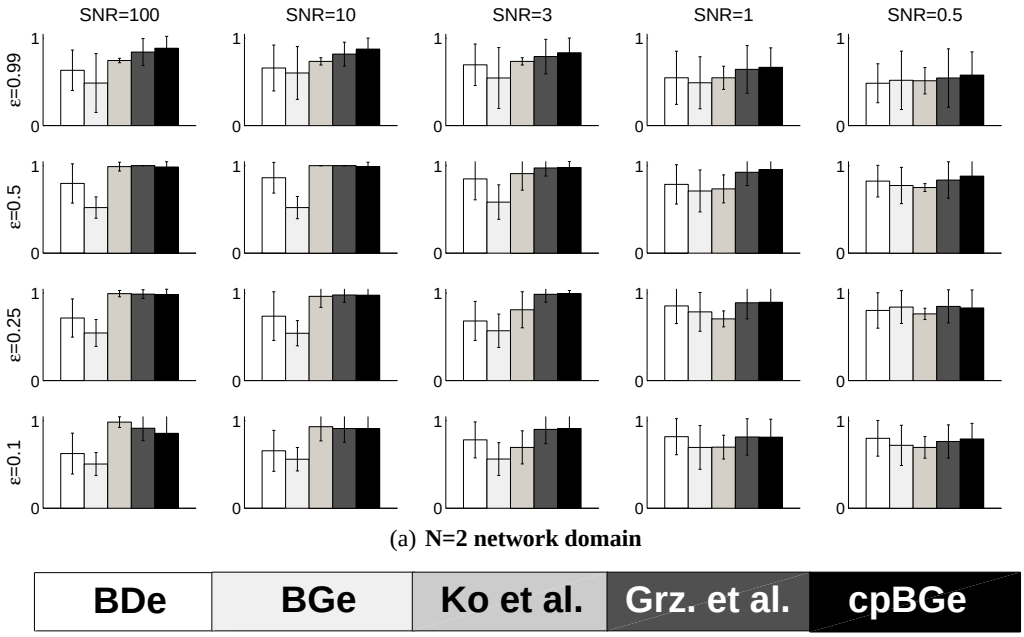
	Segment 1	Segment 2	Segment 3	Segment 4
Source	Mockler et al.(2007)	Edwards et al. (2006)	Grzegorcyk et al. (2008)	Grzegorcyk et al. (2008)
Time points	12	13	13	13
Time interval	4h	4h	2h	2h
Pretreatment entrainment	12h-light 12h-dark cycle	12h-light 12h-dark cycle	10h-light 10h-dark cycle	14h-light 14h-dark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

Table 1: Overview of the gene expression time series segments for Arabidopsis.

a certain trend in favour of $cpBGe$ is revealed by the heatmatrix in Figure 6 panel (a), as there are 9 further parameter combinations for which $cpBGe$ performs better at the uncorrected level $p = 0.025$ (dark grey cells) but only one single parameter combination ($\varepsilon = 0.99$ and SNR=3) where BGM performs better at the uncorrected level $p = 0.025$ (light grey cells). The corresponding three heatmatrices in Figures 7, 8, and 9 also reveal a trend in favour of $cpBGe$. That is, $cpBGe$ performs more often significantly better than BGM at the corrected level $p = 0.025/h$ (black cells) than vice-versa (white cells). In total there are 9 black cells but there is no white cell in the (a) panels of these three figures.

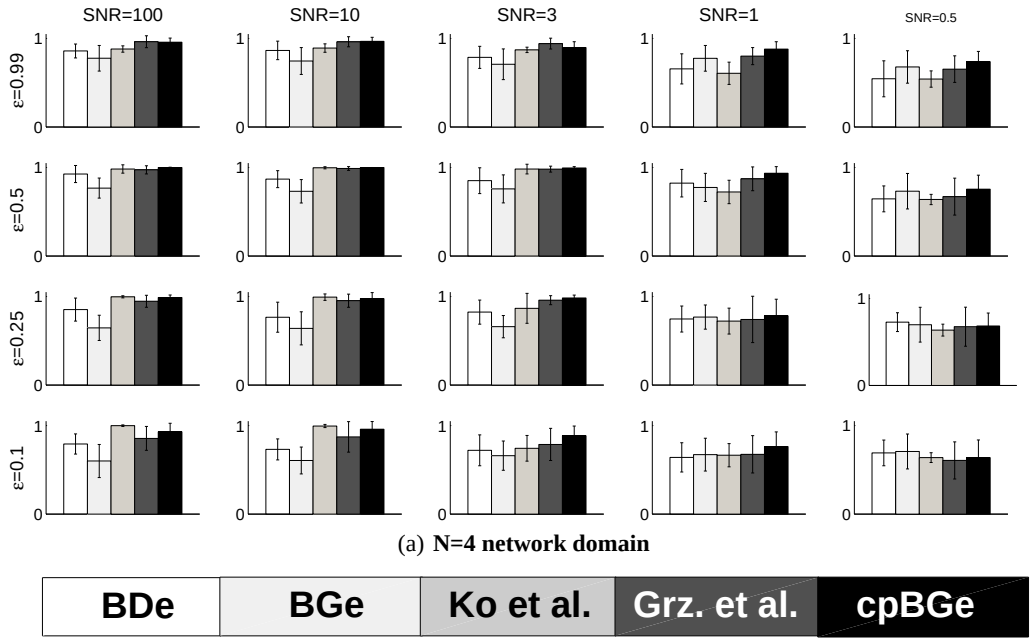
Finally, we provide further plots for the *Arabidopsis thaliana* time series gene expression data; see Figures 10-14. Figure 10 shows the time series obtained under the four experimental conditions listed in Table 1. Figure 11 shows the posterior probabilities of the change-point locations plotted against the time axis for the nine circadian genes. The dotted vertical lines indicate the true transition times (concatenation points) between the different experimental phases. For four of the genes (LHY, TOC1, PRR9 and PRR5), all known true change-points are correctly predicted. Genes PRR5 and PRR9 show various additional change-points; this might indicate that they are affected by additional heterogeneities beyond the four experimental phases. Four of the genes (CCA1, ELF3, GI, PRR3) show two change-points, at the true locations (ELF3, GI) or with a short time lag (CCA1, PRR3). For one gene (ELF4) only one change-point is predicted, at the location of the first true change-point between time series segments 1 and 2. When averaged over all nine genes, the three true change-points are correctly predicted (see Figure 3, top right panel of the main paper). A comparison of Table 1 with the locations of the peaks in Figure 11 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). While we are still seeking a biological corroboration of these predictions, Figure 11 demonstrates that the additional flexibility of the node-specific change-point model can be exploited as an exploratory tool for new hypothesis generation. Figures 12 and 13 show complementary representations. Figure 13 shows the posterior distribution of the number of components for each gene. For some genes the mode of this distribution is equal (TOC1) or close (LHY) to the chosen number of experimental phases (four). But there are also genes that display deviations. Note that the posterior distributions are consistent with the predicted change-points in Figure 11. For instance, gene ELF4, which shows only one predicted change-point in Figure 11, has a posterior distribution that peaks at two components. Gene PRR9, for which we found additional change-points in Figure 11, has a posterior distribution whose mode is shifted to a higher value, at 7 components. Again, we suggest that our node-specific change-point model provides a tool for biological hypothesis generation. When averaging over all node-specific posterior distributions, we get the distribution shown in the left panel of Figure 14. The mode of this distribution (at 3 components) is close to the true number of experimental phases, which is 4. The slight negative bias can be explained by the fact that we have imposed a restrictive prior in the form of a Poisson distribution on the number of components, as described in the main paper. Finally, Figure 12 shows another complementary representation to Figure 11. The panels show co-allocation matrices that indicate the probability with which two time points are assigned to the same component.The grey shading indicates probabilities, with white corresponding to a probability of 1, and black corresponding to a probability of 0. The structures found in Figure 12 are consistent with those of Figures 11 and 13, as

one can easily convince oneself. When averaging over all node-specific co-allocation matrices, we get the co-allocation matrix shown in the right panel of Figure 14. Note that the true change-points related to the four experimental phases are clearly discernible.



(a) **N=2 network domain**

Figure 1: **AUC scores for the network with $N = 2$ shown in Figure 1a of the main paper.** Node X is autocorrelated with autocorrelation $\sqrt{1-\varepsilon^2}$ and the regulatory mechanisms $X \rightarrow Y$ is implemented as a piecewise linear process. The figure is arranged as a 4-by-5 matrix with cells corresponding to $h = 20$ different parameter combinations. In the matrix each row corresponds to an ε parameter and each column corresponds to an SNR parameter. For each parameter setting the average AUC scores have been derived from 25 independent data instantiations. The error bars correspond to one standard deviation. In each of the $h = 20$ histograms the 1st bar corresponds to the BDe model, the second bar to the BGe model, the 3rd bar to the GM_{BIC} model of Ko et al., the 4th bar to the BGM model of Grzegorczyk et al., and the 5th bar to the proposed $cpBGe$ model.



(a) **N=4 network domain**

Figure 2: **AUC score histograms for the network with $N = 4$ nodes shown in Figure 1b of the main paper.** Node X is autocorrelated with autocorrelation $\sqrt{1-\varepsilon^2}$ and the three other regulatory mechanisms are realized by piecewise linear processes. The figure is arranged as a 4-by-5 matrix with cells corresponding to $h = 20$ different parameter combinations. In the matrix each row corresponds to an ε parameter and each column corresponds to an SNR value. For each parameter setting the average AUC scores have been derived from 25 independent data instantiations. The error bars correspond to one standard deviation. In each of the $h = 20$ histograms the 1st bar corresponds to the BDe model, the second bar to the BGe model, the 3rd bar to the GM_{BIC} model of Ko et al., the 4th bar to the BGM model of Grzegorczyk et al., and the 5th bar to the proposed $cpBGe$ model.

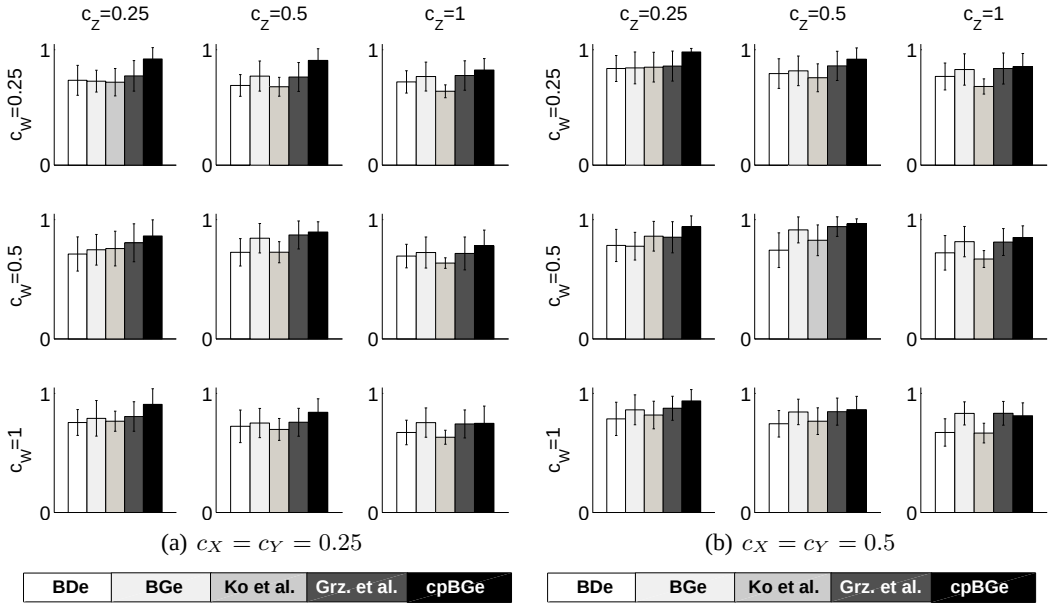


Figure 3: **AUC score histograms for the network with $N = 4$ nodes shown in Figure 1c of the main paper.** Node Z is co-regulated by three nodes X, Y, and W. While the effects of $X(t)$ and $Y(t)$ on $Z(t+1)$ are linear, node W is autocorrelated and a sinusoid signal $c_W \cdot sin(W(t)$ is given to $Z(t+1)$. There are two panels for different c_X, c_Y coefficients in the figure: (a) $c_X = c_Y = 0.25$ and (b) $c_X = c_Y = 0.5$. Both panels are arranged as 3-by-3 matrices with cells corresponding to $h = 9$ different c_W, c_Z parameter combinations. In the matrices each row corresponds to a c_W and each column corresponds to a c_Z coefficient. For each parameter setting the average AUC scores have been derived from 25 independent data instantiations. The error bars correspond to one standard deviation. In each of the $h = 18$ histograms the 1st bar corresponds to the BDe model, the second bar to the BGe model, the 3rd bar to the GM_{BIC} model of Ko et al., the 4th bar to the BGM model of Grzegorczyk et al., and the 5th bar to the proposed $cpBGe$ model.

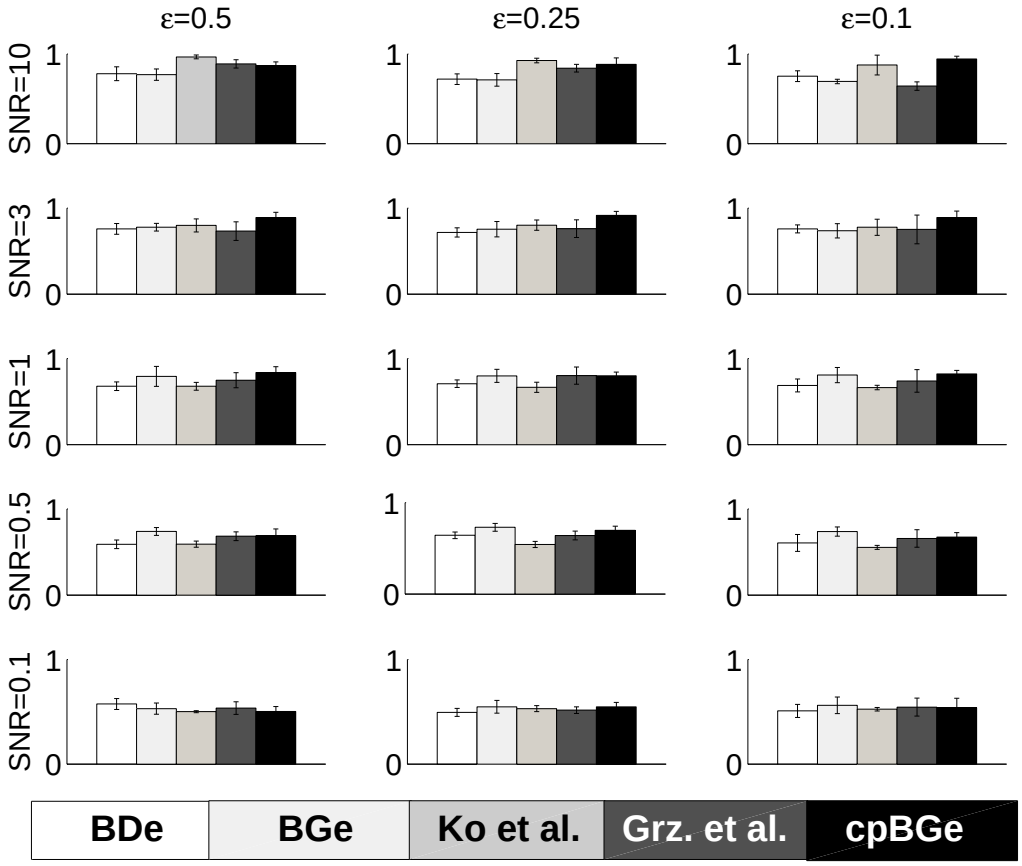
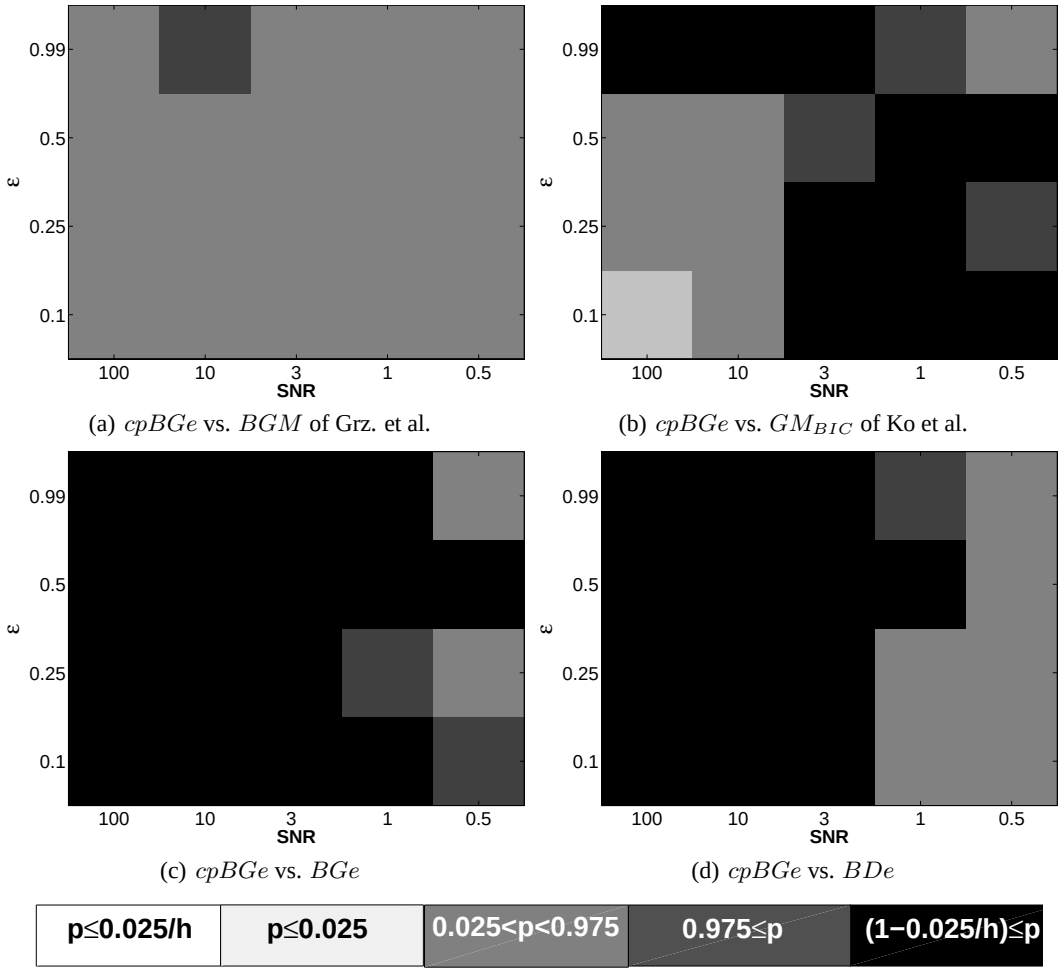


Figure 4: **AUC score histograms for the RAF-pathway with $N = 11$ nodes shown in Figure 1d of the main paper.** Node $PIP2$ is autocorrelated with autocorrelation $\sqrt{1-\varepsilon^2}$ and all other node interactions are implemented via piecewise linear processes. The figure is arranged as a 3-by-5 matrix with cells corresponding to $h = 15$ different ε (rows) and SNR (columns) combinations. For each parameter setting the average AUC scores have been derived from 5 independent data instantiations. The error bars correspond to one standard deviation. In each of the $h = 15$ histograms the 1st bar corresponds to the BDe model, the second bar to the BGe model, the 3rd bar to the GM_{BIC} model of Ko et al., the 4th bar to the BGM model of Grzegorczyk et al., and the 5th bar to the proposed $cpBGe$ model.



(a) $cpBGe$ vs. BGM of Grz. et al.

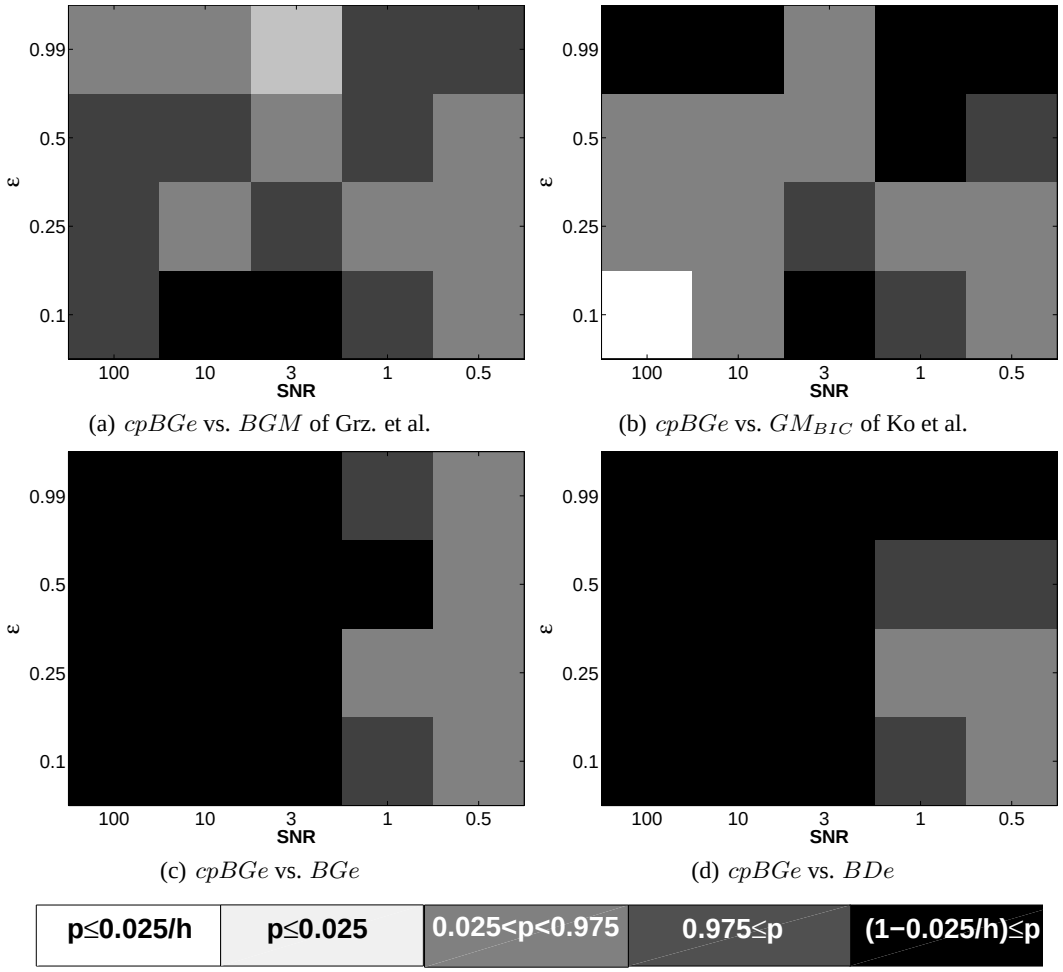
(b) $cpBGe$ vs. GM_{BIC} of Ko et al.

(c) $cpBGe$ vs. BGe

(d) $cpBGe$ vs. BDe

(e) legend

Figure 5: **Heatmatrices of significant average AUC score differences for the network with $N = 2$ nodes shown in Figure 1a of the main paper.** For each combination of SNR and ε a (two-sided) t-test for paired samples was employed to test whether $cpBGe$ performed significantly better or worse than each of the four competing methods. The p-values are visualized by heatmatrices, whereby cells are black if $cpBGe$ performed significantly better after family-wise Bonferroni correction and dark grey if $cpBGe$ performed significantly better only at the (uncorrected) level $\alpha = 0.025$, that is, without correction for multiple testing. The cells are white if $cpBGe$ performed significantly worse after family-wise Bonferroni correction and light grey if $cpBGe$ performed significantly worse only at the (uncorrected) level $\alpha = 0.025$, that is without correction for multiple testing. If there was no significant difference at all, the corresponding cells are drawn in a medium grey. In each panel a-d $h = 20$ hypothesis were tested and each single p-value was computed from 25 data instantiations.



(a) $cpBGe$ vs. BGM of Grz. et al.

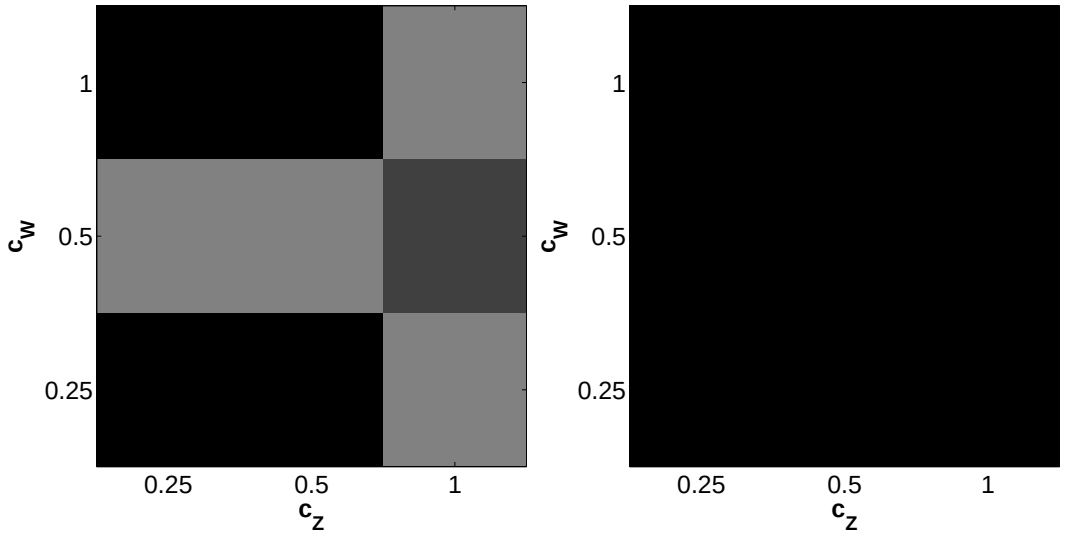
(b) $cpBGe$ vs. GM_{BIC} of Ko et al.

(c) $cpBGe$ vs. BGe

(d) $cpBGe$ vs. BDe

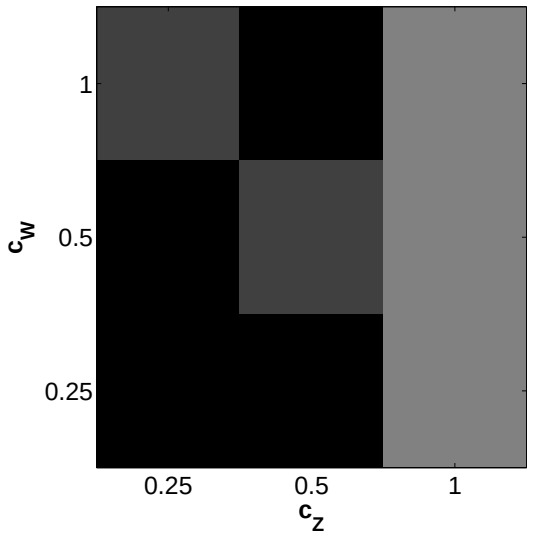
(e) legend

Figure 6: **Heatmatrices of significant differences of AUC scores for the network with $N = 4$ nodes shown in Figure 1b of the main paper.** For each combination of SNR and ε a (two-sided) t-test for paired samples was employed to test whether $cpBGe$ performed significantly better or worse than each competing method. The p-values are visualized by heatmatrices, whereby cells are black if $cpBGe$ performed significantly better after family-wise Bonferroni correction and dark grey if $cpBGe$ performed significantly better only at the (uncorrected) level $\alpha = 0.025$, that is, without correction for multiple testing. The cells are white if $cpBGe$ performed significantly worse after family-wise Bonferroni correction and light grey if $cpBGe$ performed significantly worse only at the level $\alpha = 0.025$, that is without correction for multiple testing. If there was no significant difference at all, the corresponding cells are drawn in a medium grey. In each panel a-d $h = 20$ hypothesis were tested and each single p-value was computed from 25 independent data instantiations.

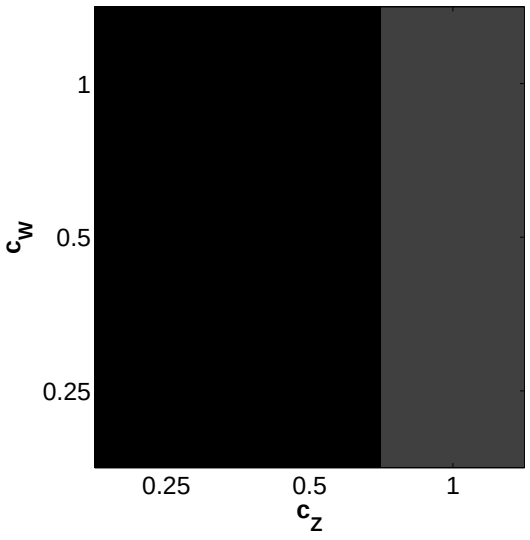


(a) $cpBGe$ vs. BGM of Grz. et al.

(b) $cpBGe$ vs. GM_{BIC} of Ko et al.



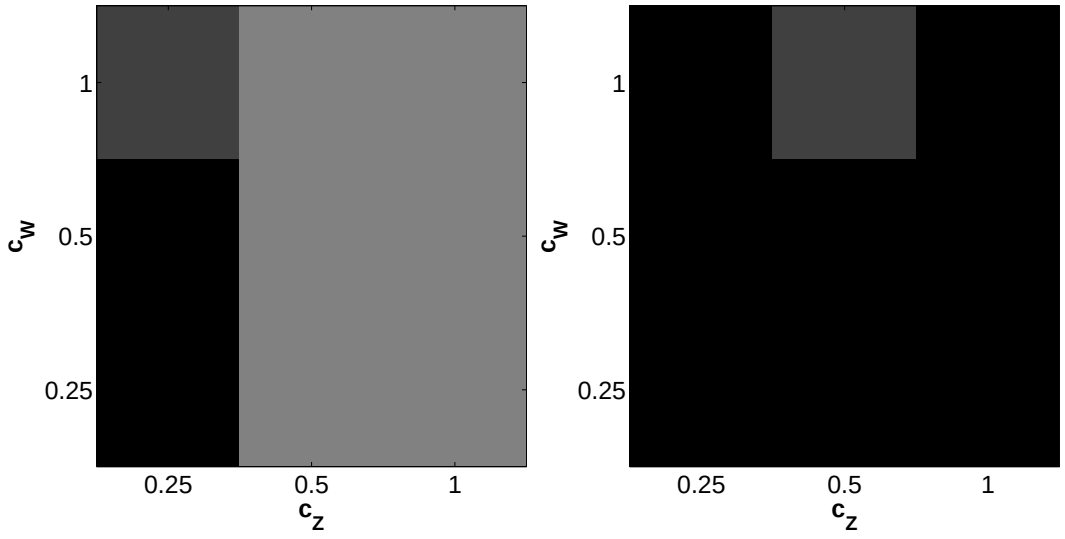
(c) $cpBGe$ vs. BGe



(d) $cpBGe$ vs. BDe

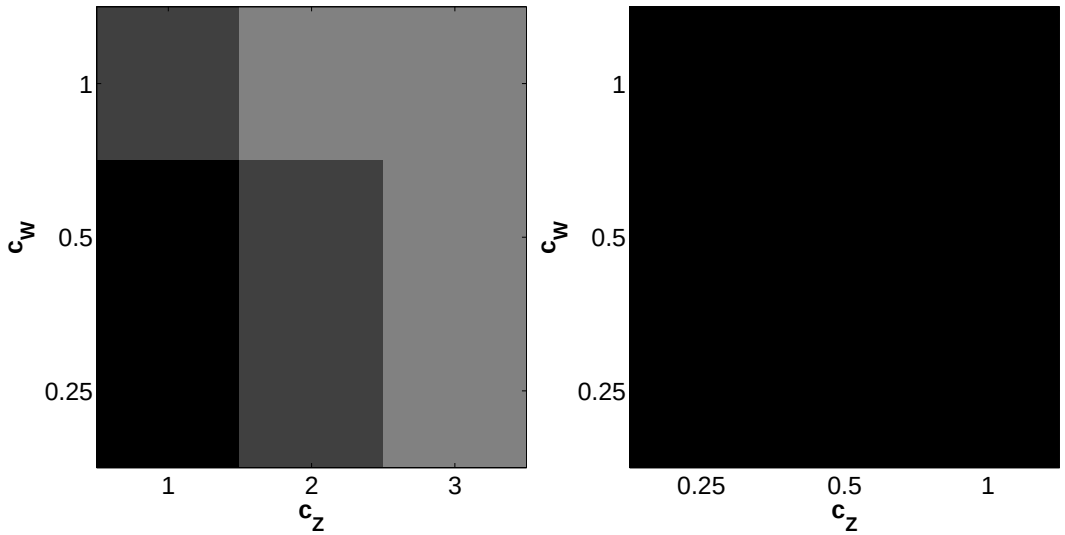
(e) legend

Figure 7: **Heatmatrices of significant differences of AUC scores for the network with $N = 4$ nodes shown in Figure 1c of the main paper with $c_X = c_Y = 0.25$.** For each parameter combination of c_W and c_Y a (two-sided) t-test for paired samples was employed to test whether $cpBGe$ performed significantly better or worse than each competing method. The p-values are visualized by heatmatrices, whereby cells are black if $cpBGe$ performed significantly better after family-wise Bonferroni correction and dark grey if $cpBGe$ performed significantly better only at the (uncorrected) level $\alpha = 0.025$, that is, without correction for multiple testing. The cells are white if $cpBGe$ performed significantly worse after family-wise Bonferroni correction and light grey if $cpBGe$ performed significantly worse only at the (uncorrected) level $\alpha = 0.025$, that is without correction for multiple testing. If there was no significant difference at all, the corresponding cells are drawn in a medium grey. In each panel a-d $h_{0.25} = 9$ hypothesis were tested and each single p-value was computed from 25 independent data instantiations. To take into consideration that further $h_{0.5} = 9$ hypothesis had to be tested for $c_X = c_Y = 0.5$ (see Figure 8) a Bonferroni correction for $h = h_{0.25} + h_{0.5} = 18$ hypothesis in total was used in each panel a-d.



(a) $cpBGe$ vs. BGM of Grz. et al.

(b) $cpBGe$ vs. GM_{BIC} of Ko et al.

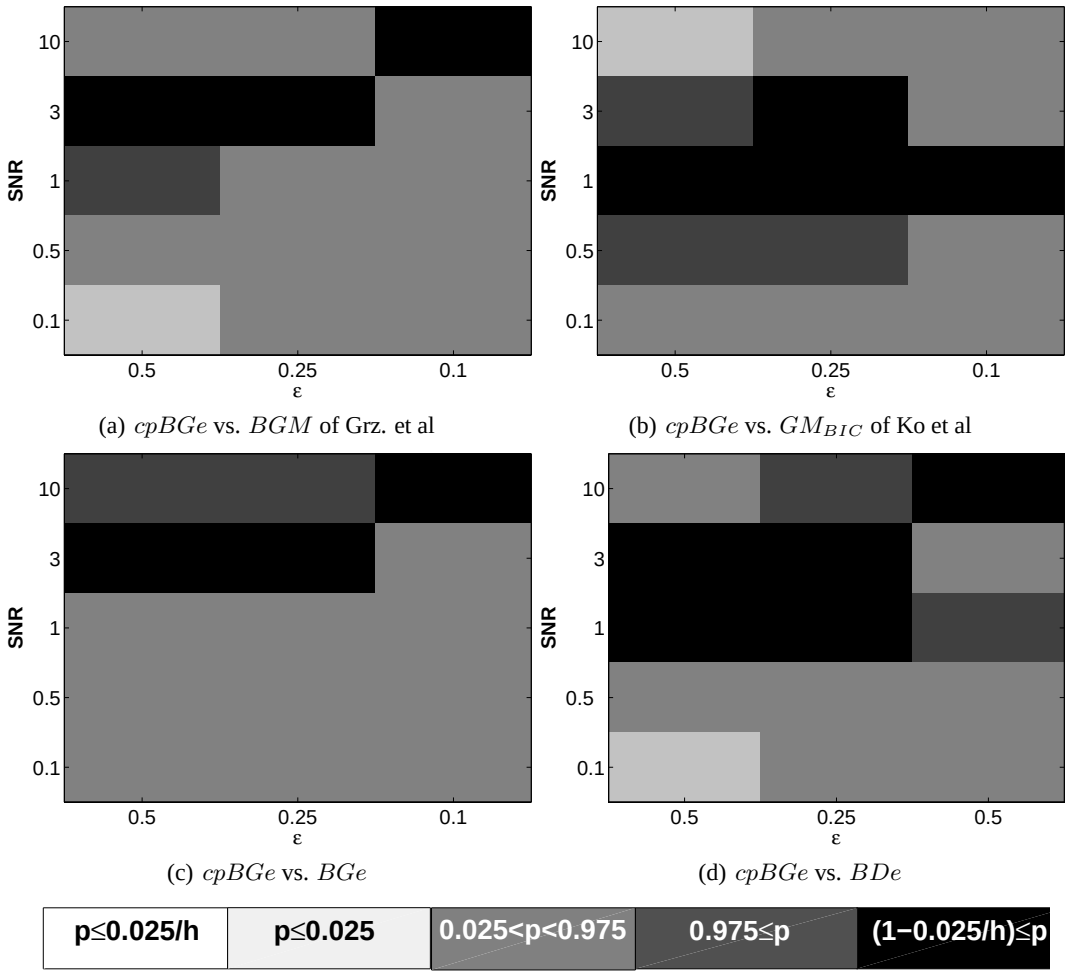


(c) $cpBGe$ vs. BGe

(d) $cpBGe$ vs. BDe

(e) legend

Figure 8: **Heatmatrices of significant differences of AUC scores for the network with $N = 4$ nodes shown in Figure 1c of the main paper with $c_X = c_Y = 0.5$.** For each parameter combination of c_W and c_Y a (two-sided) t-test for paired samples was employed to test whether $cpBGe$ performed significantly better or worse than each competing method. The p-values are visualized by heatmatrices, whereby cells are black if $cpBGe$ performed significantly better after family-wise Bonferroni correction and dark grey if BGM_2 performed significantly better only at the (uncorrected) level $\alpha = 0.025$, that is, without correction for multiple testing. The cells are white if $cpBGe$ performed significantly worse after family-wise Bonferroni correction and light grey if $cpBGe$ performed significantly worse only at the (uncorrected) level $\alpha = 0.025$, that is without correction for multiple testing. If there was no significant difference at all, the corresponding cells are drawn in a medium grey. In each panel a-d $h_{0.5} = 9$ hypothesis were tested and each single p-value was computed from 25 independent data instantiations. To take into consideration that further $h_{0.25} = 9$ hypothesis had to be tested for $c_X = c_Y = 0.5$ (see Figure 7) a Bonferroni correction for $h = h_{0.25} + h_{0.5} = 18$ hypothesis in total was used in each panel a-d.



(a) $cpBGe$ vs. BGM of Grz. et al

(b) $cpBGe$ vs. GM_{BIC} of Ko et al

(c) $cpBGe$ vs. BGe

(d) $cpBGe$ vs. BDe

Figure 9: **Heatmatrices of significant differences of AUC scores for RAF pathway shown in Figure 1d of the main paper.** For each combination of SNR and ε a (two-sided) t-test for paired samples was employed to test whether $cpBGe$ performed significantly better or worse than each competing method. The p-values are visualized by heatmatrices, whereby cells are black if $cpBGe$ performed significantly better after family-wise Bonferroni correction and dark grey if $cpBGe$ performed significantly better only at the (uncorrected) level $\alpha = 0.025$, that is, without correction for multiple testing. The cells are white if $cpBGe$ performed significantly worse after family-wise Bonferroni correction and light grey if $cpBGe$ performed significantly worse only at the (uncorrected) level $\alpha = 0.025$, that is without correction for multiple testing. If there was no significant difference at all, the corresponding cells are drawn in a medium grey. In each panel a-d $h = 15$ hypothesis were tested and each single p-value was computed from 5 independent data instantiations.

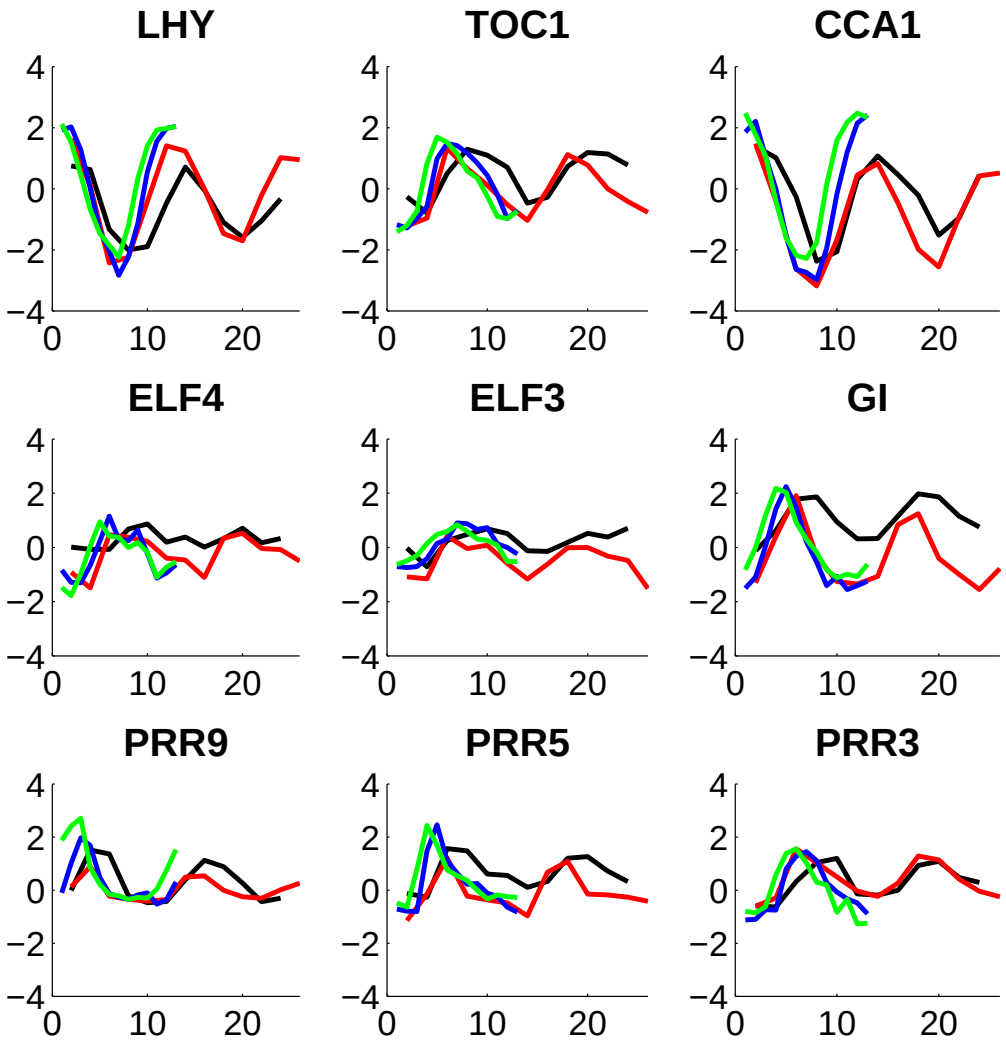


Figure 10: **Overlaid time series plots of the four *Arabidopsis thaliana* gene expression time series listed in Table 1.** Gene expression levels (y-axis) have been plotted against time (x-axis), measured in hours. The curves have been drawn in different colors for the four time series: Black curve: 1st column in Table 1 (time series from Mockler et al.). Red curve: 2nd column in Table 1 (time series from Edwards et al.). Green and blue curve: 3rd and 4th column in Table 1 (two time series from Grzegorczyk et al.: green: T_{20} (column 3 in Table 1) and blue: T_{28} (column 4 in Table 1)).

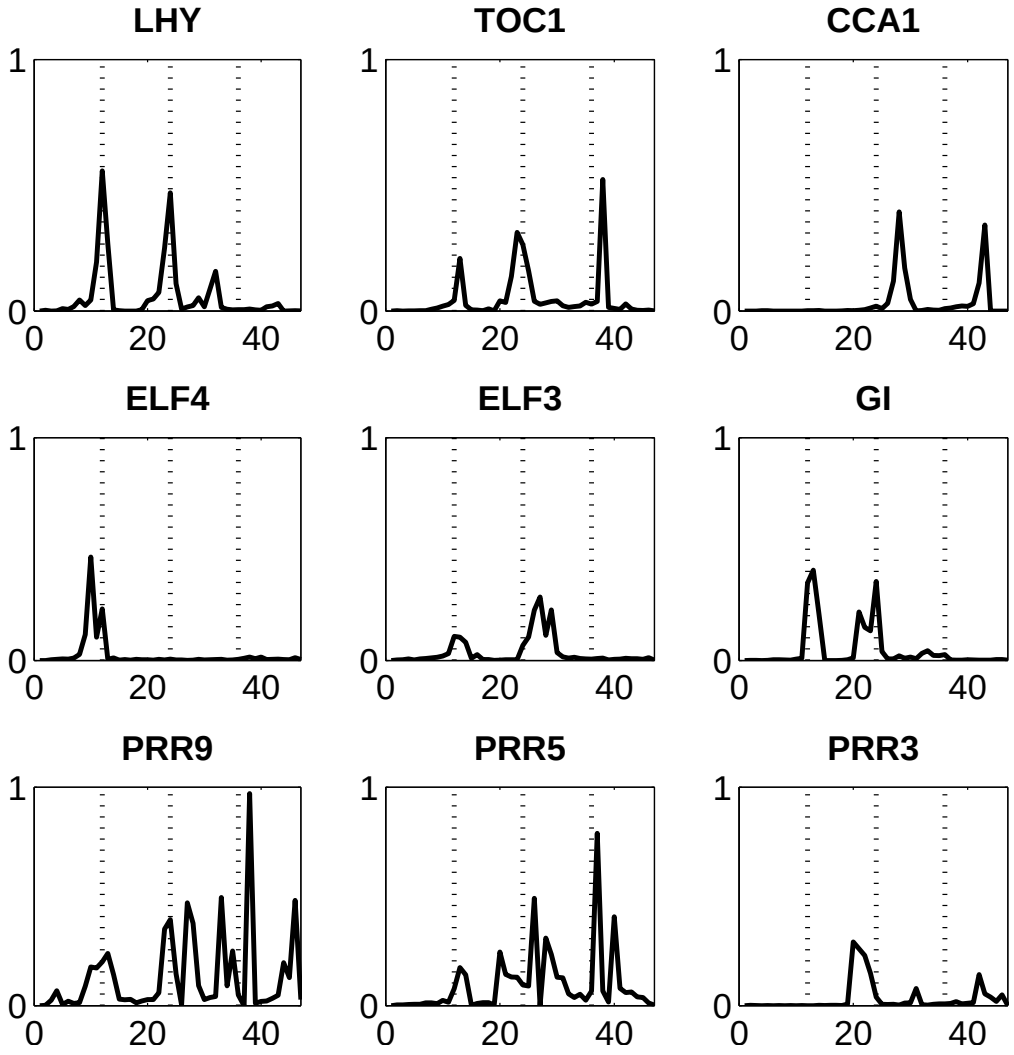


Figure 11: **cpBGe inference on *Arabidopsis thaliana* time series data: Node specific posterior probabilities of transition times for all nine circadian genes.** The vertical dotted lines indicate where a new time series segment starts, that is they mark the first time point of the new segment in Table 1. Panels (a) and (b) for genes LHY and TOC1 are also shown in the main paper.

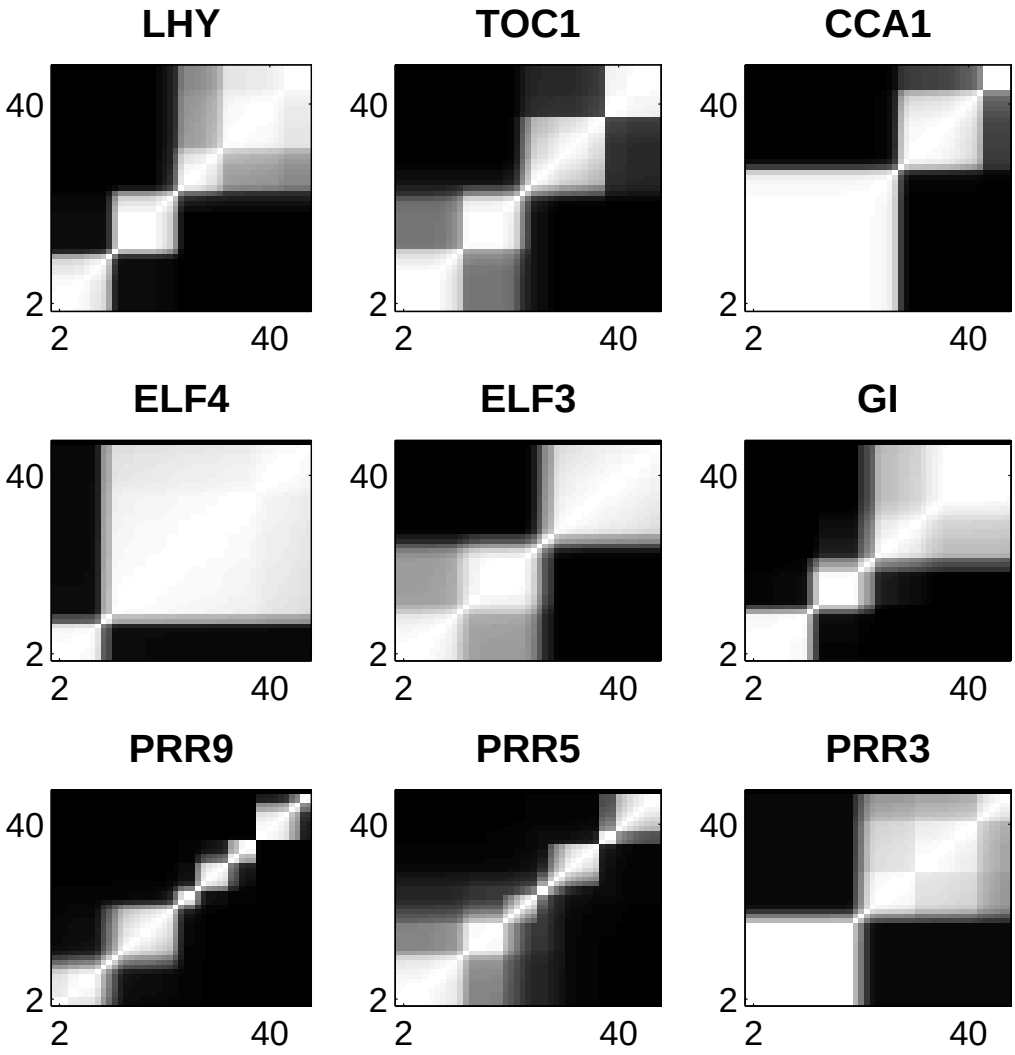


Figure 12: **cpBGe inference on *Arabidopsis thaliana* time series data: Node specific connectivity structure for all nine circadian genes.** For each circadian gene a co-allocation matrix is shown. The grey shading indicates the posterior probability of two time points being assigned to the same mixture component, ranging from 0 (black) to 1 (white). Panels (a) and (b) for genes LHY and TOC1 are also shown in the main paper.

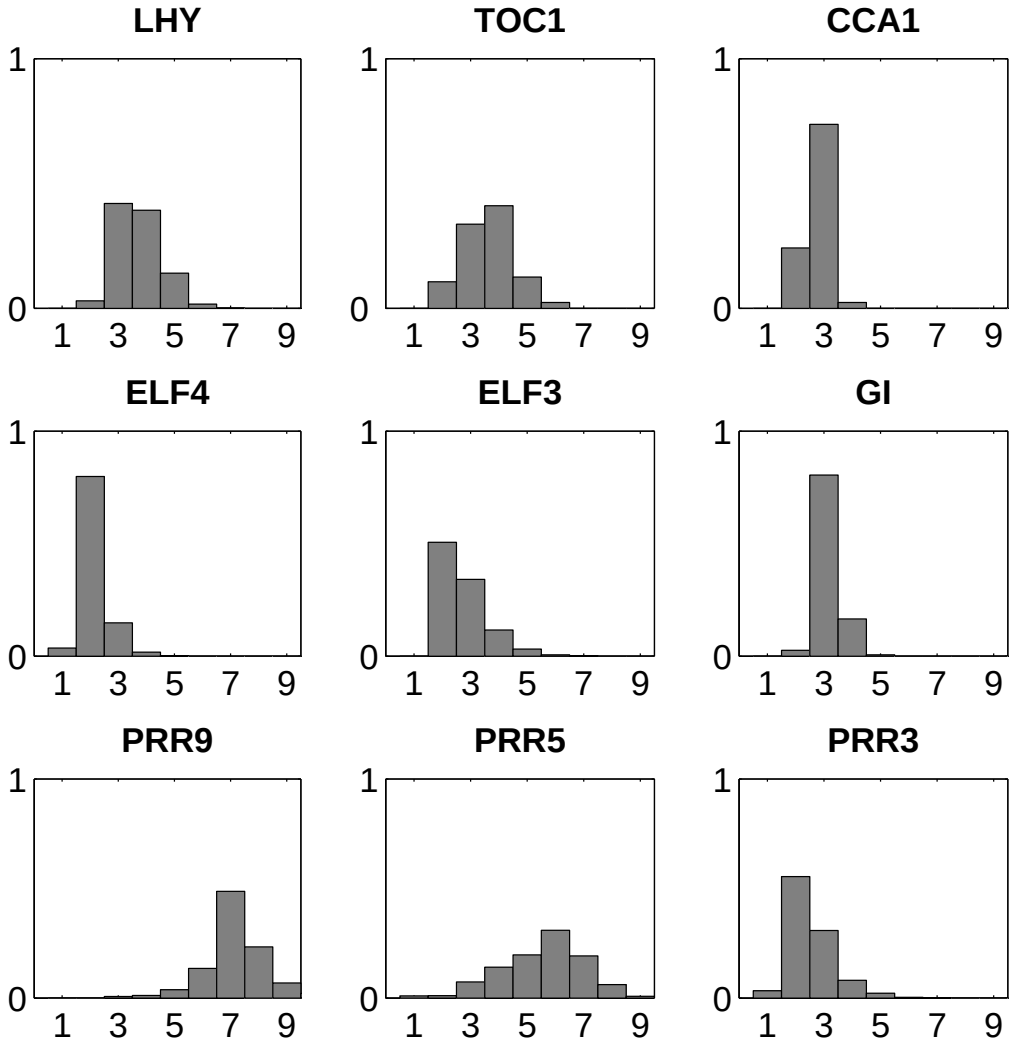


Figure 13: **cpBGe inference on *Arabidopsis thaliana* time series data: Node specific posterior probabilities of the number of mixture components.** For each of the nine circadian genes the posterior distribution of the number of mixture components (= number of transitions plus one) is represented as a histogram.

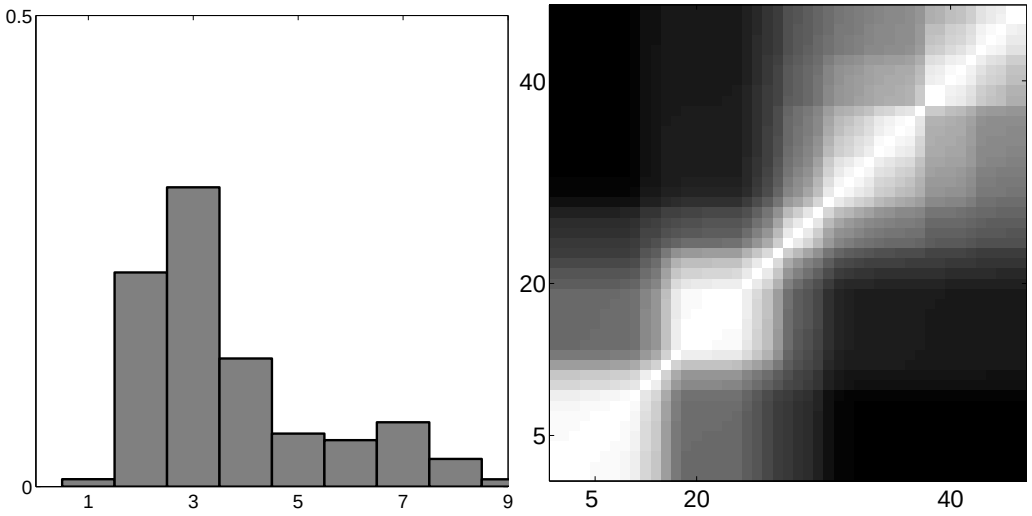


Figure 14: **cpBGe inference on *Arabidopsis thaliana* time series data: Average over all 9 genes.** (a) The average posterior distribution of the number of mixture components (= number of transitions plus one) over all nine genes is represented as a histogram. (b) Average co-allocation matrix over all nine circadian genes. 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).

References

[1] D. Geiger and D. Heckerman. Learning Gaussian networks. *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 235–243, 1994.

[2] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, Vol. 7, No.4:457–472, 1992.

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

[4] P. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

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

[6] A. Hartemink. *Principled Computational Methods for the Validation and Discovery of Genetic Regulatory Networks*. PhD thesis, Massachusetts Institute of Technology, 2001.

[7] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:245–274, 1995.

[8] Y. Ko, C. Zhai, and S. Rodriguez-Zas. Inference of gene pathways using Gaussian mixture models. In *BIBM International Conference on Bioinformatics and Biomedicine*, pages 362–367. Fremont, CA, 2007.

[9] D. Madigan and J. York. Bayesian graphical models for discrete data. *International Statistical Review*, 63:215–232, 1995.

[10] J. Robinson and A. Hartemink. Non-stationary dynamic bayesian networks. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1369–1376. 2009.

[11] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.