Aus der Fakultät Statistik
der technischen Universität Dortmund

Bayesian networks for reconstructing gene regulatory networks in systems biology research

Habilitationsschrift
zur Erlangung der Lehrbefugnis
(venia legendi)
für das Fach Statistik

vorgelegt der Fakultät Statistik der
technischen Universität Dortmund
am 26.10.2011

von
Diplom-Statistiker
Dr. Marco Grzegorczyk

Acknowledgments

Most importantly, I acknowledge the strong scientific support of three people: Prof. (emeritus) Dr. Wolfgang Urfer (Dortmund University), Dr. Dirk Husmeier (BioSS Edinburgh, UK) and Prof. Dr. Jörg Rahnenführer (TU Dortmund University).
At the end of 2002, while I was writing my Diploma thesis, Wolfgang Urfer inspired me to focus my research on Bayesian networks and gave me the possibility to do a PhD on this topic. Wolfgang Urfer supervised my PhD thesis from 2003 to 2006, and most importantly he acquainted me with Dr. Dirk Husmeier during my PhD fellowship.
Since then I could greatly benefit from Dirk Husmeier's scientific experience and advices. With Dirk Husmeier I had various fruitful collaborations and interesting scientific discussions during my post doctorial fellowship at the Center for Systems Biology at Edinburgh (CSBE) from July 2007 to December 2008. Fortunately, this important collaboration for me, could be maintained since January 2009 via various "Skype meetings", and "I'll keep my fingers crossed" that this collaboration will continue for many years.
At the end of 2009 I returned to TU Dortmund University, as I could get a post doctorial fellowship for 2 years in the "Research Training Group (Graduiertenkolleg) Statistical Modelling" at the Department of Statistics, TU Dortmund University. I am deeply grateful for the financial funding from January 2009 to December 2010, which was provided by the German Research Foundation (DFG), and to Jörg Rahnenführer, who "recruited" me for this position and enabled me to continue my research on Bayesian networks in Dortmund. Apart from various scientific inputs and lots of interesting discussions, Jörg Rahnenführer especially encouraged me giving my first lectures on Bayesian networks at TU Dortmund University and he also elated me to write this thesis. I also learned from Jörg Rahnenführer how to write applications for funding so that I could successfully apply for a personal research grant at the German Research Foundation (DFG). However, at the end of 2010 my fellowship in the Research Training Group ended, the DFG had not arrived at a decision about my research project application yet, and my only option was to wait until April 2011 for a guest lectureship for "Applied Statistics" at the Department of Mathematics of Oldenburg University, for which I had successfully applied. I strongly acknowledge that Jörg Rahnenführer provided interim funding for me for the three months until I could start the lectureship in April 2011.
I am also deeply grateful that the German Research Foundation (DFG) is providing funding for a 3 years fellowship at TU Dortmund University for my research project "Development of new Bayesian network models for systems biology research" (GR38531/1-1). This DFG research grant ensures that I can carry on with my Bayesian network research, presented in this thesis, and I am looking forward to starting the research project soon.
Finally and on a personal note, I would like to thank all members of my family for their support. Most notably, I thank my parents Claudia and Udo Grzegorczyk, since they helped me solving all those annoying *non-scientific* "everyday problems" that arose throughout the years. For example, I am deeply grateful for all those lifts to and from Dortmund airport, while I was living in Edinburgh. Another example, I would also like to mention, is that my parents borrowed me their only car for almost two months in 2009, when I was waiting for my new car after my old one was broken.

Contents

1 Computational systems biology

The ultimate objective of systems biology research is the elucidation of the regulatory networks and signalling pathways of the cell. The ideal approach would be the deduction of a detailed mathematical description of the entire system in terms of a set of coupled nonlinear differential equations from dynamic (time series) data. To this end, various mechanistic models have been proposed in the literature (e.g. see Cao and Ren (2008), Xiao and Cao (2008), Wang *et al.* (2010), and Wilkinson (2006) among others). These mechanistic models provide a powerful approach to the modeling of small systems composed of a few components. However, since high-throughput measurements at the cell level are inherently stochastic and most kinetic rate constants cannot be measured directly, the parameters of the system would have to be estimated from the data. Unfortunately, multiple parameter sets of nonlinear systems of differential equations can offer equally plausible solutions, and standard optimization techniques in high-dimensional multimodal parameter spaces are not robust and do not provide a reliable indication of the confidence intervals. Most importantly, model selection would be impeded by the fact that more complex pathway models would always provide a better explanation of the data than less complex ones, rendering this approach intrinsically susceptible to over-fitting.

Therefore, especially for static (steady state) data, correlation and mutual information based relevance network approaches have been proposed by Butte and Kohane (2000). As relevance networks neither distinguish between direct and indirect interactions (e.g. correlations and pseudo-correlations) nor extract directed connections (causal relationships), they cannot be seen as a sufficient remedy either. Therefore, in the context of gene regulatory networks more sophisticated models, such as Gaussian graphical models (Schäfer and Strimmer (2005a) and Schäfer and Strimmer (2005b)), which are based on partial correlations rather than correlations, have been proposed. Gaussian graphical models distinguish between direct and indirect interactions, but they cannot extract causal relationships either. That is, in the first instance these models learn which network components are directly associated with each other, but there is no distinction between mere correlations and causations. However, having learned a network with undirected edge connections, various techniques, e.g. those methods proposed in Spirtes *et al.* (2001) or more recently in Basso *et al.* (2005) and Margolin *et al.* (2006), can be applied to find some causal relationships (directed edge connections) among the undirected edge connections of the network.

In the seminal paper by Friedman *et al.* (2000) machine learning methods based on Bayesian network models have been proposed for the elucidation of gene regulatory network structures. Bayesian networks can be applied to both static and dynamic data so that they compete against mechanistic models for dynamic data and against other graphical models, such as Gaussian graphical models and relevance networks, for static data. Static Bayesian networks are based on directed (acyclic) graphs that encode conditional (in-)dependencies, and partially directed graphs can be learned from data (Spirtes *et al.* (2001) and Pearl (2000)). Partially directed graphs contain both undirected edges, indicating correlated nodes, and directed edges, indicating causal relationships. As discussed in Section 2.1 it depends on the underlying network topology how many directed edges can be learned.[1]

In the last decade, in particular, novel dynamic Bayesian network models have been developed by various authors, and nowadays dynamic Bayesian networks can be seen as a promising trade-off between over-simplicity and loss of computational tractabil-

[1] An empirical cross-method comparison of the network reconstruction accuracies of relevance networks, Gaussian graphical models and Bayesian networks for static (steady state) data can for example be found in Werhli *et al.* (2006).

ity (Cantone *et al.*, 2009). In a nutshell, the idea is to simplify the mathematical description of the biological system by replacing the coupled differential equations of mechanistic models by simple conditional probability distributions of a standard form such that the unknown parameters can be integrated out analytically. This results in a scoring function (the "marginal likelihood") of closed-form that depends only on the structure of the regulatory network and avoids the over-fitting problem referred to above in the context of mechanistic models. Novel fast Markov Chain Monte Carlo (MCMC) algorithms, as those proposed in Friedman and Koller (2003) and Grzegorczyk and Husmeier (2008), can be applied to systematically search the configuration space of network structures for those networks that are most consistent with the data. To obtain the closed-form expression of the marginal likelihood, two probabilistic models with their respective conjugate prior distributions have been employed in the past: the multinomial distribution with the Dirichlet prior, leading to the so-called BDe score (Cooper and Herskovits, 1992), and the linear Gaussian distribution with the normal-Wishart prior, leading to the BGe score (Geiger and Heckerman, 1994). These approaches are restricted in that they either require the data to be discretized (BDe) or can only capture linear regulatory relationships (BGe). A non-linear non-discretized model based on heteroscedastic regression has been proposed by Imoto *et al.* (2003). However, this approach no longer allows the marginal likelihood to be obtained in closed-form and requires a restrictive approximation (the Laplace approximation) to be adopted. Another nonlinear model based on node-specific Gaussian mixture models has been proposed in Ko *et al.* (2007). Again, the marginal likelihood is intractable. The authors resort to the Bayesian information criterion (BIC) of Schwarz (1978) for model selection, which is only a good approximation to the marginal likelihood in the limit of very large data sets.

The standard assumption underlying dynamic Bayesian networks (DBNs) is that of homogeneity: temporal processes and the time-series they generate are assumed to be governed by a homogeneous Markov relation. However, regulatory interactions and signal transduction processes in the cell are usually adaptive and change in response to external stimuli. Following earlier approaches aiming to relax the homogeneity assumption for undirected graphical models (Talih and Hengartner (2005) and Xuan and Murphy (2007)), various recent research efforts have therefore addressed the homogeneity assumption for dynamic Bayesian networks. An approach that has become popular recently is based on a combination of a dynamic Bayesian network with a multiple changepoint process, and the application of a Bayesian inference scheme via Reversible Jump Markov Chain Monte Carlo (RJMCMC). Robinson and Hartemink (2009) proposed a discrete non-homogeneous dynamic Bayesian network model, which allows for different network structures in different segments of the time series, with a regularization term penalizing differences among the network structures. Grzegorczyk and Husmeier (2009c) proposed a continuous non-homogeneous dynamic Bayesian network model, in which the parameters are allowed to vary, while a common network structure provides information sharing among the time series segments. Lèbre (2007), Lèbre *et al.* (2010), and Husmeier *et al.* (2010) proposed alternative continuous non-homogeneous dynamic Bayesian network models, which are based on the Bayesian linear regression model of Andrieu and Doucet (1999). The latter Bayesian network models are more flexible in that they allow the network structure to vary among the time segments. The Bayesian network model proposed in Kolar *et al.* (2009) is a close cousin of a non-homogeneous dynamic Bayesian network. But as opposed to the first four approaches, (hyper-)parameters are not consistently inferred within the Bayesian context.

The above mentioned non-homogeneous dynamic Bayesian networks can be divided into two classes according to whether changepoints are common to the whole network (*class 1*), or varying from node to node (*class 2*). The approach of *class 1*, e.g. pursued in Grzegorczyk *et al.* (2008), Robinson and Hartemink (2009), and Grze-

gorczyk *et al.* (2011), is over-restrictive, as it does not allow for individual nodes to be affected by changing processes in different ways.[2] The approach of *class 2*, e.g. pursued in Grzegorczyk and Husmeier (2009c), Lèbre (2007), Lèbre *et al.* (2010), and Husmeier *et al.* (2010) is potentially over-flexible, as it does not provide any information sharing among the nodes. When an organism undergoes transitional changes, e.g. morphogenic transitions during embryogenesis, from larva to pupa or from pupa to adult fly in Drosophila, one would expect the majority of genes to be affected by these transitions in identical ways. However, there is no mechanism in the fully flexible model that incorporates this prior notion of commonality. A novel model that regularizes between a *class 1* and *class 2* models has been developed recently (Grzegorczyk and Husmeier, 2011a) and will be presented in Subsection 3.3.6 of this thesis.

The non-homogeneous Bayesian network models that will be presented in Subsection 3.3 of this thesis infer network structures that are kept fixed among time segments. In principle, as mentioned above, allowing the network structure to change between segments leads to a more flexible model. However, the latter approach faces a conceptual and a practical problem. The *practical* problem is potential model over-flexibility.[3] Owing to the high costs of postgenomic high-throughput experiments, time series in systems biology are typically rather short. Modeling short time series segments with separate network structures will almost inevitably lead to inflated inference uncertainty, which calls for some information sharing between the segments. The *conceptual* problem is related to the very premise of a flexible network structure. This assumption is reasonable for some applications, like morphogenesis, where different time segments may be associated with different morphogenetic stages (e.g. morphogenesis in *Drosophila melanogaster* (fruit fly), as presented in Robinson and Hartemink (2009)). However, for most cellular processes on a shorter time scale, it is questionable whether it is the structure rather than just the strength of the regulatory interactions that changes with time. To use the analogy of the traffic flow network invoked in Robinson and Hartemink (2009): it is not the road system (the network structure) that changes between off-peak and rush hours, 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).

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

[3] Note that as opposed to Lèbre (2007), the network models from Robinson and Hartemink (2009) and Husmeier *et al.* (2010) partially addresses this issue via a prior distribution that discourages changes in the network structure.

2 Bayesian networks

2.1 Introduction

This section gives an introduction to static and dynamic Bayesian network methodology and introduces some graph theoretic notations that will be required throughout this thesis. The key idea behind Bayesian networks is the representation of conditional (in-)dependency assumptions among variables using directed graphs. A directed graph consists of nodes, each representing a random variable, and directed edges representing dependencies among the variables (nodes). For a set with five random variables, $\{A, \ldots, E\}$, an example of a directed (acyclic) graph, which has five directed edges, $A \to B$, $A \to C$, $B \to D$, $C \to D$, and $D \to E$, is shown in the left panel of Figure 1.
More generally, for random variables $X_1, \ldots, X_N$ the directed graph, $\mathcal{G}$, implies a set of conditional (in-)dependency relations. Conditional on the graph the distributional form of the joint probability distribution, $P(X_1, \ldots, X_N|\mathcal{G})$, has to be chosen such that these stochastical (in-)dependencies, encoded in the graph topology, $\mathcal{G}$, are conserved in the probabilistic model. Loosely speaking, the probabilistic model has to be chosen such that an observed sample of realizations of the variables, $X_1, \ldots, X_N$, can be "explained" best by graphs that encode the true (in-)dependencies among the variables.
More formally, following the *Bayesian* paradigm the goal is either to find the graph with the highest posterior probability, e.g. by a greedy-search algorithm, or to sample graphs from the posterior distribution by Markov Chain Monte Carlo (MCMC) algorithms.[4] Given the data, $\mathcal{D}$, the posterior distribution of a graph, $\mathcal{G}$, is defined as follows:

$$P(\mathcal{G}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{G})P(\mathcal{G})}{P(\mathcal{D})} \tag{1}$$

where $P(\mathcal{D}|\mathcal{G})$ is the marginal likelihood and $P(\mathcal{G})$ is the prior probability of the graph, $\mathcal{G}$. The probability $P(\mathcal{D})$ in Eq. (1) serves as a normalization constant[5] and is defined as follows:

$$P(\mathcal{D}) = \sum_{\mathcal{G}^\star} P(\mathcal{D}|\mathcal{G}^\star)P(\mathcal{G}^\star) \tag{2}$$

where the sum is over all valid directed graphs.[6] The marginal likelihood, $P(\mathcal{D}|\mathcal{G})$, quantifies how likely the observed data are conditional on the graph, $\mathcal{G}$. Assuming that the true (in-)dependencies among the variables are actually inferable from the data, $\mathcal{D}$, high marginal likelihoods can only be reached by those graphs that imply or approximate these true relationships. The graph prior distribution, $P(\mathcal{G})$, is used to assign a "weight" to each graph. These weights do not depend on the data, $\mathcal{D}$, and can be used to include *external* knowledge, which may be available from previous studies or other external sources.[7] The greater the product of the marginal likelihood, $P(\mathcal{D}|\mathcal{G})$, and the graph prior probability, $P(\mathcal{G})$, the more plausible (likely) is the graph, $\mathcal{G}$, from a *Bayesian* perspective.

[4] Loosely speaking, the posterior probability, $P(\mathcal{G}|\mathcal{D})$, quantifies how much a graph, $\mathcal{G}$, is "supported" by the observed data, $\mathcal{D}$; see Chapter 2 in Husmeier *et al.* (2005).

[5] The expression $P(\mathcal{D})$ does not depend on the graph $\mathcal{G}$.

[6] This definition of $P(\mathcal{D})$ ensures that Eq. (1) is a probability distribution over graphs; in particular Eq. (2) ensures the normalization: $\sum_{\mathcal{G}} P(\mathcal{G}|\mathcal{D}) = 1$.

[7] E.g. for applications in systems biology, graphs that are assumed to be more likely a priori, i.e. before taking the data, $\mathcal{D}$, into account, may receive a higher prior probability ("weight") than other graphs, which are assumed to be very unlikely from a biological perspective.

If the data set consists of independent realizations of the variables, static Bayesian network methodology has to be applied, as explained in more detail in Subsection 2.2, and all valid graphs have to be directed and acyclic.[8] If the variables have been measured over time, dynamic Bayesian network (DBN) methodology is required, as discussed in more detail in Subsection 2.4, and all directed graphs are valid.[9]
In systems biology, where the focus is on gene regulatory networks and protein pathways, the available data sets are often sparse. That is, there are many genes or proteins (variables) that potentially interact with each other, but there are only few observations, i.e. only few experimental measurements are made. The large number of model parameters and the relative sparsity of observational data renders inference computationally demanding. In particular, for sparse data the posterior distribution in Eq. (1) tends to be flat and one single graph is not representative for the conditional (in-)dependencies among the variables. There may be various graphs with high posterior probabilities and these graphs possibly possess substantially different sets of edges. The need to capture this inference uncertainty calls for expensive model averaging approaches, e.g. bootstrapping or Markov Chain Monte Carlo (MCMC) simulations.[10] MCMC simulations are indispensable when adopting a proper Bayesian approach to inference, which tends to be computationally less expensive than the frequentist approach of bootstrapping (Larget and Simon, 1999). From a practical point of view, marginal posterior probabilities of certain (edge-)features (e.g. special edge constellations) are of interest:

$$P(F|\mathcal{D}) = \sum_{\mathcal{G}} P(\mathcal{G}|\mathcal{D}) I_F(\mathcal{G}) \tag{3}$$

where F is a (edge-)feature, and $I_F(\mathcal{G})$ is an indicator variable, which is one if the graph, $\mathcal{G}$, possesses the feature, F, and $I_F(\mathcal{G}) = 0$ otherwise. Since the number of valid graphs grows super-exponentially in the number of variables N (Chickering, 1996), the full model averaging approach in Eq. (3) is computationally not accessible for network domains with more than a dozen variables (genes). Thus, a sample from the posterior distribution in Eq. (1) has to be taken, e.g. via Markov Chain Monte Carlo (MCMC) simulations, as explained in more detail in Subsection 2.6, and the idea is to average across this sample of graphs. Given that a graph sample of length T has been taken from the posterior distribution in Eq. (1), one "theoretically" expects all graph with high posterior probabilities to be represented according to their posterior probabilities.[11] The frequency of a feature, F, in the graph sample, $\mathcal{G}_1, ..., \mathcal{G}_T$, is then an adequate estimator, $\widehat{P(F|\mathcal{D})}$, for its true marginal posterior probability, $P(F|\mathcal{D})$, defined in Eq. (3):

$$\widehat{P(F|\mathcal{D})} = \frac{1}{T} \sum_{i=1}^{T} I_F(\mathcal{G}_i) \tag{4}$$

Recalling that $P(\mathcal{D})$ is a normalization constant, it can be seen from Eq. (1) that the posterior probability of a graph, $P(\mathcal{G}|\mathcal{D})$, is proportional to the marginal likelihood, $P(\mathcal{D}|\mathcal{G})$, times the graph prior distribution, $P(\mathcal{G})$, symbolically:

$$P(\mathcal{G}|\mathcal{D}) \propto P(\mathcal{D}|\mathcal{G}) P(\mathcal{G}) \tag{5}$$

The graph prior distribution, $P(\mathcal{G})$, can be used to incorporate biological prior knowledge, or in the absence of genuine prior knowledge about the regulatory network, $\mathcal{G}$,

[8] A directed graph is acyclic if there is no loop of directed edges; see Subsection 2.2 for details.
[9] In DBNs all directed graphs are valid independently of whether they are acyclic or not.
[10] Rather than searching for one single graph with the highest posterior probability, e.g. by greedy-search algorithms.
[11] In practical applications, the MCMC simulation may have failed to converge so that this assumption may be violated. To avoid misleading results it is important to check for proper convergence and mixing; convergence diagnostics are discussed in Subsection 2.7.

a uniform distribution may be assumed for $P(\mathcal{G})$.[12] Alternatively, $P(\mathcal{G})$ may also be chosen such that graphs are penalized according to their "complexities", e.g. quantified as the number of edges. However, it is the marginal likelihood, $P(\mathcal{D}|\mathcal{G})$, where the probabilistic Bayesian network model comes in: A probabilistic model has to be chosen which is in consistency with the conditional (in-)dependencies encoded in the graph, $\mathcal{G}$. The probabilistic model determines the distributional form, $\mathcal{F}$, of the variables and the corresponding parameter vector, $\boldsymbol{\theta} := \boldsymbol{\theta}|\mathcal{G}$, such that the conditional (in-)dependencies implied by the graph, $\mathcal{G}$, are conserved. Given a suitable probabilistic model, the likelihood of the data, $P(\mathcal{D}|\mathcal{G}, \boldsymbol{\theta})$, can be computed for each instantiation of the parameter vector, $\boldsymbol{\theta}$. The marginal likelihood, $P(\mathcal{G}|\mathcal{D})$, in Eq. (5) is the integral of the likelihood over the whole parameter space:

$$P(\mathcal{D}|\mathcal{G}) = \int P(\mathcal{D}|\mathcal{G}, \boldsymbol{\theta}) P(\boldsymbol{\theta}|\mathcal{G}) d\boldsymbol{\theta} \tag{6}$$

where $P(\boldsymbol{\theta}|\mathcal{G})$ is the prior distribution of the unknown parameter vector, $\boldsymbol{\theta}$. Employing the marginal likelihood, $P(\mathcal{D}|\mathcal{G})$, rather than $P(\mathcal{D}|\mathcal{G}, \widehat{\boldsymbol{\theta}}_{ML})$, i.e. the likelihood with the maximum likelihood estimator $\widehat{\boldsymbol{\theta}}_{ML}$ for the unknown parameters, is beneficial, since the marginal likelihood includes an inherent penalty for unnecessary complexity and so efficiently guards against over-fitting (Bishop, 2006). The marginal likelihood, $P(\mathcal{D}|\mathcal{G})$, will also be called the "score" of the graph, $\mathcal{G}$, throughout this thesis. In the two most popular Bayesian network models the prior distribution, $P(\boldsymbol{\theta}|\mathcal{G})$, of the unknown parameters, $\boldsymbol{\theta}$, is assumed to be the conjugate distribution of the likelihood distribution, $P(\mathcal{D}|\mathcal{G}, \boldsymbol{\theta})$, so as to obtain a closed form solution of the integral in Eq. (6). These aspects will be introduced more formally in Subsection 2.2 for static Bayesian networks and in Subsection 2.4 for dynamic Bayesian networks.

2.2 Static Bayesian network models

Static Bayesian networks (BNs) are interpretable and flexible models for representing probabilistic relationships among interacting variables. The graph, $\mathcal{G}$, of a BN consists of a set of N nodes (variables), $X_1, \ldots, X_N$, and a set of directed edges between these nodes. If there is a directed edge pointing from node X_i to node X_j, symbolically $X_i \to X_j$ or formally $\mathcal{G}(i, j) = 1$, then X_i is called a parent (node) of X_j, and X_j is called a child (node) of X_i. The parent set of node X_n, symbolically π_n, is defined as the set of all parent nodes of X_n, $\pi_n = \{X_i | i = 1, \ldots, N : \mathcal{G}(i, n) = 1\}$. There is a one-to-one mapping between the graph, $\mathcal{G}$, and the N parent sets π_n; i.e. $\mathcal{G}(i, n) = 1$ if $X_i \in \pi_n$; and vice-versa $\mathcal{G}(i, n) = 0$ if $X_i \notin \pi_n$. Hence, it can be written symbolically: $\mathcal{G} = \{\pi_1, \ldots, \pi_N\}$. If a node X_k can be reached by following a path of directed edges $X_i \to \ldots \to X_k$ starting at node X_i, then X_k is called a descendant of X_i, and X_i is called an ancestor of X_k. The graph structure of a static Bayesian network is defined to be a directed acyclic graph (DAG), that is, a directed graph in which no node can be its own descendant.[13] As an example, consider the directed acyclic graph (DAG) shown in the left panel of Figure 1. There are five variables, namely $A, \ldots, E$, and node A is a parent node of both nodes B and C, i.e. B and C are child nodes of node A. The parent set, π_A, of node A is the empty set, as there is no directed edge converging on node A. The parent set of node B is $\pi_B = \{A\}$,

[12] Werhli and Husmeier (2007) show how to extract and systematically incorporate biological prior knowledge about edge connections from available network data bases.

[13] Graphically this means that a graph is acyclic if there are no cycles of directed edges (loops), such as $X_i \to \ldots \to X_i$.

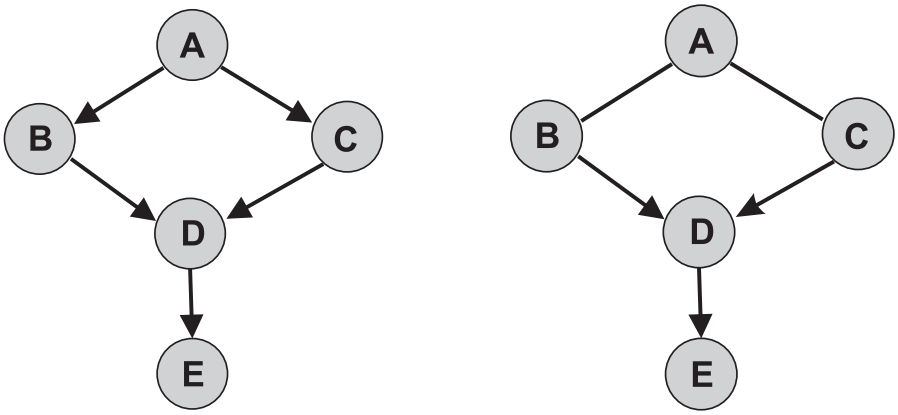


Figure 1: **A directed acyclic graph (DAG) and its CPDAG representation.** The left panel shows a directed acyclic graph (DAG) for the variables $A, \ldots, E$. The right panel shows the completed partially directed acyclic graph (CPDAG) representation of the DAG.

and the parent set of node D is $\pi_D = \{B, C\}$. Since there is a path of directed edges leading from node A to node D, e.g. $A \to B \to C$ or $A \to C \to D$, node A is an ancestor of node D, and vice-versa, node D is a descendant of node A.
In Bayesian network models the directed edges of the DAG are assumed to imply several conditional (in-)dependence relations: Most importantly that (i) conditional on its parent set, π_n, each node, X_n, becomes stochastically independent of its other ancestors, (ii) nodes without common ancestors are marginally stochastically independent, and (iii) nodes with common descendants become stochastically dependent when conditional on one or more of their mutual descendants.[14]
As an example consider two small networks with three nodes A, B, and C: (i) In the first network $\mathcal{G}_1$: $A \to B \to C$ the nodes A and C are marginally dependent, $P(A, B|\mathcal{G}_1) \neq P(A|\mathcal{G}_1)P(C|\mathcal{G}_1)$, but conditional on its parent node B, node C becomes stochastically independent of node A, symbolically $P(C|A, B, \mathcal{G}_1) = P(C|B, \mathcal{G}_1)$. (ii) In the second network $\mathcal{G}_2$: $A \to B \leftarrow C$ the nodes A and C are marginally independent: $P(A, C|\mathcal{G}_2) = P(A|\mathcal{G}_2)P(C|\mathcal{G}_2)$, but (iii) in the same network $\mathcal{G}_2$ the nodes A and C have the common descendant B and therefore become dependent when conditional on B, $P(A, C|B, \mathcal{G}_2) \neq P(A|B, \mathcal{G}_2)P(C|B, \mathcal{G}_2)$.
Given a probabilistic model that ensures that the above mentioned assumptions (i)-(iii) hold, it can be derived straightforwardly that conditional on a directed acyclic graph, $\mathcal{G}$, and a fixed parameter set, the joint probability distribution can be factorized:

$$P(X_1, \ldots, X_N|\mathcal{G}, \boldsymbol{\theta}) = \prod_{n=1}^{N} P(X_n|\pi_n, \boldsymbol{\theta}_n) \tag{7}$$

where $\boldsymbol{\theta}$ is the total parameter vector, composed of node-specific subvectors, $\boldsymbol{\theta}_n$. Thus, DAGs imply sets of conditional (in-)dependence assumptions for static Bayesian networks (BNs), and so factorizations of the joint probability distribution in which each node X_n depends on its parent set, π_n, only. The graph, $\mathcal{G}$, in the left panel of Figure 1, for example, implies the following factorization of the joint probability distribution:

$$P(A, \ldots, E|\mathcal{G}, \boldsymbol{\theta}) = P(A|\boldsymbol{\theta}_A)P(B|\{A\}, \boldsymbol{\theta}_B)P(C|\{A\}, \boldsymbol{\theta}_C)P(D|\{B, C\}, \boldsymbol{\theta}_D)P(E|\{D\}, \boldsymbol{\theta}_E) \tag{8}$$

[14]The first assumption (i) is known as the *Markov assumption*.

In the context of *static* Bayesian networks, more than one DAG can imply exactly the same set of conditional (in-)dependencies, and if two DAGs assert the same set of conditional (in-)dependence assumptions, those DAGs are said to be equivalent. This relation of graph equivalence imposes a set of equivalence classes over DAGs. The DAGs within an equivalence class have the same underlying undirected graph, but may disagree on the direction of some of the edges. Verma and Pearl (1990) prove that two DAGs are equivalent if and only if they have the same skeleton and the same set of v-structures. The skeleton of a DAG is defined as the undirected graph which results from ignoring all edge directions. And a v-structure denotes a configuration, $X_i \rightarrow X_n \leftarrow X_k$, of two directed edges converging on the same node, X_n, without an edge between X_i and X_k (Chickering, 1995). The DAG in the left panel of Figure 1 possesses one v-structure, namely: $B \rightarrow D \leftarrow C$. Chickering (1995) shows that equivalence classes of DAGs can be uniquely represented using completed partially directed acyclic graphs (CPDAGs). A CPDAG has the same skeleton as the original DAG, but possesses both directed (compelled) and undirected (reversible) edges. Every compelled (directed) edge, $X_i \rightarrow X_j$, in a CPDAG denotes that all DAGs of this equivalence class contain this directed edge, while every reversible (undirected) edge, $X_i - X_j$, in the CPDAG representation denotes that some DAGs in the equivalence class contain the directed edge, $X_i \rightarrow X_j$, while other DAGs in the equivalence class contain the oppositely orientated edge, $X_i \leftarrow X_j$. A directed edge of the DAG is compelled in the CPDAG if its reversal changes the set of v-structures, otherwise it may be either compelled or reversible. In the DAG in the left panel of Figure 1 the edges $B \rightarrow D$ and $C \rightarrow D$ are compelled, as reversing one of these two edges would delete the v-structure $B \rightarrow D \leftarrow C$. The edge $D \rightarrow E$ is also compelled, as its reversal would create two new v-structures $B \rightarrow D \leftarrow E$ and $C \rightarrow D \leftarrow E$. The remaining two edges $A \rightarrow B$ and $A \rightarrow C$ are both reversible. An algorithm that takes as input a DAG and outputs the corresponding CPDAG representation can be found in Chickering (2002). The CPDAG representation of the DAG in the left panel of Figure 1 is shown in the right panel.

Because of these equivalence classes of DAGs (represented by CPDAGs) it is important to keep in mind that *not* all directed edges in a static Bayesian network can be interpreted causally, e.g. the edges $A \rightarrow B$ and $A \rightarrow C$ of the DAG shown in Figure 1 cannot be interpreted causally, since they are reversible (undirected) in the CPDAG representation. Like a Bayesian network, a causal network is mathematically represented by a DAG. However, the edges in a causal network have a stricter interpretation: the parents of a variable are its immediate causes. In the presentation of a causal network it is meaningful to make the causal Markov assumption (Pearl, 2000): Given the values of a variable's immediate causes, it is independent of its earlier causes. Under this assumption, a causal network can be interpreted as a static Bayesian network in that it satisfies the corresponding Markov independencies. However, the reverse does *not* hold. The probability models for static Bayesian networks that are usually considered give the same scores (marginal likelihood values) for equivalent DAGs, so that only equivalence classes (i.e. CPDAGs) can be learned from static data.

The domain variables, $X_1, \ldots, X_N$, can be composed to a random vector, $\vec{X} = (X_1, \ldots, X_N)^T$. Given a N-by-m data set matrix, $\mathcal{D}$, that consists of m independent realizations of this random vector, $\vec{X}$, let $\mathcal{D}_{n,i}$ and $\mathcal{D}_{\pi_n,i}$ denote the realizations of X_n and π_n in data point i. It then follows from Eq. (7) that the likelihood of static Bayesian networks (BNs) is given by:

$$P(\mathcal{D}|\mathcal{G},\boldsymbol{\theta}) = \prod_{n=1}^{N}\prod_{i=1}^{m} P\Big(X_n = \mathcal{D}_{n,i}|\pi_n = \mathcal{D}_{\pi_n,i},\boldsymbol{\theta}_n\Big) \tag{9}$$

where the node-specific subvectors, $\boldsymbol{\theta}_n$, specify the local conditional distributions in the factorization. From Eq. (9) and under the assumption of parameter independence:

$$P(\boldsymbol{\theta}|\mathcal{G}) = \prod_{n=1}^{N} P(\boldsymbol{\theta}_n|\pi_n) \tag{10}$$

the marginal likelihood, defined in Eq. (6), is given by

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

$$\Psi(\mathcal{D}_n^{\pi_n}) = \int \prod_{i=1}^{m} P\Big(X_n = \mathcal{D}_{n,i}|\pi_n = \mathcal{D}_{\pi_n,i},\boldsymbol{\theta}_n\Big)P(\boldsymbol{\theta}_n|\pi_n)d\boldsymbol{\theta}_n \tag{12}$$

where $\mathcal{D}_n^{\pi_n} := \{(\mathcal{D}_{n,i},\mathcal{D}_{\pi_n,i}) : 1 \le i \le m\}$ is the subset of data pertaining to node X_n and its parent set, π_n. For each variable, X_n, the term $\Psi(\mathcal{D}_n^{\pi_n})$ is usually called the *local score* of X_n $(n = 1,\ldots,N)$.
In the BGe model (Geiger and Heckerman, 1994) a linear Gaussian distribution is chosen for the local conditional distribution, $P(X_n|\pi_n,\boldsymbol{\theta}_n)$ in Eq. (12), and the conjugate normal-Wishart distribution is assigned to the local prior distributions, $P(\boldsymbol{\theta}_n|\pi_n)$.[15]
The BDe model (Geiger and Heckerman, 1994) can be used for discrete data. The local conditional distributions, $P(X_n|\pi_n,\boldsymbol{\theta}_n)$, in Eq. (12) are then assumed to be a set of multinomial distributions (one for each realization of the parent nodes), and conjugate Dirichlet distributions are assigned to the local prior distributions, $P(\boldsymbol{\theta}_n|\pi_n)$.[16]
Under fairly weak regularity conditions discussed in Geiger and Heckerman (1994) (parameter modularity), the integral in Eq. (12) has a closed form solution for both Bayesian network models (BGe and BDe) and equivalent scores are assigned to equivalent DAGs.

2.3 The Gaussian BGe model 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 (1994). Given a data set, $\mathcal{D}$, with m observations of the nodes (variables), $X_1,\ldots,X_N$:

$$\mathcal{D} = \begin{pmatrix} \mathcal{D}_{1,1} & \mathcal{D}_{1,2} & \ldots & \mathcal{D}_{1,m-1} & \mathcal{D}_{1,m} \\ \mathcal{D}_{2,1} & \mathcal{D}_{2,2} & \ldots & \mathcal{D}_{2,m-1} & \mathcal{D}_{2,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathcal{D}_{N,1} & \mathcal{D}_{N,2} & \ldots & \mathcal{D}_{N,m-1} & \mathcal{D}_{N,m} \end{pmatrix} \tag{13}$$

so that $\mathcal{D}_{n,j}$ denotes the j-th realization of the n-th node, X_n, let $\mathcal{D}_{\cdot,j}$ denote the j-th column of $\mathcal{D}$: $\mathcal{D}_{\cdot,j} = (\mathcal{D}_{1,j},\ldots,\mathcal{D}_{N,j})^T$, i.e. the j-th realization vector of the N nodes. 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, $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$, with an unknown mean vector, $\boldsymbol{\mu}$, and an unknown covariance matrix, $\boldsymbol{\Sigma}$. The prior joint distribution of the mean vector, $\boldsymbol{\mu}$, and the precision matrix, $\mathbf{W} = \boldsymbol{\Sigma}^{-1}$, is supposed to be the normal-Wishart distribution, that is, the conditional distribution of $\boldsymbol{\mu}$ given

[15] Bayesian Gaussian (BG) network model with score equivalence (e).
[16] Bayesian Discrete (BD) network model with score equivalence (e).

$\mathbf{W}$ is the $\mathcal{N}(\boldsymbol{\mu}_0, (v \cdot \mathbf{W})^{-1})$ Gaussian distribution, where $v > 0$, and the marginal distribution of the precision (inverse covariance) matrix, $\mathbf{W}$, is a Wishart distribution with $\alpha > N+1$ degrees of freedom and covariance matrix, $\mathbf{T}_0$, denoted $\mathcal{W}(\alpha, \mathbf{T}_0)$. The condition $\alpha > N + 1$ ensures that the second moments of the posterior distribution are finite; see also Eq. (26) in Geiger and Heckerman (1994). Geiger and Heckerman show that the marginal likelihood, $P(\mathcal{D}|\mathcal{G})$, from Eq. (9) can then – under fairly weak conditions of parameter independence and parameter modularity – be computed in closed form.
For convenience, it can first be defined:

$$\mathbf{T}_{\mathcal{D},m} := \mathbf{T}_0 + \mathbf{S}_{\mathcal{D},m} + \frac{v \cdot m}{v + m}(\boldsymbol{\mu}_0 - \overline{\mathcal{D}_m})(\boldsymbol{\mu}_0 - \overline{\mathcal{D}_m})^T \tag{14}$$

where

$$\overline{\mathcal{D}_m} := \frac{1}{m}\sum_{j=1}^{m} \mathcal{D}_{.,j} \tag{15}$$

is the mean of the m realization vectors and $\mathbf{S}_{\mathcal{D},m}$ is the empirical covariance matrix multiplied by $m - 1$:

$$\mathbf{S}_{\mathcal{D},m} := \sum_{j=1}^{m} (\mathcal{D}_{.,j} - \overline{\mathcal{D}_m}) \cdot (\mathcal{D}_{.,j} - \overline{\mathcal{D}_m})^T \tag{16}$$

$\mathbf{T}_0$, $\boldsymbol{\mu}_0$, α, and v are hyperparameters of the normal-Wishart prior and have to be specified in advance. $\mathbf{T}_0$ is an N-by-N matrix, $\boldsymbol{\mu}_0$ is a N-by-1 column vector, and v and α are 1-dimensional and usually referred to as total prior precision parameters. Furthermore, the following definition can be introduced:

$$c(n, \alpha) := \left\{ 2^{\alpha \cdot n/2} \cdot \pi^{n \cdot (n-1)/4} \cdot \prod_{i=1}^{n} \Gamma\left(\frac{\alpha + 1 - i}{2}\right) \right\}^{-1} \tag{17}$$

The marginal likelihood from Eq. (9) can then be computed as follows (Geiger and Heckerman (1994)):

$$P(\mathcal{D}|\mathcal{G}) = \prod_{n=1}^{N} \Psi(\mathcal{D}_n^{\pi_n}) = \prod_{n=1}^{N} \frac{P(\mathcal{D}^{\{X_n, \pi_n\}}|\mathcal{G}_F(\{X_n, \pi_n\}))}{P(\mathcal{D}^{\{\pi_n\}}|\mathcal{G}_F(\pi_n))} \tag{18}$$

where X_n is the n-th variable, π_n is the parent set of X_n in the graph, $\mathcal{G} = \{\pi_1, \ldots, \pi_n\}$, $\mathcal{D}^{\{X_n, \pi_n\}}$ 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 these full graphs do *not* impose any independencies among the variables in the subsets $\{X_n, \pi_n\}$ and $\{\pi_n\}$, respectively.
The marginal likelihood of the data subset, $\mathcal{D}^{\{S\}} \subset \mathcal{D}$, corresponding to the m realizations of the $N^\dagger$-dimensional subset, $S \subset \{X_1, \ldots, X_N\}$, given a full graph, $\mathcal{G}_F(S)$, for the subset, S, can be computed as follows (Geiger and Heckerman, 1994):

$$\begin{aligned} P(\mathcal{D}^S|\mathcal{G}_F(S)) \quad = \quad & (2\pi)^{-\frac{N^\dagger \cdot m}{2}} \cdot \left\{ \frac{v}{v+m} \right\}^{N^\dagger/2} \cdot \frac{c(N^\dagger, \alpha)}{c(N^\dagger, \alpha + m)} \\ & \cdot det(\mathbf{T}_0^S)^{\frac{\alpha}{2}} \cdot det(\mathbf{T}_{\mathcal{D},m}^S)^{-\frac{\alpha+m}{2}} \end{aligned} \tag{19}$$

where $det(\mathbf{T}_0^S)$ and $det(\mathbf{T}_{\mathcal{D},m}^S)$ denote the determinants of the submatrices, $\mathbf{T}_0^S$ and $\mathbf{T}_{\mathcal{D},m}^S$ consisting only of those $N^\dagger$ rows and columns that correspond to variables in

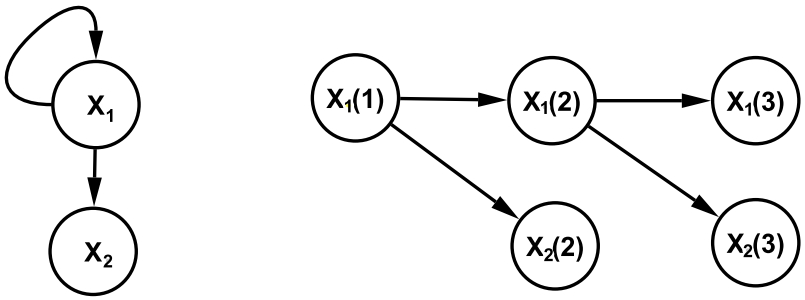


Figure 2: **State space graph and corresponding dynamic Bayesian network of order $\tau = 1$.** The left panel shows a recurrent state space graph containing two nodes. Node X_1 has a recurrent feedback loop and acts as a regulator of node X_2. The right panel shows the same graph unfolded in time. In the right panel $X_1(t)$ and $X_2(t)$ denote the random variables X_1 and X_2 at time point t.

the subset, S. $\mathbf{T}_{\mathcal{D},m}$ was defined in Eq. (14), and $c(N^\dagger, \alpha)$ and $c(N^\dagger, \alpha + m)$ can be computed with Eq. (17).
It can be seen from Eq. (18) that each $\Psi(.)$ term in Eq. (9) can be computed in closed form when the BGe model is used. These $\Psi(.)$ terms are usually referred to as *local (BGe) scores* of the nodes. Throughout this thesis the greek letter Ψ will be used for terms $\Psi(.)$ that can be computed in closed form. In particular, local (BGe) scores for subsets of the data, $\mathcal{D}$, will be required.

2.4 Dynamic Bayesian network models

Dynamic Bayesian networks (DBNs) can be applied if the random vector, $\vec{X} = (X_1, \ldots, X_N)^T$, has been measured over time. Different from static Bayesian networks all interactions between nodes are then subject to a time delay, τ, where τ is called the order of the DBN model. An edge from X_j to X_n, symbolically $X_j \to X_n$ and formally $\mathcal{G}(j, n) = 1$, in a first order DBN, $\tau = 1$, indicates that the realization of X_n at time point t is conditionally dependent on the realization of X_j at time point $t-1$. As for static Bayesian networks π_n denotes the parent set of X_n $(n = 1, \ldots, N)$, and there is a one-to-one mapping between the graph, $\mathcal{G}$, and the system of parent sets, $\{\pi_1, \ldots, \pi_N\}$. Because of the time delay, τ, of interactions, DBNs are based on a bipartite graph structure between two time steps t and $t + 1$ $(t = 2, \ldots, m)$ so that the acyclicity constraint, which is fundamental for the factorization in static Bayesian networks (see Subsection 2.2), is guaranteed to be satisfied. The bipartite graph structure of DBNs is illustrated graphically in Figure 2.
Given a N-by-m data set matrix, $\mathcal{D}$, where $\mathcal{D}_{n,t}$ and $\mathcal{D}_{\pi_n,t}$ are the realizations of X_n and π_n at time point t, DBNs of order $\tau = 1$ are based on the following homogeneous Markov chain expansion:

$$P(\mathcal{D}|\mathcal{G}, \boldsymbol{\theta}) = \prod_{n=1}^{N} \prod_{t=2}^{m} P\Big(X_n = \mathcal{D}_{n,t} | \pi_n = \mathcal{D}_{\pi_n,t-1}, \boldsymbol{\theta}_n\Big) \tag{20}$$

and the marginal likelihood is given by:

$$P(\mathcal{D}|\mathcal{G}) \;=\; \int P(\mathcal{D}|\mathcal{G}, \boldsymbol{\theta}) P(\boldsymbol{\theta}|\mathcal{G}) d\boldsymbol{\theta} \;=\; \prod_{n=1}^{N} \Psi(\mathcal{D}_n^{\pi_n}) \tag{21}$$

$$\Psi(\mathcal{D}_n^{\pi_n}) \quad = \quad \int \prod_{t=2}^{m} P\Big(X_n = \mathcal{D}_{n,t} | \pi_n = \mathcal{D}_{\pi_n,t-1}, \boldsymbol{\theta}_n\Big) P(\boldsymbol{\theta}_n | \pi_n) d\boldsymbol{\theta}_n \qquad (22)$$

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 X_n's parent set, π_n, with a time lag $\tau = 1$. In this thesis only first order DBNs with $\tau = 1$ will be considered, and as for static Bayesian networks $\Psi(\mathcal{D}_n^{\pi_n})$ will be called the *local (DBN) score* of X_n. Although various different probabilistic models have been proposed for modeling the local DBN scores, the Bayesian network models that will be presented in this thesis have been formulated using the dynamic variants of the two standard models for Bayesian networks, BDe and BGe.[17]

2.5 The Gaussian BGe scoring metric for dynamic Bayesian networks

Assume that a time series, $(X_1(t), \ldots X_N(t))_{t=1,\ldots,m}^T$, has been collected for a dynamic Bayesian network (DBN) with N nodes, $X_1, \ldots, X_N$, and a Markovian dependence structure of order $\tau = 1$. For time series data the columns of the data matrix in Eq. (13) do not represent independent (steady-state) observations: the t-th column of $\mathcal{D}$ is the realization of the nodes at time point t ($t = 1, \ldots, m$), and the edges indicate interactions with a time delay, τ. For $\tau = 1$ an edge $X_i \rightarrow X_j$ indicates that the realization, $\mathcal{D}_{j,t}$, of X_j at time point t depends on the realization, $\mathcal{D}_{i,t-1}$, of X_i at the previous time point $t-1$. This can be taken into account in the context of the Gaussian BGe model by building new data matrices – one for each node – from the original data matrix of size N-by-m given in Eq. (13). In principle, there are two alternative variants of the dynamic BGe model that can be used. It depends on whether so called "self-feedback loops" (autoregressive edges), that is edges having the same node as starting and end point, e.g. the edge $X_1 \rightarrow X_1$ in Figure 2, are allowed in the graph or not.

Dynamic Bayesian networks <u>without</u> self-feedback-loops:
When "self-feedback loops" are *not* allowed, the following N matrices $\mathcal{D}(n)$ ($n = 1, \ldots, N$) of size N-by-$(m-1)$ can be built from the original (time series) data matrix

[17] In addition to the two standard Bayesian network models, namely BGe and BDe, various other modeling frameworks have been proposed and applied for the local DBN scores, $\Psi(\mathcal{D}_n^{\pi_n})$; e.g. Bayesian regression models (Rogers and Girolami (2005) and Lèbre *et al.* (2010)). Different from static Bayesian networks, a broader class of probability models can be used for dynamic Bayesian networks (DBNs), since the score equivalence aspect, discussed in Subsection 2.2, is not required. The modeling framework for the local scores of static Bayesian networks is restricted to such models that satisfy a global equivalence condition: $P(\mathcal{D}|\mathcal{G}_1) = P(\mathcal{D}|\mathcal{G}_2)$ for equivalent DAGs G_1 and G_2. Due to the time lag $\tau = 1$ of interactions in DBNs, different graphs always imply different sets of conditional (in-)dependency assumptions and there are no equivalent graphs that have to give identical scores (marginal likelihood values). Without a global equivalence condition the local DBN scores can, in principle, be modeled independently for each node X_n ($n = 1, \ldots, N$).

given in Eq. (13):

$$\mathcal{D}(n) = \begin{pmatrix} \mathcal{D}_{1,1} & \mathcal{D}_{1,2} & \dots & \mathcal{D}_{1,m-1} \\ \mathcal{D}_{2,1} & \mathcal{D}_{2,2} & \dots & \mathcal{D}_{2,m-1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{D}_{n-1,1} & \mathcal{D}_{n-1,2} & \dots & \mathcal{D}_{n-1,m-1} \\ \mathcal{D}_{n,2} & \mathcal{D}_{n,3} & \dots & \mathcal{D}_{n,m} \\ \mathcal{D}_{n+1,1} & \mathcal{D}_{n+1,2} & \dots & \mathcal{D}_{n+1,m-1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{D}_{N,1} & \mathcal{D}_{N,2} & \dots & \mathcal{D}_{N,m-1} \end{pmatrix} \tag{23}$$

Each matrix $\mathcal{D}(n)$ is obtained by deleting the last column of $\mathcal{D}$ and substituting the n-th row $(\mathcal{D}_{n,1}, \dots, \mathcal{D}_{n,m-1})$ for $(\mathcal{D}_{n,2}, \dots, \mathcal{D}_{n,m})$ afterwards. The novel data matrices $\mathcal{D}(n)$ consist of $m-1$ observations for the N nodes, and the hyperparameters T_0 and μ_0 have the same dimensions as in the static BGe model. The closed-form solution of the marginal likelihood for DBNs is very similar to the closed form solution for static Bayesian networks, presented in Subsection 2.3. The matrix $\mathbf{T}_{\mathcal{D}(n)}$ has to be computed for each data set, $\mathcal{D}(n)$, and Eq. (18) is substituted for:

$$P(\mathcal{D}|\mathcal{G}) = \prod_{n=1}^{N} \Psi(\mathcal{D}_n^{\pi_n}) \tag{24}$$

where

$$\Psi(\mathcal{D}_n^{\pi_n}) = \frac{P(\mathcal{D}(n)^{\{X_n,\pi_n\}}|\mathcal{G}_F(\{X_n,\pi_n\}))}{P(\mathcal{D}(n)^{\{\pi_n\}}|\mathcal{G}_F(\pi_n))} \tag{25}$$

Eq. (19) has to be replaced by:

$$\begin{aligned} P(\mathcal{D}(n)^S|\mathcal{G}_F(S)) \quad &= \quad (2\pi)^{-\frac{N^\dagger \cdot (m-1)}{2}} \cdot \left\{ \frac{v}{v+(m-1)} \right\}^{N^\dagger/2} \cdot \frac{c(N^\dagger, \alpha)}{c(N^\dagger, \alpha+(m-1))} \\ &\quad \cdot det(\mathbf{T}_0^S)^{\frac{\alpha}{2}} \cdot det(\mathbf{T}^S_{\mathcal{D}(n),(m-1)})^{-\frac{\alpha+(m-1)}{2}} \end{aligned} \tag{26}$$

where $\mathcal{G}_F(S)$ is a full graph for the subset of nodes, S, of cardinality $N^\dagger$, and $\mathbf{T}_0^S$ and $\mathbf{T}^S_{\mathcal{D}(n),(m-1)}$ are sub-matrices, as explained in Section 2.3.

Dynamic Bayesian networks <u>with</u> self-feedback-loops:
When "self-feedback loops" are allowed, the following N matrices $\mathcal{D}(n)$ $(n = 1, \dots, N)$ of size $(N+1)$-by-$(m-1)$ can be built from the original (time series) data matrix given in Eq. (13):

$$\mathcal{D}(n) = \begin{pmatrix} \mathcal{D}_{1,1} & \mathcal{D}_{1,2} & \dots & \mathcal{D}_{1,m-1} \\ \mathcal{D}_{2,1} & \mathcal{D}_{2,2} & \dots & \mathcal{D}_{2,m-1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{D}_{N,1} & \mathcal{D}_{N,2} & \dots & \mathcal{D}_{N,m-1} \\ \mathcal{D}_{n,2} & \mathcal{D}_{n,3} & \dots & \mathcal{D}_{n,m} \end{pmatrix} \tag{27}$$

$n = 1, \dots, N$. Each matrix $\mathcal{D}(n)$ is obtained by deleting the last column of $\mathcal{D}$ and adding a novel row $(\mathcal{D}_{n,2}, \dots, \mathcal{D}_{n,m})$ of length $m-1$, i.e. the n-th row of $\mathcal{D}$ shifted leftwards by 1, as the $(N+1)$-th row of $\mathcal{D}(n)$. The additional $(N+1)$-th row can be identified with a new node (variable), X_{N+1}. This new variable is the n-th variable with a time shift of size $\tau = 1$ and the time delay, τ, can be taken into account by

substituting X_n for X_N when computing the local score, $\Psi(\mathcal{D}_n^{\pi_n})$, for node X_n; see Eq. (29) below. The data matrices, $\mathcal{D}(n)$, consist of observations for $N+1$ variables so that the hyperparameters T_0 and μ_0 have to be an $(N+1)$-by-$(N+1)$ matrix and an $(N+1)$-by-1 column vector, respectively, here. The matrix $\mathbf{T}_{\mathcal{D}(n)}$ can be computed for each data set, $\mathcal{D}(n)$, and Eq. (18) is substituted for:

$$P(\mathcal{D}|\mathcal{G}) = \prod_{n=1}^{N} \Psi(\mathcal{D}_n^{\pi_n}) \tag{28}$$

where

$$\Psi(\mathcal{D}_n^{\pi_n}) = \frac{P(\mathcal{D}(n)^{\{X_{N+1},\pi_n\}}|\mathcal{G}_F(\{X_{N+1},\pi_n\}))}{P(\mathcal{D}(n)^{\{\pi_n\}}|\mathcal{G}_F(\pi_n))} \tag{29}$$

Eq. (19) has to be replaced by:

$$\begin{aligned} P(\mathcal{D}(n)^S|\mathcal{G}_F(S)) \quad &= \quad (2\pi)^{-\frac{N^\dagger\cdot(m-1)}{2}} \cdot \left\{\frac{v}{v+(m-1)}\right\}^{N^\dagger/2} \cdot \frac{c(N^\dagger,\alpha)}{c(N^\dagger,\alpha+(m-1))} \\ &\quad \cdot det(\mathbf{T}_0^S)^{\frac{\alpha}{2}} \cdot det(\mathbf{T}^S_{\mathcal{D}(n),(m-1)})^{-\frac{\alpha+(m-1)}{2}} \end{aligned} \tag{30}$$

where $\mathcal{G}_F(S)$ is a full graph for the subset of nodes, S, of cardinality $N^\dagger$, and $\mathbf{T}_0^S$ and $\mathbf{T}^S_{\mathcal{D}(n),(m-1)}$ are sub-matrices, as explained in Section 2.3.

2.6 Markov Chain Monte Carlo (MCMC) sampling of graphs

Markov Chain Monte Carlo (MCMC) methods can be used for sampling graphs, $\mathcal{G}$, from the posterior distribution, $P(\mathcal{G}|\mathcal{D})$, given in Eq. (1). In this subsection two different Metropolis-Hastings sampling schemes are presented, namely: the *structure MCMC* sampler of Madigan and York (1995) and the *order MCMC* sampler of Friedman and Koller (2003). Both MCMC sampling schemes can be used to generate a graph sample, $\mathcal{G}_1, ..., \mathcal{G}_T$, from the posterior distribution, $P(\mathcal{G}|\mathcal{D})$. Subsequently, the graph sample can be employed to estimate marginal posterior probabilities of (edge-)features; see Eq. (4) in Subsection 2.1. The structure MCMC algorithm can be used for inference in static and dynamic Bayesian networks. But it is known that structure MCMC tends to converge very slow for static Bayesian networks, especially, when the posterior landscape is peaked (Friedman and Koller, 2003).[18] Therefore, the order MCMC algorithm was developed and proposed by Friedman and Koller (2003), as an alternative to the structure MCMC algorithm for static Bayesian networks.

The structure MCMC sampling scheme:
The structure MCMC sampling scheme of Madigan and York (1995) is a Metropolis-Hastings sampler that generates a sample of graphs, $\mathcal{G}_1, ..., \mathcal{G}_T$, as follows: Given a graph, $\mathcal{G}_i$, a new candidate graph, $\mathcal{G}_{i+1}$, is proposed with probability:

$$Q(\mathcal{G}_{i+1}|\mathcal{G}_i) = \left\{ \begin{array}{ll} \frac{1}{|\mathcal{N}(\mathcal{G}_i)|} & ,\mathcal{G}_{i+1} \in \mathcal{N}(\mathcal{G}_i) \\ 0 & ,\mathcal{G}_{i+1} \notin \mathcal{N}(\mathcal{G}_i) \end{array} \right\} \tag{31}$$

[18] Structure MCMC is based on single edge operations that yield relatively small steps in the configuration space of graphs. Because of the acyclicity constraint, that has to be kept in static Bayesian network, MCMC simulations tend to get trapped in local optima, from which they cannot escape with small steps.

where $\mathcal{N}(\mathcal{G}_i)$ denotes the *neighborhood* of $\mathcal{G}_i$, that is the set of all valid graphs that can be reached from $\mathcal{G}_i$ by the deletion, addition or reversal of one single edge of the current graph, $\mathcal{G}_i$, and $|\mathcal{N}(\mathcal{G}_i)|$ is the cardinality of this set, $\mathcal{N}(\mathcal{G}_i)$. For dynamic Bayesian networks all directed graphs are valid, while the graphs $\mathcal{G}_{i+1}$ have to be acyclic when static Bayesian networks are employed. Thus, in the context of static Bayesian networks it has to be checked, which edges can be added to $\mathcal{G}_i$ and which edges can be reversed in $\mathcal{G}_i$ without violating the acyclicity-constraint; efficient algorithms for these checks can be found in Giudici and Castelo (2003). In the Metropolis-Hastings algorithm the proposed graph $\mathcal{G}_{i+1}$ is accepted with the acceptance probability $A(\mathcal{G}_{i+1}|\mathcal{G}_i) = \min\{1, R(\mathcal{G}_{i+1}|\mathcal{G}_i)\}$ where

$$R(\mathcal{G}_{i+1}|\mathcal{G}_i) = \frac{P(\mathcal{G}_{i+1}|\mathcal{D})}{P(\mathcal{G}_i|\mathcal{D})} \cdot \frac{Q(\mathcal{G}_i|\mathcal{G}_{i+1})}{Q(\mathcal{G}_{i+1}|\mathcal{G}_i)} = \frac{P(\mathcal{D}|\mathcal{G}_{i+1})P(\mathcal{G}_{i+1})}{P(\mathcal{D}|\mathcal{G}_i)P(\mathcal{G}_i)} \cdot \frac{|\mathcal{N}(\mathcal{G}_i)|}{|\mathcal{N}(\mathcal{G}_{i+1})|} \tag{32}$$

while the Markov chain is left unchanged, symbolically $\mathcal{G}_{i+1} := \mathcal{G}_i$, if the new graph is not accepted. $\{\mathcal{G}_i\}_{i=1,2,3,...}$ is then a Markov chain in the configuration space of graphs, whose Markov transition kernel, $K(\tilde{\mathcal{G}}|\mathcal{G})$, for a move from $\mathcal{G}$ to $\tilde{\mathcal{G}}$ is given by the product of the proposal probability and the acceptance probability; for $\tilde{\mathcal{G}} \neq \mathcal{G}$:

$$K(\tilde{\mathcal{G}}|\mathcal{G}) = Q(\tilde{\mathcal{G}}|\mathcal{G}) \cdot A(\tilde{\mathcal{G}}|\mathcal{G}) = \frac{1}{|\mathcal{N}(\mathcal{G})|} \cdot A(\tilde{\mathcal{G}}|\mathcal{G}) \tag{33}$$

and

$$K(\mathcal{G}|\mathcal{G}) = 1 - \sum_{\tilde{\mathcal{G}} \in \mathcal{N}(\mathcal{G})} Q(\tilde{\mathcal{G}}|\mathcal{G}) \cdot A(\tilde{\mathcal{G}}|\mathcal{G}) = 1 - \sum_{\tilde{\mathcal{G}} \in \mathcal{N}(\mathcal{G})} \frac{1}{|\mathcal{N}(\mathcal{G})|} \cdot A(\tilde{\mathcal{G}}|\mathcal{G}) \tag{34}$$

Per construction it is guaranteed that the Markov transition kernel satisfies the equation of detailed balance:

$$\frac{P(\tilde{\mathcal{G}}|\mathcal{D})}{P(\mathcal{G}|\mathcal{D})} = \frac{K(\tilde{\mathcal{G}}|\mathcal{G})}{K(\mathcal{G}|\tilde{\mathcal{G}})} \tag{35}$$

Under ergodicity, that is a sufficient condition for the Markov chain, $\{\mathcal{G}_i\}_{i=1,2,3,...}$, to converge, the posterior distribution, $P(\mathcal{G}|\mathcal{D})$, is the stationary distribution:

$$P(\tilde{\mathcal{G}}|\mathcal{D}) = \sum_{\mathcal{G}} K(\tilde{\mathcal{G}}|\mathcal{G}) P(\mathcal{G}|\mathcal{D}) \tag{36}$$

A reasonable approach adopted in most applications is to impose a limit on the cardinality of the parent sets. This limit is referred to as the fan-in (restriction). The practical advantage of this restriction on the maximum number of edges converging on a node is a reduction of the computational complexity, which improves the convergence (e.g. see Chapter 8 in Husmeier *et al.* (2005)). Fan-in restrictions can be justified in the context of biological expression data, as many experimental results have shown that the expression of a gene is usually controlled by a comparatively small number of active regulator genes, while on the other hand regulator-genes seem to be nearly unrestricted in the number of genes they regulate. The imputation of a fan-in restriction leads to a further reduction of a graph's neighborhood: graphs that contain nodes with too many parent nodes, that is more than the fan-in value, have to be removed from the respective neighborhoods.

The order MCMC sampling scheme:
The order MCMC approach of Friedman and Koller (2003) is a Markov Chain Monte Carlo (MCMC) sampling scheme that generates a sample of node orders, $\prec_1, ..., \prec_T$, from the posterior distribution, $P(\prec|\mathcal{D})$, over node orders, $\prec$, in the context of static Bayesian networks. The state space of the Markov chain is the set of all $N!$ possible

orders of the variables (nodes).
Each node order, $\prec = (X_{\sigma(1)}, ..., X_{\sigma(N)})$, can be seen as implied through a permutation, σ, of the indices $\{1, ..., N\}$, and the meaning of an order, $\prec = (X_{\sigma(1)}, ..., X_{\sigma(N)})$, is that it represents the set of DAGs that are consistent with it in the following sense: A DAG, $\mathcal{G}$, is consistent with the node order, $\prec$, if the parent sets, $\pi_{\sigma(n)}$, are restricted to the nodes $X_{\sigma(1)}, \ldots, X_{\sigma(n-1)}$ that are standing leftwards of $X_{\sigma(n)}$ in the order $\prec$ $(n = 1, \ldots, N)$. That is, if $X_{\sigma(j)}$ precedes $X_{\sigma(i)}$ in $\prec$, then $X_{\sigma(i)}$ is *not* allowed to be a parent node of $X_{\sigma(j)}$. A fan-in restriction can be realized by additionally restricting the cardinalities of the parent sets, $\pi_{\sigma(n)}$ $(n = 1, \ldots, N)$.
For the order MCMC sampling scheme, Friedman and Koller (2003) assume a graph prior distribution, $P(\mathcal{G})$, that can be factorized such that there is one local factor for each node X_n $(n = 1, \ldots, N)$:

$$P(\mathcal{G}) = \prod_{n=1}^{N} P(\pi_n) \tag{37}$$

where $\mathcal{G} = \{\pi_1, \ldots, \pi_N\}$, and they assume a uniform prior over node orders, that is $P(\prec) = \frac{1}{N!}$. Friedman and Koller (2003) introduce a simple *flip-operator* that exchanges one node for another in the current node order to generate a Metropolis Hastings sampler in the space of node orders. This leads to the following proposal probabilities:

$$Q(\prec_{i+1} \mid \prec_i) = \left\{ \begin{array}{cc} \frac{2}{N \cdot (N-1)} & , \prec_{i+1} \in \Pi(\prec_i) \\ 0 & , \prec_{i+1} \notin \Pi(\prec_i) \end{array} \right\} \tag{38}$$

where $\Pi(\prec_i)$ is the set of all node orders that can be reached from $\prec_i$ by exchanging two nodes for each other in $\prec_i$, while leaving the positions of all other nodes in $\prec_i$ unchanged. To guarantee convergence of the Markov chain, $\{\prec_i\}_{i=1,2,3,...}$, to the posterior probability, $P(\prec \mid \mathcal{D})$, the acceptance probabilities in the Metropolis Hastings algorithm are set to $A(\prec_{i+1} \mid \prec_i) = \min\{1, R(\prec_{i+1} \mid \prec_i)\}$, where:

$$R(\prec_{i+1} \mid \prec_i) = \frac{P(\prec_{i+1} \mid \mathcal{D})}{P(\prec_i \mid \mathcal{D})} \times \frac{Q(\prec_i \mid \prec_{i+1})}{Q(\prec_{i+1} \mid \prec_i)} = \frac{P(\mathcal{D} \mid \prec_{i+1})}{P(\mathcal{D} \mid \prec_i)} \times 1 \tag{39}$$

and the likelihood, $P(\mathcal{D} \mid \prec)$, of a node order, $\prec = (X_{\sigma(1)}, ..., X_{\sigma(N)})$, is given by:

$$P(\mathcal{D} \mid \prec) = \prod_{n=1}^{N} \sum_{\pi \in \mathcal{U}_{\sigma(n)}^{\prec}} \Psi(\mathcal{D}_n^{\pi}) \cdot P(\pi) \tag{40}$$

where $\mathcal{U}_{\sigma(n)}^{\prec}$ is the set of all possible parent sets for node $X_{\sigma(n)}$ that contain exclusively nodes standing leftwards to X_n in the node order $\prec = (X_{\sigma(1)}, ..., X_{\sigma(N)})$, and the local scores $\Psi(\mathcal{D}_n^{\pi_n})$ were defined in Eq. (12). As the orders $\prec_i$ and $\prec_{i+1}$ differ by the position of two nodes only, the likelihood ratio in Eq. (39) can be computed efficiently, as explained in Friedman and Koller (2003).
For practical applications of order MCMC it is useful to pre-compute and store the values of the products: $\Psi^{\star}(\mathcal{D}_n^{\pi_n}) := \Psi(\mathcal{D}_n^{\pi_n}) \cdot P(\pi_n)$ in Eq. (40) for all nodes X_n and their potential parent sets π_n $(n = 1, \ldots, N)$ before starting the order MCMC simulation. The two likelihoods required for computing Eq. (39) can then be computed from these cached lists, and the local scores in Eq. (40) do not have to be re-computed every time when needed. Friedman and Koller (2003) also introduce a pruning approach which can be used to further reduce the computational complexity of the summations in Eq. (40) for large networks.
Subsequently, in a second step, having sampled node orders, $\prec_1, ..., \prec_T$, from the posterior distribution, $P(\prec \mid \mathcal{D})$, a sample of DAGs can be obtained by a very simple

sampling approach (Friedman and Koller, 2003): Given the order, $\prec$, for each node, X_n, its parent set, π_n, can be sampled independently from the following distribution:

$$P(\pi_n|\ \prec, \mathcal{D}) = \frac{\Psi(\mathcal{D}_n^{\pi_n}) \cdot P(\pi_n) \cdot I(X_n, \pi_n, \prec)}{\sum_\pi \Psi(\mathcal{D}_n^{\pi}) \cdot P(\pi) \cdot I(X_n, \pi, \prec)} \tag{41}$$

where $\Psi(\mathcal{D}_n^{\pi})$ was defined in Eq. (12), the sum in the denominator is over all possible parent sets, π, of X_n, and $I(X_n, \pi, \prec)$ is an indicator function, which is 1 if all nodes in π precede node X_n in the order $\prec$, and 0 otherwise. Sampling a parent set, π_n, for each node, X_n, independently from Eq. (41) yields a complete DAG, $\mathcal{G} = \{\pi_1, \ldots, \pi_N\}$, see Friedman and Koller (2003) for further details.
Although Friedman and Koller (2003) show that order MCMC is superior to structure MCMC with regard to convergence and mixing of the resulting Markov chain, the method is not without shortcomings. When the likelihood term has a low weight, e.g. if there are few observations only, then the graph prior distribution has a noticeable effect on the posterior probabilities. That is, the assumption that each node order, $\prec$, has the same prior probability, $P(\prec) = \frac{1}{N!}$, leads to a change of the form of the originally determined prior over DAGs, $P(\mathcal{G})$. DAGs that are consistent with more orders are more likely than DAGs consistent with fewer orders. For instance, the (empty) DAG without any edges can be sampled from all $N!$ node orders, while a DAG of the type $X_{\sigma(1)} \rightarrow X_{\sigma(2)} \rightarrow .... \rightarrow X_{\sigma(N)}$ can be sampled from one single node order only, namely $\prec = (X_{\sigma(1)}, ..., X_{\sigma(N)})$. To recapitulate this: While the prior on the graphs given the node order, $P(\mathcal{G})$, can be specified, the explicit computation of the prior over graphs requires a marginalization over orders: $P(\mathcal{G}) = \sum_\prec P(\mathcal{G}|\ \prec) \cdot P(\prec)$, where

$$P(\mathcal{G}|\ \prec) = \prod_{n=1}^{N} P(\pi_n|\ \prec) \tag{42}$$

$$P(\pi_n|\ \prec) = \frac{P(\pi_n) \cdot I(X_n, \pi_n, \prec)}{\sum_\pi P(\pi) \cdot I(X_n, \pi, \prec)} \tag{43}$$

and $\mathcal{G} = \{\pi_1, \ldots, \pi_N\}$. The distortion inherent in the marginalization means that it is effectively impossible to exactly specify the prior over graphs. This is not necessarily a problem for large data sets, where the dominant contribution to the posterior distribution stems from the marginal likelihood. But it can be a problem in contemporary systems biology, though, where the number of experimental conditions relative to the complexity of the investigated system, and hence the weight of the marginal likelihood, is relatively low.

The burn-in phase:
For both MCMC sampling schemes, structure MCMC and order MCMC, it takes some time until the Markov chain converges to its stationary distribution. Therefore, the Markov chains have to be simulated for "long enough" to ensure that they have reached their stationary distributions. The early phase, before convergence has been reached, of an MCMC simulation is called the *burn-in* phase. After the burn-in phase, samples from the posterior distributions can be taken. However, when every graph (order) after the burn-in phase is sampled, i.e. one graph (order) in each MCMC iteration, the sample of graphs (orders) tends to be auto-correlated. Therefore, the graph (order) sample is usually thinned out by keeping a distance between sampling. Usually, it is thinned out by taking samples at equally spaced iterations; e.g. by sampling every 10-th, 100-th, or 1000-th iteration after the burn-in phase.

2.7 Convergence diagnostics and network reconstruction accuracy

When Bayesian networks are inferred with Markov Chain Monte Carlo (MCMC) simulations, e.g. with structure MCMC or order MCMC, described in Subsection 2.6, the output is a sample of directed graphs, $\mathcal{G}_1, ..., \mathcal{G}_T$, and the second step is to estimate marginal posterior probabilities, $P(F|\mathcal{D})$, of (edge-)features, F, from this sample; see Eq. (4) in Subsection 2.1. For both static and dynamic Bayesian networks the focus is usually on *directed edge features*. There is a *directed edge feature* $F_{n\to j}$ from X_n to X_j in a graph $\mathcal{G}$, symbolically $F_{n\to j}(\mathcal{G}) = 1$, if the graph, $\mathcal{G}$, possesses the directed edge, $X_n \to X_j$, and $F_{n\to j}(\mathcal{G}) = 0$ otherwise. In the context of dynamic Bayesian networks (DBNs) Eq. (4) can be applied directly to the graph sample, $\mathcal{G}_1, ..., \mathcal{G}_T$, and the relative frequency

$$e_{n,j} := \widehat{P}(F_{n\to j}|\mathcal{D}) = \frac{1}{T}\sum_{t=1}^{T} F_{n\to j}(\mathcal{G}_t) \tag{44}$$

is an estimator of the marginal posterior probability of the directed edge feature, $F_{n\to j}$ ($n, j = \{1, \ldots, N\}$, and $n \neq j$ if self-feedback-loops are invalid).
In the context of static Bayesian networks it has to be taken into account that there are equivalence classes of graphs. Equivalent graphs encode for the same set of conditional (in-)dependencies, and thus have the same marginal likelihood, $P(\mathcal{D}|\mathcal{G})$. All graphs within an equivalence class share the same skeleton but edge directions may differ; those edges that appear with both directions within an equivalence class cannot be interpreted causally. As described in Subsection 2.2, each equivalence class can be represented by a completed partially directed acyclic graph (CPDAG), which comprises both directed and undirected edges. A CPDAG possesses the directed edge $X_n \to X_j$ if all graphs in the corresponding equivalence class share the edge $X_n \to X_j$, and the CPDAG possesses the undirected edge $X_n - X_j$ if all graphs in the equivalence class have an edge connection between X_n and X_j, but do not agree with regard to the orientation.[19] With regard to directed edge features, undirected edges $X_n - X_j$ can be either withdrawn (i.e. ignored) or the undirected edges can be substituted for two directed edges with opposite directions, symbolically $X_n \leftrightarrow X_j$. If the latter interpretation is chosen, the marginal posterior probability of the directed edge feature $F_{n\to j}$ can be estimated as follows:

$$e_{n,j} := \widehat{P}(F_{n\to j}|\mathcal{D}) = \frac{1}{T}\sum_{t=1}^{T} F_{n\to j}(\mathcal{G}_t^\star) \tag{45}$$

where $\mathcal{G}_t^\star$ is the CPDAG of the t-th graph, $\mathcal{G}_t$, from the sample, $\mathcal{G}_1, \ldots, \mathcal{G}_T$, and $F_{n\to j}$ is a binary indicator variable, which is one if the CPDAG $\mathcal{G}_t^\star$ contains either the directed edge $X_n \to X_j$ or the undirected edge $X_n - X_j$, where the latter is interpreted as superposition of two oppositely oriented edges, symbolically $X_n \leftrightarrow X_j$.
Let $e_{n,j} = \widehat{P}(F_{n\to j}|\mathcal{D})$ denote the marginal edge posterior probability of the individual edge $X_n \to X_j$ ($n, j \in \{1, \ldots, N\}$ and $n \neq j$ for static Bayesian networks and dynamic Bayesian networks where self-feedback-loops are invalid). Standard convergence diagnostics compare the inference results of independent and differently seeded MCMC simulations on the same data set. Assuming that H independent MCMC simulations have been performed, the marginal edge posterior probabilities can be computed independently from each MCMC simulation output $h = 1, \ldots, H$. Let $\{e_{n,j,h}\}$ denote

[19] That is, there is at least one graph in the equivalence class in which the edge $X_n \to X_j$ can be found, and there is at least one other graph in the same equivalence class that possesses the oppositely oriented edge $X_n \leftarrow X_j$.

the set of marginal edge posterior probabilities computed from the graph sample, generated by the h-th MCMC simulation ($h = 1, \ldots, H$). As a first check of convergence, the marginal edge posterior probability sets $\{e_{n,j,h_1}\}$ and $\{e_{n,j,h_2}\}$ computed from two independent MCMC simulations h_1 and h_2 ($h_1, h_2 \in \{1, \ldots, H\}$, $h_1 \neq h_2$) can be plotted against each other in a scatter plot. Ideally, all MCMC simulations should have yield (almost) identical marginal edge posterior probabilities so that all points $(e_{n,j,h_1}, e_{n,j,h_2})$ in the scatter plot are located around the diagonal; i.e. independent MCMC simulations should yield similar (strongly correlated) results. Although this simple diagnostic is not a sufficient criterion for convergence, since all H simulations can have become trapped in the same local optimum, the assumption of convergence can surely be withdrawn if there are points $(e_{n,j,h_1}, e_{n,j,h_2})$ in the scatter plot that strongly deviate from the diagonal.
Another more commonly applied standard convergence diagnostic for MCMC sampling schemes is based on potential scale reduction factors (PSRFs), which are usually monitored along the number of MCMC iterations. Again, assuming that H independent MCMC simulations, with $2s$ MCMC iterations each, have been performed on the same data set. The first s iterations can be discarded as burn-in phase, and in the sampling phase, I_s graph samples can be taken from the remaining s MCMC iterations.[20] Again, let $e_{n,j,h}$ denote the marginal edge posterior probability of the edge $X_n \rightarrow X_j$ computed from the h-th MCMC simulation. For each individual edge $X_n \rightarrow X_j$ the "between-chain" variance, $\mathcal{B}(n,j)$, and the "within-chain" variance, $\mathcal{W}(n,j)$, of its marginal edge posterior probability are defined as follows (Brooks and Gelman, 1998):

$$\mathcal{B}(n,j) = \frac{1}{H-1} \sum_{h=1}^{H} (e_{n,j,h} - \overline{e}_{n,j,.})^2 \tag{46}$$

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

$$\mathcal{W}(n,j) = \frac{1}{H(I_s-1)} \sum_{h=1}^{H} \sum_{t=1}^{I_s} (\mathcal{G}_t^h(n,j) - e_{n,j,h})^2 \tag{47}$$

where $\mathcal{G}_t^h$ is the t-th graph in the sample $\mathcal{G}_1^h, \ldots, \mathcal{G}_{I_s}^h$ obtained from the h-th MCMC simulation, and $\mathcal{G}_t^h(n,j) = 1$ if $\mathcal{G}_t^h$ contains the edge $X_n \rightarrow X_j$ while $\mathcal{G}_t^h(n,j) = 0$ otherwise. Following Brooks and Gelman (1998) the $PSRF(n,j)$ of the individual edge $X_n \rightarrow X_j$ is then given by:

$$PSRF(n,j) = \frac{(1 - \frac{1}{I_s})\mathcal{W}(n,j) + (1 + \frac{1}{H})\mathcal{B}(n,j)}{\mathcal{W}(n,j)} \tag{48}$$

where PSRF values near 1 indicate that each of the H MCMC simulations is close to the stationary distribution. As PSRF-based convergence diagnostic, the fraction of edges, $\mathcal{C}(\xi)$, whose PSRF is lower than a pre-defined threshold value, ξ, can be used. For dynamic Bayesian networks, for which self-feedback-loops are valid, there are N^2 edges and $\mathcal{C}(\xi)$ is given by:

$$\mathcal{C}(\xi) = \frac{1}{N^2} \sum_{n=1}^{N} \sum_{j=1}^{N} Z_{PSRF<\xi}(PSRF(n,j)) \tag{49}$$

where $Z_{PSRF<\xi}(PSRF(n,j)) = 1$ if $PSRF(n,j) < \xi$, and $Z_{PSRF<\xi}(PSRF(n,j)) = 0$ otherwise. For static Bayesian networks and dynamic Bayesian networks without

[20] The number of samples I_s that can be taken in the sampling-phase is limited by the sampling phase length s and the distance (no. of iterations) between sampling.

self-feedback loops there are only $N \times (N-1)$ directed edges and $\mathcal{C}(\xi)$ is given by:

$$\mathcal{C}(\xi) = \frac{1}{N(N-1)} \sum_{n,j:n \neq j} Z_{PSRF<\xi}(PSRF(n,j)) \tag{50}$$

where $Z_{PSRF<\xi}(PSRF(n,j)) = 1$ if $PSRF(n,j) < \xi$, and $Z_{PSRF<\xi}(PSRF(n,j)) = 0$ otherwise. The PSRF-based criterion is advantageous to scatter plot diagnostics, since it can straightforwardly be monitored along the number of MCMC iterations, $2s$.

Finally, when the true graph or at least a gold-standard graph for the domain is known, the concept of *ROC curves* and *AUROC values* can be used to evaluate the learning performance of Bayesian network inference. Let $e^{\star}_{i,j} = 1$ indicate that there is a directed edge feature $X_i \to X_j$ in the true graph, while $e^{\star}_{i,j} = 0$ indicates that this edge feature is not given in the true graph. The Bayesian network approach outputs a posterior probability estimate, $e_{i,j} = \widehat{P}(F_{i \to j}|\mathcal{D}) \in [0,1]$, for each (binary) directed edge feature, $e^{\star}_{i,j} \in \{0,1\}$.

Let $\epsilon(\zeta) = \left\{ X_i \to X_j | e_{i,j} = \widehat{P}(F_{i \to j}|\mathcal{D}) > \zeta \right\}$ denote the set of directed edges whose posterior probability estimates, $e_{i,j}$, exceed a given threshold, $\zeta \in [0,1]$. Since the true network topology is known, for each $\epsilon(\zeta)$ the number of true positive $TP[\zeta]$, false positive $FP[\zeta]$, true negative $TN[\zeta]$, and false negative $FN[\zeta]$ edges can be counted. And from these values the true positive rate $TPR[\zeta] = TP[\zeta]/(TP[\zeta] + FN[\zeta])$ (also called *recall* or *sensitivity*), the false positive rate $FPR[\zeta] = FP[\zeta]/(TN[\zeta] + FP[\zeta])$ (also called *inverse specificity*), and the precision $PRE[\zeta] = TP[\zeta]/(TP[\zeta] + FP[\zeta])$ can be computed.

Plotting the sensitivities $TPR[\zeta]$ (vertical axis) against the corresponding inverse specificities $FPR[\zeta]$ (horizontal axis) and connecting neighboring points by linear interpolation gives the receiver operator characteristic (ROC) curve. The area under the ROC curve (AUC or 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 expectation. 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 by plotting the precision values, $PRE[\zeta]$, (vertical axis) against the corresponding recall values, $TPR[\zeta]$, (horizontal axis). Different from ROC curves, neighboring points in the PR curve cannot be connected by straight lines and a non-linear interpolation is required.[21] The interpolation scheme described in Davis and Goadrich (2006) can be employed. As the precision is not defined for TP=0 and FP=0 (PRE=0/0), the Precision-Recall curve is usually integrated on the interval $[(1/E), 1]$ only, where E is the number of edges of the true graph; i.e. one has to restrict on the area where at least one of the true edges has been learned. The area under the precision-recall curve can be referred to as AUC or AUC-PR value. A more detailed description and a theoretical comparison of both criteria, ROC curves and PR curves, can be found in Davis and Goadrich (2006).

[21] The linear interpolation has to be done in terms of the true positives (TPs) and false positives (FPs) which corresponds to a nonlinear interpolation in the precision recall representation.

3 Research contributions

3.1 List of selected publications

1. **Grzegorczyk, M.** and Husmeier, D. (2008) Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move. *Machine Learning*, **71(2-3)**, 265-305.

2. **Grzegorczyk, M.**, Husmeier, D., Edward, D.E., Ghazal, P., and Millar, A.J. (2008) Modelling non-stationary gene regulatory processes with a non-homogeneous Bayesian network network and the allocation sampler. *Bioinformatics*, **24(18)**, 2071-2078.

3. Ickstadt, K., Bornkamp, B., **Grzegorczyk, M.**, Wieczorek, J., Sheriff, M.R., Grecco, H.E. and Zamir, E. (2010) Nonparametric Bayesian Networks. **In:** *Bayesian Statistics 9*, Bernardo et al. (eds.), Oxford University Press, 283-316.

4. **Grzegorczyk, M.**, Husmeier, D., and Rahnenführer, J. (2011) Modelling non-stationary gene regulatory processes with the BGM model. *Computational Statistics*, **26(2)**, 199-218.

5. **Grzegorczyk, M.**, Husmeier, D., and Rahnenführer, J. (2010) Modelling non-stationary gene regulatory processes. *Advances in Bioinformatics*, **Volume 2010**, online article, 21 pages.

6. **Grzegorczyk, M.** and Husmeier, D. (2009) Non-stationary continuous Bayesian networks. **In:** *Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS2009)*, Bengio et al. (eds.), Curran Associates, 682-690.

7. **Grzegorczyk, M.** and Husmeier, D. (2011) Non-homogeneous dynamic Bayesian networks for continuous data. *Machine Learning*, **83(3)**, 355-419.

8. **Grzegorczyk, M.** and Husmeier, D. (2011) Improvements in the reconstruction of time-varying networks: dynamic programming and regularization by information sharing among genes. *Bioinformatics*, **27(5)**, 693-699.

3.2 Research Goals

As pointed out in Section 1, Bayesian network models are a promising trade-off between over-simplicity and loss of computational tractability when reverse-engineering gene regulatory networks and protein pathways in contemporary systems biology research (Cantone *et al.*, 2009). However, in comparative evaluation studies, e.g. see Werhli *et al.* (2006) or Grzegorczyk (2006), it was found that Bayesian networks do not always yield significantly better network reconstruction accuracies than computationally cheaper methods, such as relevance networks and Gaussian graphical models. This suggests that the higher computational costs of Bayesian network inference can be justified only under certain circumstances. For example in Werhli *et al.* (2006) it was found that Bayesian networks perform significantly better than relevance networks and Gaussian graphical models for interventional data, i.e. data that have been collected after active experimental interventions.[22] In Grzegorczyk (2006) it was found that the full potential of Bayesian networks can also be exploited if the underlying network possesses certain topological features.[23] On the other hand, apart from the substantially higher computational costs, Bayesian networks were never found to be inferior to those simpler network reconstruction methods. The increased computational costs of Bayesian network inference stem from the required Markov Chain Monte Carlo (MCMC) simulations, and thus an important research goal is to develop novel faster converging MCMC sampling schemes for Bayesian networks. That is, it is the availability of fast MCMC sampling schemes that makes Bayesian networks competitive to computationally cheaper methods for large networks even if the full modeling potential is not required. The first important research contribution to static Bayesian network methodology was the development of a novel MCMC sampling scheme (Grzegorczyk and Husmeier, 2008), which will be presented in Subsection 3.3.1.

For dynamic time series data Bayesian networks are in competition with mechanistic models based on coupled differential equations. From a theoretical and biological point of view coupled differential equation models yield a detailed description of the cellular regulatory processes, and therefore, mechanistic models seem to be more suitable than dynamic Bayesian network models (Vyshemirsky and Girolami, 2008). However, from a practical perspective mechanistic differential equation models are associated with several serious shortcomings, as already described in Section 1. Most notably, marginal likelihoods cannot be computed in closed form and maximum likelihood estimators have to be employed for evaluating the models. Hence, model selection is impeded by the fact that more complex pathway models always provide a better explanation of the data than less complex ones, rendering these approaches intrinsically susceptible to over-fitting. Standard complexity regularization criteria, such as the Bayesian information criterion BIC (Schwarz, 1978) or Akaike's information criterion AIC (Akaike, 1983), which penalize a model (network) according to its

[22] For example in the form of gene knockouts and gene over-expressions. How to deal with interventional data in Bayesian networks has for example been described in Pournara and Wernisch (2004).

[23] As explained in Section 2.2, in static Bayesian networks not every edge in the underlying directed acyclic graph (DAG) can be interpreted causally. Completed partially directed acyclic graphs (CPDAGs) have to be extracted from the directed acyclic graphs (DAGs), and CPDAGs consist of both: directed and undirected edges. However, every edge that participates in a v-structure and every edge whose reversal would yield a new v-structure can be interpreted causally. Therefore, the number of directed edges that can be extracted with static Bayesian networks strongly depends on the number of v-structures in the true underlying network topology. Loosely speaking, if the true underlying network possesses many v-structures, then static Bayesian networks tend to infer graphs with many directed edges (causalities) what makes them superior to other reverse engineering methods that are based on undirected edges. See Grzegorczyk (2006) and Section 2.2 for more details.

complexity (numbers of parameters[24]), do not provide a sufficient remedy, since these criteria provide reliable results only for unimodal posterior distributions that (at least approximately) follow a multivariate Gaussian distribution. But mechanistic models for regulatory networks are usually defined in terms of laws of chemical kinetics so that the imposed regulatory network interactions are highly nonlinear. This yields highly non-Gaussian marginal likelihoods and posterior distributions for which the regularization criteria systematically fail. Recently, promising novel approaches based on Annealed Importance Sampling (Neal, 2001) and Annealing-Melting Integration (Gelman *et al.*, 2004) for computing marginal likelihoods for mechanistic models have been proposed (Vyshemirsky and Girolami, 2008). However, even when the required computations are carried out on modern parallel computer clusters, as implemented by Vyshemirsky and Girolami (2008), these approaches are currently restricted to very small mechanistic network models with very few network nodes (Vyshemirsky and Girolami, 2008).
Another very important research goal is therefore to develop more flexible dynamic Bayesian network models, which allow for a more detailed description of the true regulatory relationships without losing the marginal likelihood, which intrinsically avoids data over-fitting (Bishop, 2006), as a closed-form scoring function. In Subsection 3.3.2 the *Bayesian Gaussian Mixture* (BGM) Bayesian network model (Grzegorczyk *et al.*, 2008) will be described. The BGM model can be applied to both static (steady-state) and dynamic (time series) data and combines Gaussian Bayesian network methodology with a Bayesian mixture model. Latent variables are used in the BGM model to assign the data points to different mixture components (classes). All components share the same network topology to allow for some information sharing among classes, but each component (data subset) is modeled separately with the Gaussian BGe score. This approach yields a novel probabilistic model that is capable of modeling both non-homogeneous dynamic and non-linear static gene-regulatory processes. However, it is this flexibility that renders the BGM model suboptimal for time series data. By substituting the free allocation of data points for a multiple changepoint process, the intuitive prior notion, that adjacent time points are likely to be governed by similar distributions, can be taken into account. Employing a multiple changepoint process rather than a free allocation mixture model for data segmentation yields the *dynamic variant* BGM_D (Grzegorczyk *et al.* (2010) and Grzegorczyk *et al.* (2011)) of the BGM model, which will be described in Subsection 3.3.3. Both models BGM and its dynamic variant BGM_D employ network-wide changepoints that are common to all nodes of the network. That is, in these two models all nodes are affected by the changepoints in identical ways. Introducing the concept of node-specific changepoints enables a greater model flexibility, since individual nodes can then be affected by changing processes in different ways. This generalization of the BGM_D model yields the *changepoint BGe* (cpBGe) model (Grzegorczyk and Husmeier, 2009c), which will be discussed in Subsection 3.3.4. The earlier work on the cpBGe model (Grzegorczyk and Husmeier, 2009c) pursued inference with Reversible Jump Markov Chain Monte Carlo (RJMCMC) and combined the classical structure MCMC sampling scheme from Subsection 2.6 with additional birth, death, and re-allocation moves for the node-specific changepoints (Green, 1995). The dynamic programming scheme of Fearnhead (2006) allows for sampling changepoints from the proper conditional distribution directly so that no RJMCMC sampling scheme with potential mixing and convergence problems is required. As shown in Subsection 3.3.5, incorporating Fearnhead's dynamic programming scheme within a Gibbs sampling scheme for the cpBGe model (Grzegorczyk and Husmeier, 2011b), substantially improves convergence and mixing. Finally, in Subsection 3.3.6 the novel *regularized cpBGe model* will be presented (Grzegorczyk and Husmeier, 2011a). The regularized

[24]The number of parameters of a network model depends on the number of edges.

Model	**BGM**	**BGM$_D$**	**cpBGe**	**improved cpBGe**	**regularized cpBGe**
Subsection	3.3.2	3.3.3	3.3.4	3.3.5	3.3.6
Sampling scheme	RJMCMC	RJMCMC	RJMCMC	Gibbs	Gibbs and RJMCMC
Allocation format	Free allocation	Change-points	Change-points	Change-points	Change-points
Required data	static or dynamic	dynamic	dynamic	dynamic	dynamic
Segment-ation(s)	network-wide	network-wide	node-specific	node-specific	node-cluster specific

Table 1: **Overview to the non-homogeneous dynamic Bayesian network models that have been contributed to systems biology research.** These novel Bayesian network models will be described in more detail in Subsection 3.3 and are non-homogeneous with respect to the network parameter distributions, while the network topology is the same for all segments. Only the BGM model can be used for static data, since it employs a free allocation of data points. The other models employ multiple changepoint processes and are therefore restricted to dynamic time series data. The changepoints in the two models BGM and BGM$_D$ are network-wide, i.e. common to the whole network. The (improved) cpBGe model yields a higher modeling flexibility by introducing node-specific changepoints that vary from node to node. Individual nodes can then be affected by changing processes in different ways. While the cpBGe model was originally proposed with a RJMCMC sampling scheme for inference, the improved cpBGe model employs a Gibbs sampling scheme and incorporates dynamic programming for changepoint sampling. The regularized cpBGe model infers node clusters that share the same changepoints. That is, it regularizes between BGM$_D$ and cpBGe and subsumes both models as limiting cases. The corresponding MCMC sampling scheme for the regularized cpBGe model uses both (i) RJMCMC steps for inferring clusters of nodes and (ii) Gibbs sampling steps for inferring the network topology and the changepoint sets for the clusters.

cpBGe model extends the cpBGe model by introducing a Bayesian clustering and information sharing scheme among the node-specific changepoints, which provides a mechanism for automatic model complexity tuning. The regularized cpBGe model infers clusters of network nodes, and nodes in the same cluster share the same changepoints. That is, the regularized cpBGe model regularizes between the BGM$_D$ and the cpBGe model and subsumes these two models as limiting cases: The regularized cpBGe model corresponds to the BGM$_D$ model if there is one single cluster to which all network nodes are assigned so that the changepoints of this cluster are shared by the whole network; and the regularized cpBGe model corresponds to the cpBGe model if there is a separate cluster for every single node so that changepoints are effectively node-specific. A brief overview to these novel non-homogeneous dynamic Bayesian network models, which will be described in more detail in Subsection 3.3, can be found in Table 1.

3.3 Brief discussion of publications

3.3.1 Improving the structure MCMC sampler for static Bayesian networks

- **ORIGINAL PUBLICATION: Grzegorczyk, M.** and Husmeier, D. (2008) Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move. *Machine Learning*, **71(2-3)**, 265-305.

Summary:
The objective was to improve mixing and convergence of the classical structure MCMC sampling scheme for static Bayesian networks, described in Subsection 2.6. This could be achieved by introducing a novel and more involved edge-reversal move. The main advantage of the upgraded structure MCMC sampling scheme is that it incurs the bias intrinsic to the order MCMC scheme of Friedman and Koller (2003) without increased computational costs. On various synthetic network data from the UCI repository (Newman *et al.*, 1998) it was demonstrated that the improved structure MCMC sampler compares favorably with both sampling schemes: (i) the (original) structure MCMC sampler and (ii) the order MCMC sampler. That is, the new sampling scheme converges approximately as fast as the order MCMC sampling scheme, i.e. substantially faster than the classical structure MCMC sampling scheme, but different from the order MCMC sampling scheme, the new upgraded structure MCMC sampler does not yield biased results.

Motivation:
The structure MCMC sampling scheme for static Bayesian networks was described in Subsection 2.6 and proposes exclusively moves based on single-edge-operations, such as single edge deletions, additions and reversals in the configuration space of directed acyclic graphs. Since these single-edge-moves are relatively small in the configuration space of graphs, the structure MCMC algorithm tends to get trapped in local maxima, and mixing and convergence are rather slow; especially when the posterior distribution, $P(\mathcal{G}|\mathcal{D})$, has various peaks. Mixing and convergence is considerably better for the order MCMC sampling scheme in the space of node orders, as proposed by Friedman and Koller (2003) and briefly summarized in Subsection 2.6. But the disadvantage of order MCMC is that the prior on graphs, $P(\mathcal{G})$, cannot be defined explicitly; see Subsection 2.6 for details. An approach that is intrinsically unable to specify the prior, $P(\mathcal{G})$, explicitly is not entirely satisfactory, since the graph prior distribution can have a substantial influence on the posterior distribution and hence on the outcome of inference; in particular for sparse data. Sampling in the space of node orders is therefore not a sufficient remedy, and the idea was to improve the original structure MCMC sampler by introducing a new edge reversal move. The key idea behind this novel move is to allow for larger modifications of the current DAG so that local maxima can be left.
Static Bayesian networks are based on directed acyclic graphs (DAGs) and the structure MCMC approach allows the reversal of an edge, only if the reversal leads to a new valid acyclic graph. Therefore, the first obvious problem is that it depends on the overall structure of the current graph which edges $X_i \rightarrow X_j$ can be reversed to $X_i \leftarrow X_j$. This shortcoming can be avoided by changing the parent sets π_i and π_j of both nodes that are connected by the edge in a more involved way. The new move samples two new parent sets $\tilde{\pi}_i$ and $\tilde{\pi}_j$ for both nodes such that the corresponding

edge $X_i \to X_j$ points into the opposite direction, symbolically $X_i \leftarrow X_j$, in the new graph $\tilde{\mathcal{G}}$ and the overall structure of $\tilde{\mathcal{G}}$ is acyclic again. But even for those edges that can already be reversed by the classical structure MCMC reversal move, this new edge reversal move has a clear advantage: By sampling completely new parent sets for both nodes, instead of changing the direction of one single edge only, the acceptance probability can be increased substantially. This is because the classical edge reversal move does not take into consideration whether the reversal of the single edge is useful in combination with the other nodes in the current parent sets π_i and π_j. That is, the other nodes in π_i and π_j are unadapted to the reversed edge. The new reversal move guarantees that both parent sets are merely resampled according to their posterior probabilities (scores). The new parent sets $\tilde{\pi}_i$ and $\tilde{\pi}_j$ can be expected to be "higher-scoring" on average, as they are adapted to the edge reversal from $X_i \to X_j$ to $X_i \leftarrow X_j$.

The mathematical model:
In the context of static Bayesian networks, described in Subsection 2.2, the following additional definitions can be introduced: For node X_n let $\mathcal{G}^{\{X_n\}\leftarrow\emptyset}$ denote the graph obtained by setting the parent set, π_n, of node X_n to the empty set, that is, the new graph is obtained by *orphaning* node X_n (i.e. by removing from $\mathcal{G}$ all edges converging on X_n). Correspondingly, for two nodes X_i and X_j let $\mathcal{G}^{\{X_i,X_j\}\leftarrow\emptyset}$ be the graph obtained by *orphaning* both nodes X_i and X_j. The graph in which the old parent set, π_n, of node X_n is replaced for a new parent set, $\tilde{\pi}_n$, is referred to as $\mathcal{G}^{X_n\leftarrow\tilde{\pi}_n}$, and correspondingly the graph in which the old parent sets of both nodes X_i and X_j are replaced by new parent sets, $\tilde{\pi}_i$ and $\tilde{\pi}_j$, respectively, is denoted $\mathcal{G}^{X_i\leftarrow\tilde{\pi}_i,X_j\leftarrow\tilde{\pi}_j}$. These graph structures $\mathcal{G}^{X_n\leftarrow\tilde{\pi}_n}$ and $\mathcal{G}^{X_i\leftarrow\tilde{\pi}_i,X_j\leftarrow\tilde{\pi}_j}$ are not necessarily acyclic ones. It is the indicator function $\delta : \{\mathcal{G}\} \to \{0,1\}$ in the space of directed (not necessarily acyclic) graphs that indicates if a graph is acyclic or not; i.e. $\delta(\mathcal{G}) = 1$ if $\mathcal{G}$ is *acyclic*, and $\delta(\mathcal{G}) = 0$ if $\mathcal{G}$ is cyclic. Moreover, two partition functions are required. Given a DAG, $\mathcal{G} = \{\pi_1, \ldots, \pi_N\}$, and a graph prior distribution that can be factorized, as in Eq. (37), $P(\mathcal{G}) = \prod_{n=1}^{N} P(\pi_n)$, the first partition function, $Z(.)$, is a sum of local scores over all those parent sets, π, of node X_n for which $\mathcal{G}^{X_n\leftarrow\pi}$ is a valid DAG.

$$Z(X_n|\mathcal{G}) \;:=\; \sum_{\pi:\,\delta(\mathcal{G}^{X_n\leftarrow\pi})=1} \Psi(\mathcal{D}_n^{\pi}) \cdot P(\pi) \tag{51}$$

where the local scores, $\Psi(.)$, were specified in Eq. (12). Given a DAG, $\mathcal{G}$, and a node, X_m, the second partition function, $Z^{\star}(.)$, is a sum of local scores over all those parent sets, π, of node X_n, which contain X_m, symbolically: $X_m \in \pi$, and for which $\mathcal{G}^{X_n\leftarrow\pi}$ is a valid DAG.

$$Z^{\star}(X_n|\mathcal{G}, X_m) \;:=\; \sum_{\substack{\pi\,:\;\delta(\mathcal{G}^{X_n\leftarrow\pi})=1 \\ X_m\,\in\,\pi}} \Psi(\mathcal{D}_n^{\pi}) \cdot P(\pi). \tag{52}$$

The new edge reversal (REV) move consists of three steps and works as follows:

First step: *Given a DAG, $\mathcal{G}$, randomly select one of its directed edges $X_i \to X_j$ from a uniform distribution over all edges of $\mathcal{G}$ and orphan both nodes X_i and X_j to obtain the new DAG: $\mathcal{G}_{\odot} := \mathcal{G}^{\{X_i,X_j\}\leftarrow\emptyset}$.*

Second step: *Sample a new parent set, $\tilde{\pi}_i$, for node X_i which contains X_i's former child node X_j, symbolically $X_j \in \tilde{\pi}_i$, and does not lead to any directed cycles when added to $\mathcal{G}_{\odot}$, symbolically: $\delta(\mathcal{G}_{\odot}^{X_i\leftarrow\tilde{\pi}_i}) = 1$. The new parent set, $\tilde{\pi}_i$, is sampled*

from the following modified Boltzmann distribution:

$$Q(\tilde{\pi}_i|\mathcal{G}_\odot, X_j) \;=\; \frac{\Psi(\mathcal{D}_i^{\tilde{\pi}_i}) \cdot P(\tilde{\pi}_i) \cdot \delta(\mathcal{G}_\odot^{X_i \leftarrow \tilde{\pi}_i}) \cdot I(\tilde{\pi}_i, X_j)}{Z^\star(X_i|\mathcal{G}_\odot, X_j)}, \tag{53}$$

whereby $I(\tilde{\pi}_i, X_j) = 1$ if $X_j \in \tilde{\pi}_i$, and $I(\tilde{\pi}_i, X_j) = 0$ otherwise, and the partition function, $Z^\star(.)$, was defined in Eq. (52). Having sampled the new parent set, $\tilde{\pi}_i$, for node X_i, set $\mathcal{G}_\oplus := \mathcal{G}_\odot^{X_i \leftarrow \tilde{\pi}_i} = \mathcal{G}^{X_i \leftarrow \tilde{\pi}_i, X_j \leftarrow \emptyset}$, so that $\mathcal{G}_\oplus$ is a valid DAG that contains the oppositely oriented edge, $X_i \leftarrow X_j$.

Third step : Finally, sample a new parent set, $\tilde{\pi}_j$, for node X_j from the following distribution:

$$Q(\tilde{\pi}_j|\mathcal{G}_\oplus) \;=\; \frac{\Psi(\mathcal{D}_j^{\tilde{\pi}_j}) \cdot P(\tilde{\pi}_j) \cdot \delta(\mathcal{G}_\oplus^{X_j \leftarrow \tilde{\pi}_j})}{Z(X_j|\mathcal{G}_\oplus)}, \tag{54}$$

where the partition function, $Z(.)$, was defined in Eq. (51). That is, sample a new parent set, $\tilde{\pi}_j$, for node X_j, so that the graph, $\mathcal{G}_\oplus^{X_j \leftarrow \tilde{\pi}_j}$, which possesses the reversed edge, $X_j \leftarrow X_i$, remains acyclic. The DAG, $\tilde{\mathcal{G}} := \mathcal{G}_\oplus^{X_j \leftarrow \tilde{\pi}_j} = \mathcal{G}^{X_i \leftarrow \tilde{\pi}_i, X_j \leftarrow \tilde{\pi}_j}$, is proposed by the new reversal move.

The proposal probability, $Q^\triangleright(\tilde{\mathcal{G}}|\mathcal{G})$, of the REV move from $\mathcal{G}$ to $\tilde{\mathcal{G}}$ is then given by:

$$Q^\triangleright(\tilde{\mathcal{G}}|\mathcal{G}) = \frac{1}{N^\dagger} \cdot \frac{\Psi(\mathcal{D}_i^{\tilde{\pi}_i}) \cdot P(\tilde{\pi}_i) \cdot \delta(\mathcal{G}_\odot^{X_i \leftarrow \tilde{\pi}_i}) \cdot I(\tilde{\pi}_i, X_j)}{Z^\star(X_i|\mathcal{G}_\odot, X_j)} \times \frac{\Psi(\mathcal{D}_j^{\tilde{\pi}_j}) \cdot P(\tilde{\pi}_j) \cdot \delta(\mathcal{G}_\oplus^{X_j \leftarrow \tilde{\pi}_j})}{Z(X_j|\mathcal{G}_\oplus)} \tag{55}$$

where $N^\dagger$ is the number of edges in $\mathcal{G}$, $\mathcal{G}_\odot := \mathcal{G}^{X_i \leftarrow \emptyset, X_j \leftarrow \emptyset}$, $\mathcal{G}_\oplus := \mathcal{G}_\odot^{X_i \leftarrow \tilde{\pi}_i} = \mathcal{G}^{X_i \leftarrow \tilde{\pi}_i, X_j \leftarrow \emptyset}$, and the partition functions were defined in Eqns. (51-52). The indicator function, $I(\tilde{\pi}_i, X_j)$, is equal to 1 if $\tilde{\pi}_i$ contains X_j, and 0 otherwise, while the function $\delta(.)$ indicates acyclic graphs.
Each REV move changes the parent sets of two nodes and leaves the parent sets of all other nodes unchanged. With regard to the computation of the acceptance probability of the Metropolis Hastings algorithm a complementary REV move leading backward from $\tilde{\mathcal{G}}$ to $\mathcal{G}$ has to be designed, and it turns out (see original paper (Grzegorczyk and Husmeier, 2008) for details) that for each REV move from $\mathcal{G}$ to $\tilde{\mathcal{G}}$ by reversing the edge $X_i \rightarrow X_j$, there is exactly one inverse REV move leading back from $\tilde{\mathcal{G}}$ to $\mathcal{G}$. The inverse move selects the edge $X_j \rightarrow X_i$ in $\tilde{\mathcal{G}}$ for edge reversal, and orphaning both nodes X_i and X_j in the first step yields the DAG $\tilde{\mathcal{G}}_\odot$. In the second step the parent set π_j of X_j in $\mathcal{G}$ has to be re-sampled and assigned as new parent set for X_j in $\tilde{\mathcal{G}}_\odot$, which gives the DAG $\tilde{\mathcal{G}}_\oplus$. Finally, in the third step the parent set π_i of X_i in $\mathcal{G}$ has to be re-sampled and assigned as the new parent set for X_i in $\tilde{\mathcal{G}}_\oplus$, which gives the DAG $\mathcal{G}$ again.
The acceptance probability for a REV move from $\mathcal{G}$ to the graph $\tilde{\mathcal{G}}$ ($\mathcal{G} \neq \tilde{\mathcal{G}}$) is given by: $A^\triangleright(\tilde{\mathcal{G}}|\mathcal{G}) = \min\left\{1, R^\triangleright(\tilde{\mathcal{G}}|\mathcal{G})\right\}$ where

$$R^\triangleright(\tilde{\mathcal{G}}|\mathcal{G}) = \frac{P(\tilde{\mathcal{G}}|\mathcal{D})}{P(\mathcal{G}|\mathcal{D})} \frac{Q^\triangleright(\mathcal{G}|\tilde{\mathcal{G}})}{Q^\triangleright(\tilde{\mathcal{G}}|\mathcal{G})} \tag{56}$$

$Q^\triangleright(\tilde{\mathcal{G}}|\mathcal{G})$ is the proposal probability for a move from $\mathcal{G}$ to $\tilde{\mathcal{G}}$, and $Q^\triangleright(\mathcal{G}|\tilde{\mathcal{G}})$ is the proposal probability for a move from $\tilde{\mathcal{G}}$ to $\mathcal{G}$. All local scores corresponding to unaffected nodes cancel out in Eq. (56), and all that remains to be computed are the modified partition functions:

$$A^\triangleright(\tilde{\mathcal{G}}|\mathcal{G}) = \min\left\{1, \frac{N^\dagger}{\tilde{N}^\dagger} \cdot \frac{Z^\star(X_i|\mathcal{G}_\odot, X_j)}{Z^\star(X_j|\tilde{\mathcal{G}}_\odot, X_i)} \cdot \frac{Z(X_j|\mathcal{G}_\oplus)}{Z(X_i|\tilde{\mathcal{G}}_\oplus)}\right\} \tag{57}$$

where $N^{\dagger}$ is the number of edges in $\mathcal{G}$, $\tilde{N}^{\dagger}$ is the number of edges in $\tilde{\mathcal{G}}$, and the partition functions were defined in Eqns. (51-52). The transition probability for a move from $\mathcal{G}$ to $\tilde{\mathcal{G}}$ is then given by: $\mathcal{K}^{\triangleright}(\tilde{\mathcal{G}}|\mathcal{G}) = Q^{\triangleright}(\tilde{\mathcal{G}}|\mathcal{G}) \cdot A^{\triangleright}(\tilde{\mathcal{G}}|\mathcal{G})$ where $Q^{\triangleright}(\tilde{\mathcal{G}}|\mathcal{G})$ was defined in Eq. (55) and $A^{\triangleright}(\tilde{\mathcal{G}}|\mathcal{G})$ was defined in Eq. (57).

Inference:
The complexity of the computations required for the proposed edge reversal move can be substantially reduced using the same ideas and approximations that were proposed in Friedman and Koller (2003) for order MCMC. In particular, for each node X_n the products $\Psi^{\star}(\mathcal{D}_n^{\pi_n}) := \Psi(\mathcal{D}_n^{\pi_n}) \cdot P(\pi_n)$ can be pre-computed and stored, instead of re-computing them each time when required.
The classical structure MCMC sampler, described in Subsection 2.6, is based on three different single-edge-operations (edge additions, edge deletions and edge reversals). In the improved structure MCMC algorithm in each iteration either one of the three single-edge-operation moves or one of the novel REV edge reversal moves is performed. More precisely, a probability $p_R \in [0, 1]$ can be pre-determined with which a REV move is chosen, and a classical single-edge-operation move is performed with the inverse probability $p_S := 1 - p_R$. The probability p_R can be adapted during the burn-in period so as to optimize the acceptance probability rate. However, in simulations it was found that the results varied little over quite a large range of p_R around $1/15$; therefore the fixed value $p_R = 1/15$ was used. See original publication (Grzegorczyk and Husmeier, 2008) for details.

Selected application(s):
The improved structure MCMC sampler, which includes the novel edge reversal move (REV), is superior to the two classical samplers, described in Subsection 2.6, since it converges as fast as order MCMC without yielding biased results. The first empirical study of this thesis demonstrates that order MCMC actually yields biased results, while the original structure MCMC sampler and the improved (REV) structure MCMC sampler are unbiased. For small network domains with up to $N = 5$ nodes the true posterior probabilities of directed edge features can be computed with Eq. (3) in reasonable time. And these true posterior probabilities, obtained by exhaustive enumeration of all valid directed acyclic graphs, can be compared with the directed edge feature posterior probability estimates obtained from samples of the three MCMC sampling schemes using Eq. (4). In the original study this was done for various data sets available from the UCI repository (Newman *et al.*, 1998). For networks with too many nodes random subnetworks with $N = 5$ nodes were selected. Figure 3 shows the deviations between the estimated and the true posterior probabilities for the three MCMC samplers for different data subsets from the congressional voting records data set (VOTE) and the solar FLARE data set. The complete VOTE data set is available from the UCI repository and consists of $N = 16$ discrete variables and $m = 435$ observations. The complete FLARE data set is also available from the UCI repository and consists of $N = 13$ discrete variables and $m = 1389$ observations. Data subsets with $m = 10$, $m = 25$, and $m = 50$ observations were randomly selected from the available observations, and the BDe model was employed for the Bayesian networks, since both data sets VOTE and FLARE are discrete.[25] Figure 3 shows that neither the original nor the improved structure MCMC sampler yield biased inference results. All marginal edge posterior probability estimates are located near the reference line, indicating no systematic

[25] In the UCI repository plenty of values for the VOTE data set are missing. Observations, in which the value of one of the 5 selected variables was missing, were removed before randomly drawing the data subsets.

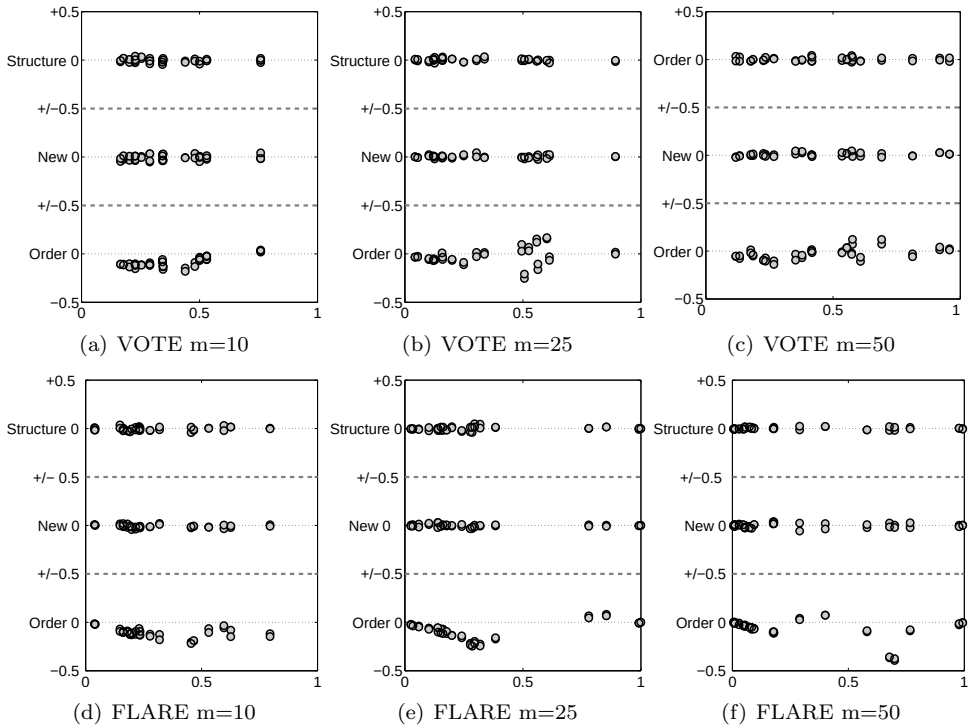
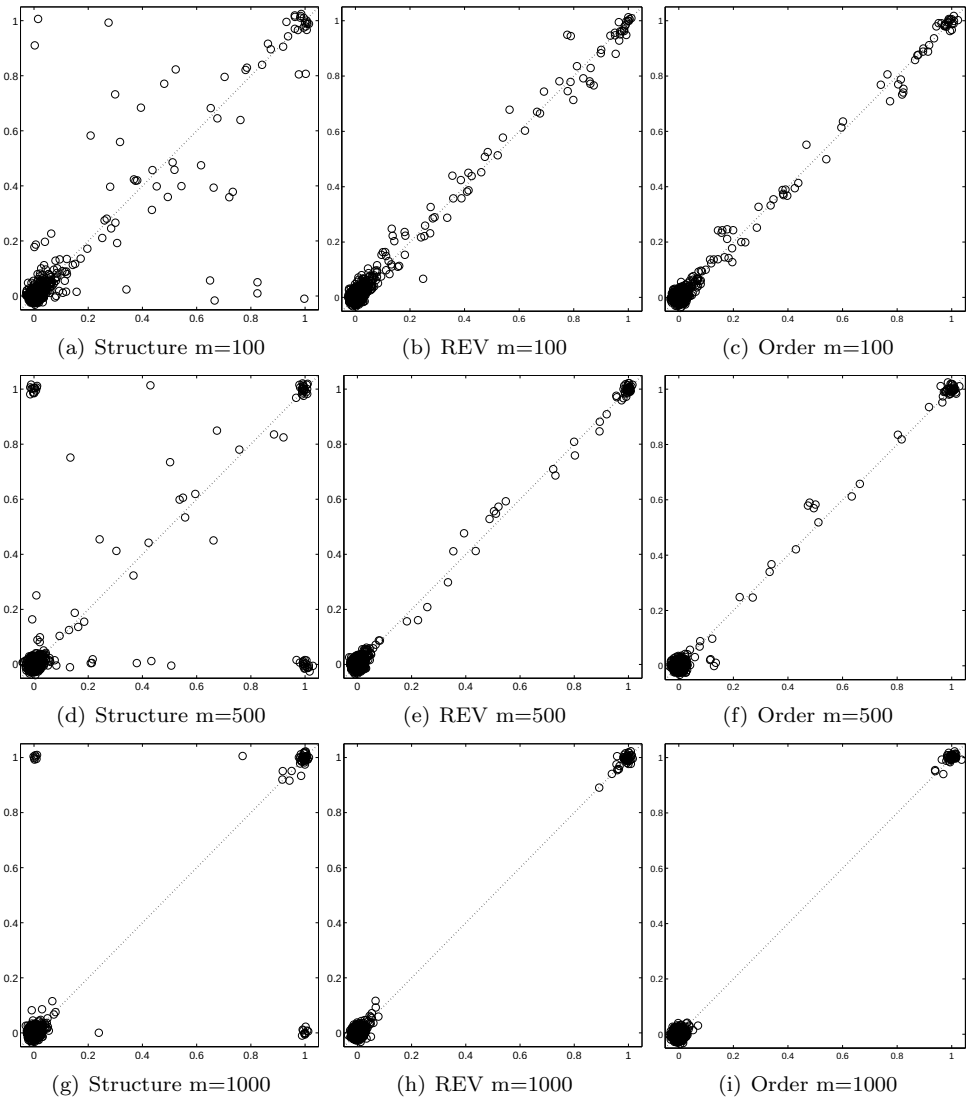


Figure 3: **Deviations between true and estimated directed edge feature posterior probabilities for different subsets of the VOTE and FLARE data.** In each panel the true posteriors are plotted along the x-axis, and parallel to the x-axis there is a thin reference line for each of the three MCMC samplers. The upper line corresponds to structure, the center line corresponds to the new REV structure, and the lower line corresponds to the order MCMC sampler. The reference lines correspond to zero deviation. The points around each line correspond to the edge features: their x-coordinates are the true posterior probabilities while their y-coordinates - relative to the corresponding reference line - reflect their deviations. The true posterior probabilities were obtained by full model averaging.

deviation between the estimated and the true marginal edge posterior probabilities. Only for the order MCMC sampler there are systematic deviations between the estimated and the true marginal edge posterior probabilities. For the sparse data sets with $m = 10$ observations the deviations tend to be stronger, since the biased graph prior distribution has then a larger effect on the posterior distribution. For the data sets with $m = 50$ observations the dominant contribution to the posterior distribution stems from the marginal likelihood and the deviations are less substantial.

To demonstrate that the improved REV structure MCMC sampling scheme converges as fast as the biased order MCMC sampling scheme, independent and differently seeded MCMC simulations were performed with all three samplers on data sets with $m = 100$, $m = 500$, and $m = 1000$ observations from the ALARM network (Beinlich *et al.*, 1989). The ALARM network is a discrete network from the UCI repository which possesses $N = 37$ nodes and 47 directed edges; see Beinlich *et al.* (1989) for details. The MCMC simulation lengths and sample sizes were chosen such that exactly the same computational costs were spend for all three sampling schemes (see original publication (Grzegorczyk and Husmeier, 2008) for details). As described in Subsection 2.7, for each of the three sampling schemes the marginal edge posterior probabilities of directed edge features can be computed from

each independent MCMC simulation output with Eq. (45), and the results can be plotted against each other. Figure 4 shows scatter plots of these marginal directed edge feature posterior probability estimates. It can be seen from the scatter plots that the results of REV structure MCMC and order MCMC do not depend on the initialization. For each sample size m the differently seeded runs yield very similar results, while the classical structure MCMC sampler has not converged sufficiently. It appears that the number of strongly deviating posterior probability estimates increases in the data set size m. This is due to the fact that the posterior probability landscape becomes more rugged with increasing m, which hampers convergence. It can also be observed in the scatter plots that the inference uncertainty is reduced for larger data sets; i.e. the posterior probability becomes more bimodal, with values clustering around 0 and 1.



(a) Structure m=100 (b) REV m=100 (c) Order m=100

(d) Structure m=500 (e) REV m=500 (f) Order m=500

(g) Structure m=1000 (h) REV m=1000 (i) Order m=1000

Figure 4: **Convergence control via scatter plots.** Scatter plots that compare the marginal posterior probability estimates of the directed edge features (see Eq. (45) in Subsection 2.7) on the Alarm network (Beinlich *et al.*, 1989) data sets with $m = 100$, $m = 500$, and $m = 1000$ observations. Left column: structure MCMC, center column: REV structure MCMC, and right column: order MCMC. In each plot every point corresponds to a single directed edge feature and its coordinates were estimated from two DAG samples of size 1000 obtained by two independent and differently seeded MCMC runs. The x coordinates were estimated from a sample obtained by an empty seeded (structure and REV-structure MCMC) or randomly seeded (order MCMC) run, and the y coordinates were estimated from a sample obtained by a greedy search seeded MCMC run. The coordinates of all points were randomly perturbed (by adding an $N(0, 0.01^2)$-distributed error to each coordinate) to visualize clusters of points.

3.3.2 Modeling non-homogeneous Bayesian networks with a free allocation model

- **ORIGINAL PUBLICATION no. 1: Grzegorczyk, M.**, Husmeier, D., Edward, D.E., Ghazal, P., and Millar, A.J. (2008) Modelling non-stationary gene regulatory processes with a non-homogeneous Bayesian network network and the allocation sampler. *Bioinformatics*, **24(18)**, 2071-2078.
- **ORIGINAL PUBLICATION no. 2:** Ickstadt, K., Bornkamp, B., **Grzegorczyk, M.**, Wieczorek, J., Sheriff, M.R., Grecco, H.E. and Zamir, E. (2010) Nonparametric Bayesian Networks. **In:** *Bayesian Statistics 9*, Bernardo et al. (eds.), Oxford University Press, 283-316.

Summary:
The objective was to propose and evaluate a probabilistic approach based on Bayesian networks for modeling *non-homogeneous* and *non-linear* gene-regulatory processes. The method combines Bayesian networks with a Bayesian mixture model that uses latent variables to assign individual data points to different mixture components (classes). The practical inference follows the Bayesian paradigm, i.e. the network, the number of classes and the assignment of latent variables can be sampled with Markov Chain Monte Carlo (MCMC) from the posterior distribution, using the recently proposed allocation sampler (Nobile and Fearnside, 2007) as an alternative to Reversible Jump Markov Chain Monte Carlo (RJMCMC). The novel model is called the *Bayesian Gaussian Mixture* (BGM) Bayesian network model, and was evaluated using three criteria: network reconstruction, statistical significance and biological plausibility. In terms of network reconstruction, improved results were found for both (i) a synthetic network of known structure and (ii) for a small real regulatory network derived from the literature. In the first publication (Grzegorczyk *et al.* (2008)), the statistical significance of the improvement was assessed on gene expression time series for two different systems (viral challenge of macrophages, and circadian rhythms in plants), where the proposed new BGM Bayesian network model tends to outperform the classical linear Gaussian BGe model for Bayesian networks. Regarding biological plausibility, it was found that the inference results obtained with the proposed BGM model are in excellent agreement with biological findings, predicting dichotomies that one would expect to find in the studied biological systems. The second publication (Ickstadt *et al.* (2010)) is more theoretically orientated and discusses the novel BGM model from a Bayesian perspective in the broader context of non-parametric Bayesian mixture models. Furthermore, the biochemical processes of protein binding in cellular signalling cascades were modeled with a system of coupled differential equations to generate realistic data sets, on which the performance of the BGM model could be evaluated properly; see Ickstadt *et al.* (2010) for details.

Motivation:
To obtain the closed-form expression of the marginal likelihood in Bayesian networks two probabilistic models with their respective conjugate prior distributions have been employed in the past: the multinomial distribution with the Dirichlet prior, leading to the so-called BDe model (Cooper and Herskovits, 1992), and the linear Gaussian distribution with the normal-Wishart prior, leading to the BGe model (Geiger and Heckerman, 1994). These approaches are restricted in that they either require the data to be discretized (BDe) or can only capture linear regulatory relationships (BGe). A non-linear non-discretized model based on heteroscedastic

regression has been proposed by Imoto *et al.* (2003). However, this approach no longer allows the marginal likelihood to be obtained in closed-form and requires a restrictive approximation (the Laplace approximation for integrals) to be used. Another nonlinear model based on node-specific Gaussian mixture models has been proposed in Ko *et al.* (2007). Again, the marginal likelihood is intractable. The authors resort to the Bayesian information criterion (BIC) of Schwarz (1978) for model selection, which is only a good approximation to the marginal likelihood in the limit of very large data sets, and employ the Expectation Maximization (EM) algorithm (Dempster *et al.*, 1977) for inference. The proposed BGM model is a non-linear generalization of the static (see Subsection 2.3) and dynamic (see Subsection 2.5) Gaussian BGe model for Bayesian networks, and can be motivated by the fact that any probability distribution can, in principle, be approximated arbitrarily closely by a mixture model. In the BGM model data points are assigned to different compartments (subsets of the data) with the allocation sampler of Nobile and Fearnside (2007). Model parameters and their distributions are allowed to differ between compartments, while information is shared among the compartments via a common network structure. Conditional on the common network structure, each compartment (data subset) is modeled separately and independently with the Gaussian BGe model for Bayesian networks.

The mathematical model:
As explained in Subsections 2.2-2.4 it is assumed that there is a N-by-m data matrix, $\mathcal{D}$, and that the columns are m observations of the random vector $\vec{X} = (X_1, \ldots, X_N)^T$. The BGM Bayesian network model can be applied to both: *static* data, when each observation (column of $\mathcal{D}$) is an independent realization of $\vec{X}$, and *dynamic* data, when the observations are time dependent with a homogeneous Markovian dependence structure. In this thesis the focus will be on first-order dynamic Bayesian networks with $\tau = 1$, and it is referred to the original publication (Grzegorczyk *et al.*, 2008) for static Bayesian networks.
In addition to those notations that were introduced for standard dynamic Bayesian networks in Subsection 2.4, let $\vec{\mathcal{V}}$ be an allocation vector, whose entries are latent variables. The vector, $\vec{\mathcal{V}}$, gives an allocation of the m observations to $\mathcal{K}$ mixture components: For $j = 1, \ldots, m$: $\vec{\mathcal{V}}(j) = k$ means that the j-th observation is allocated to the k-th component ($1 \leq k \leq \mathcal{K}$). $\mathcal{D}^{(\vec{\mathcal{V}},k)}$ denotes the data subset consisting of all observations allocated to the k-th component by $\vec{\mathcal{V}}$ ($1 \leq k \leq \mathcal{K}$). The joint posterior probability of a graph, $\mathcal{G}$, an allocation vector, $\vec{\mathcal{V}}$, and $\mathcal{K}$ mixture components is assumed to factorize as follows:

$$P(\mathcal{G}, \mathcal{K}, \vec{\mathcal{V}}|\mathcal{D}) = \frac{P(\mathcal{G}, \mathcal{K}, \vec{\mathcal{V}}, \mathcal{D})}{P(\mathcal{D})} \propto P(\mathcal{G}, \mathcal{K}, \vec{\mathcal{V}}, \mathcal{D}) = P(\mathcal{K})P(\vec{\mathcal{V}}|\mathcal{K})P(\mathcal{G})P(\mathcal{D}|\mathcal{G}, \mathcal{K}, \vec{\mathcal{V}}) \tag{58}$$

where

$$P(\mathcal{D}|\mathcal{G}, \mathcal{K}, \vec{\mathcal{V}}) = \prod_{k=1}^{\mathcal{K}} P(\mathcal{D}^{(\vec{\mathcal{V}},k)}|\mathcal{G}) \tag{59}$$

From Eq. (59) it can be seen that the marginal likelihood terms, $P(\mathcal{D}^{(\vec{\mathcal{V}},k)}|\mathcal{G})$, for the data subsets, $\mathcal{D}^{(\vec{\mathcal{V}},k)}$, given the graph, $\mathcal{G} = \{\pi_1, \ldots, \pi_n\}$, can be computed independently with one of the standard Bayesian network model; e.g. the Gaussian BGe model. That is, if no observation is allocated to the k-th component, symbolically $\mathcal{D}^{(\vec{\mathcal{V}},k)} = \emptyset$, then $P(\mathcal{D}^{(\vec{\mathcal{V}},k)}|\mathcal{G})$ is equal to 1, and for non-empty components Eq. (21)

can be employed:

$$P(\mathcal{D}^{(\vec{\mathcal{V}},k)}|\mathcal{G}) = \int P(\mathcal{D}^{(\vec{\mathcal{V}},k)}|\mathcal{G},\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathcal{G})d\boldsymbol{\theta} = \prod_{n=1}^{N}\Psi(\mathcal{D}_n^{(\vec{\mathcal{V}},k),\pi_n}) \quad (60)$$

$$\Psi(\mathcal{D}_n^{(\vec{\mathcal{V}},k),\pi_n}) = \int \prod_{t:\vec{\mathcal{V}}(t)=k} P\Big(X_n = \mathcal{D}_{n,t}|\pi_n = \mathcal{D}_{\pi_n,t-1},\boldsymbol{\theta}_n\Big)P(\boldsymbol{\theta}_n|\pi_n)d\boldsymbol{\theta}_n \quad (61)$$

which yields a closed-form expression for $P(\mathcal{D}^{(\vec{\mathcal{V}},k)}|\mathcal{G})$ for each component k if a standard Bayesian network model (BDe or BGe) is employed. Since there is a closed-form solution for each factor, the marginal likelihood of the proposed BGM model, $P(\mathcal{D}|\mathcal{G},\mathcal{K},\vec{\mathcal{V}})$, in Eq. (59) can also be computed in closed-form.
For the graph prior, $P(\mathcal{G})$, in Eq. (58) a standard prior, e.g. a uniform distribution over networks, can be chosen. For the prior on $\mathcal{K}$, $P(\mathcal{K})$, a truncated Poisson distribution with parameter $\lambda = 1$ restricted to $1 \leq \mathcal{K} \leq \mathcal{K}^{MAX}$ can be selected, and it is further assumed that the probability distribution of the allocation vector, $\vec{\mathcal{V}}$, conditional on $\mathcal{K}$ is given by:

$$P(\vec{\mathcal{V}}|\mathcal{K},\vec{p}) = \prod_{k=1}^{\mathcal{K}} p_k^{n_k} \quad (62)$$

where $\vec{p} = (p_1,\ldots,p_{\mathcal{K}})^T$ with $\sum_{k=1}^{\mathcal{K}} p_k = 1$ are the non-negative mixture weights, and n_k is the number of observations allocated to the k-th mixture component by $\vec{\mathcal{V}}$, symbolically $n_k := |\mathcal{D}^{(\vec{\mathcal{V}},k)}|$. The prior on the mixture weights, $\vec{p} = (p_1,\ldots,p_{\mathcal{K}})^T$, is chosen to be a Dirichlet distribution, $P(\vec{p}) = Dir(\alpha_1,\ldots,\alpha_{\mathcal{K}})$, with hyperparameters $\vec{\alpha} = (\alpha_1,\ldots\alpha_{\mathcal{K}})^T$. This prior is conjugate, and the marginal probability of $\vec{\mathcal{V}}$ conditional on $\mathcal{K}$ is thus given by:

$$P(\vec{\mathcal{V}}|\mathcal{K}) = \int P(\vec{\mathcal{V}}|\mathcal{K},\vec{p})P(\vec{p})d\vec{p} = Dir(n_1+\alpha_1,\ldots,n_{\mathcal{K}}+\alpha_{\mathcal{K}}) \quad (63)$$

This choice of prior distributions for $P(\mathcal{G})$, $P(\mathcal{K})$, and $P(\vec{\mathcal{V}}|\mathcal{K})$ combined with the closed form solutions for $P(\mathcal{D}^{(\vec{\mathcal{V}},k)}|\mathcal{G})$ $(1 \leq k \leq \mathcal{K})$ from Eq. (60) ensures that the joint probability distribution, $P(\mathcal{G},\mathcal{K},\vec{\mathcal{V}},\mathcal{D})$, which is proportional to the posterior distribution (see Eq. (58)), can be computed in closed-form.

Inference:
The new Gaussian mixture allocation MCMC sampling scheme generates a sample $\{\mathcal{G}^i,\mathcal{K}^i,\vec{\mathcal{V}}^i\}_{i=1,\ldots,T}$ from the joint posterior distribution, $P(\mathcal{G},\mathcal{K},\vec{\mathcal{V}}|\mathcal{D})$, given in Eq. (58) and comprises six different types of moves in the configuration space, $\Big\{[\mathcal{G},\mathcal{K},\vec{\mathcal{V}}]\Big\}$. The first move type is a classical structure MCMC single-edge-operation on the graph, $\mathcal{G}$, while the number of components, $\mathcal{K}$, and the allocation vector, $\vec{\mathcal{V}}$, are left unchanged. According to Eq. (31) a new candidate graph, $\tilde{\mathcal{G}}$, is randomly drawn out of the set of neighbor graphs, $\mathcal{N}(\mathcal{G})$, that can be reached from the current graph, $\mathcal{G}$, by one single edge deletion, edge addition or edge reversal. The new state, $[\tilde{\mathcal{G}},\mathcal{K},\vec{\mathcal{V}}]$, is accepted or rejected according to Eq. (32) where the likelihood terms, $P(\mathcal{D}|\mathcal{G})$, in Eq. (32) have to be replaced by the corresponding $P(\mathcal{D}|\mathcal{G},\mathcal{K},\vec{\mathcal{V}})$ terms of the BGM model, given in Eq. (59). The acceptance probability for a move from $[\mathcal{G},\mathcal{K},\vec{\mathcal{V}}]$ to $[\tilde{\mathcal{G}},\mathcal{K},\vec{\mathcal{V}}]$ is then given by $A = \min\{1,R\}$ where

$$R = \frac{P(\tilde{\mathcal{G}},\mathcal{K},\vec{\mathcal{V}}|\mathcal{D})}{P(\mathcal{G},\mathcal{K},\vec{\mathcal{V}}|\mathcal{D})}\cdot\frac{|\mathcal{N}(\mathcal{G})|}{|\mathcal{N}(\tilde{\mathcal{G}})|} = \frac{P(\tilde{\mathcal{G}})\prod_{k=1}^{\mathcal{K}}P(\mathcal{D}^{(\vec{\mathcal{V}},k)}|\tilde{\mathcal{G}})}{P(\mathcal{G})\prod_{k=1}^{\mathcal{K}}P(\mathcal{D}^{(\vec{\mathcal{V}},k)}|\mathcal{G})}\cdot\frac{|\mathcal{N}(\mathcal{G})|}{|\mathcal{N}(\tilde{\mathcal{G}})|} \quad (64)$$

The five other move types are adapted from Nobile and Fearnside (2007) and operate on $\vec{\mathcal{V}}$ or on both $\mathcal{K}$ and $\vec{\mathcal{V}}$. The resulting algorithm is effectively a RJMCMC sampling scheme in the discrete space of network structures and latent allocation vectors, where the Jacobian in the acceptance criterion is always 1, and hence can be omitted. If there are $\mathcal{K} > 2$ mixture components, then moves of the type M1 and M2 can be used to re-allocate some observations from one component k_1 to another one k_2. That is, a new allocation vector $\vec{\mathcal{V}}^*$ is proposed while $\mathcal{G}$ and $\mathcal{K}$ are left unchanged. The Ejection move type proposes to increment the number of mixture components by 1 and simultaneously tries to re-allocate some observations to fill the new component. To this end, it randomly selects a mixture component and tries to re-allocate some of its observations to the newly proposed component $\mathcal{K}+1$ while the graph, $\mathcal{G}$, is left unchanged. The Absorption move is complementary to the Ejection move and proposes to decrement the number of mixture components by 1. To this end, it randomly selects two mixture components and deletes one of them after having re-allocated all of its observations to the other component. The acceptance probabilities for M1, M2, Ejection, and Absorption moves from $[\mathcal{G}, \mathcal{K}, \vec{\mathcal{V}}]$ to $[\mathcal{G}, \mathcal{K}^*, \vec{\mathcal{V}}^*]$ are of the same functional form:

$$A = \left\{ 1, \frac{P(\mathcal{D}|\mathcal{G}, \mathcal{K}^*, \vec{\mathcal{V}}^*)}{P(\mathcal{D}|\mathcal{G}, \mathcal{K}, \vec{\mathcal{V}})} \cdot \frac{P(\vec{\mathcal{V}}^*|\mathcal{K}^*)}{P(\vec{\mathcal{V}}|\mathcal{K})} \cdot \frac{P(\mathcal{K}^*)}{P(\mathcal{K})} \cdot \frac{Q(\vec{\mathcal{V}}^*|\vec{\mathcal{V}})}{Q(\vec{\mathcal{V}}|\vec{\mathcal{V}}^*)} \right\} \tag{65}$$

where the likelihood terms have been specified in Eq. (59), the proposal probabilities $Q(.|.)$ depend on the move type (M1, M2, Ejection or Absorption), and $\mathcal{K}^* = \mathcal{K}$ for M1 and M2 moves, $\mathcal{K}^* = \mathcal{K}+1$ for Ejection moves, and $\mathcal{K}^* = \mathcal{K}-1$ for Absorption moves.

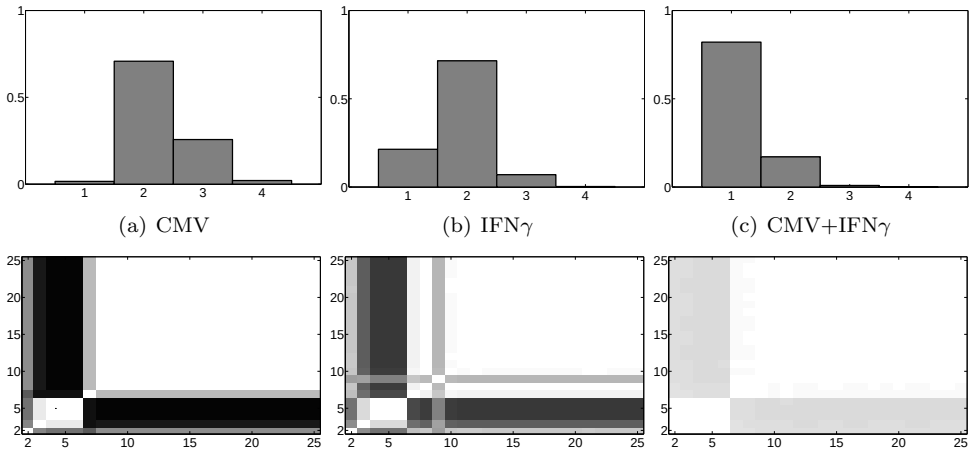
E.g. the M1 move works as follows: If there is one mixture component only, symbolically $\mathcal{K} = 1$, a different type of move has to be selected. Otherwise randomly select two mixture components, e.g. the i-th and the j-th among the $\mathcal{K}$ available. Draw a random number p from a Beta distribution whose parameters are equal to the corresponding hyperparameters α_i and α_j of the Dirichlet prior, $P(\vec{p}) = Dir(\alpha_1, \ldots, \alpha_{\mathcal{K}})$, on the mixture weights, $\vec{p}$. Re-allocating each observation currently belonging either to the i-th or to the j-th component with probability p to component i and with the complementary probability $1-p$ to component j gives the new allocation vector, $\vec{\mathcal{V}}^*$. Nobile and Fearnside (2007) show that for M1 proposal probabilities holds:

$$\frac{Q(\vec{\mathcal{V}}^*|\vec{\mathcal{V}})}{Q(\vec{\mathcal{V}}|\vec{\mathcal{V}}^*)} = \left\{ \frac{P(\vec{\mathcal{V}}^*|\mathcal{K}^*)}{P(\vec{\mathcal{V}}|\mathcal{K})} \right\}^{-1} \tag{66}$$

Therefore, the corresponding terms in Eq. (65) cancel out. Furthermore, as the M1 move does not change the number of components, $\mathcal{K}^* = \mathcal{K}$, all that remains to compute is the likelihood ratio: $\frac{P(\mathcal{D}|\mathcal{G},\mathcal{K},\vec{\mathcal{V}}^*)}{P(\mathcal{D}|\mathcal{G},\mathcal{K},\vec{\mathcal{V}})}$. For M1 moves all except the i-th and the j-th factor cancel out from the ratio when the likelihoods are factorized according to Eq. (59). Hence the acceptance probability for an M1 move from $[\mathcal{G}, \vec{\mathcal{V}}, \mathcal{K}]$ to $[\mathcal{G}, \vec{\mathcal{V}}^*, \mathcal{K}]$ is given by:

$$A = \min \left\{ 1, \frac{P(\mathcal{D}^{(\vec{\mathcal{V}}^*,i)}|\mathcal{G})}{P(\mathcal{D}^{(\vec{\mathcal{V}},i)}|\mathcal{G})} \frac{P(\mathcal{D}^{(\vec{\mathcal{V}}^*,j)}|\mathcal{G})}{P(\mathcal{D}^{(\vec{\mathcal{V}},j)}|\mathcal{G})} \right\} \tag{67}$$

and the four probabilities can be computed with Eq. (60). The details for the other four moves can be found in the original publication (Grzegorczyk *et al.*, 2008). The sixth move type uses Gibbs sampling to re-allocate one single observation by sampling its new allocation from the corresponding univariate conditional distribution while leaving $\mathcal{G}$, $\mathcal{K}$ and the other components of $\vec{\mathcal{V}}$ unchanged; see Grzegorczyk *et al.* (2008) for details.



(a) CMV (b) IFNγ (c) CMV+IFNγ

(d) CMV (e) IFNγ (f) CMV+IFNγ

Figure 5: **MCMC inference results obtained with the BGM model for the macrophage data.** *Top row:* Histograms of the numbers of BGM components for the three macrophage gene expression time series. For each experimental condition the posterior probability (vertical axis) of the number of components $\mathcal{K}$ (horizontal axis) have been estimated from the MCMC trajectories. *Bottom row:* Graphical representation of the temporal connectivity structure for the macrophage gene expression data. The figure shows heatmap representations that indicate the estimated posterior probability of two time points being assigned to the same state (component). The probabilities are represented by a grey shading, where white corresponds to a probability of 1, and black corresponds to a probability of 0. The numbers on the axes represent the time points of the time course experiment. The analysis was repeated for all three experimental conditions CMV, IFNγ and CMV+IFNγ, as explained in the text.

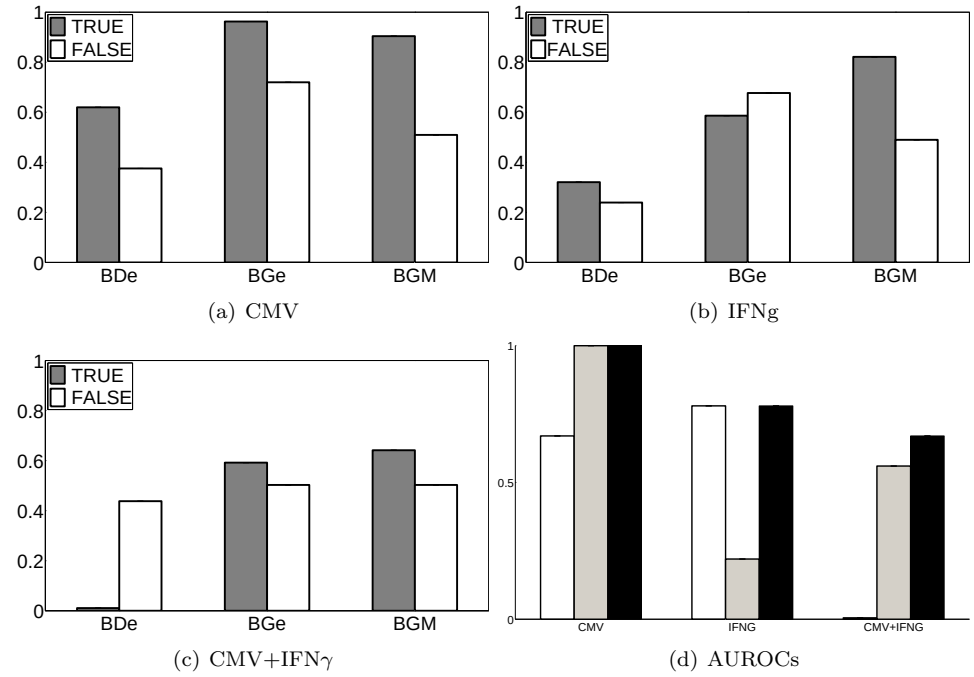
Selected application(s):

Interferons (IFNs) play a pivotal role in the innate and adaptive mammalian immune response against infection, and central research efforts therefore aim to elucidate their regulatory interactions (Honda *et al.*, 2006). In the study presented here, the BGM Bayesian network model was applied to gene expression time series from bone marrow derived macrophages, which were sampled at 25 $\times$ 30 minute time intervals. The macrophages were subjected to three external conditions: (1) infection with Cytomegalovirus (CMV), (2) treatment with Interferon Gamma (IFNγ), and (3) infection with Cytomegalovirus after pre-treatment with IFNγ (CMV+IFNγ). The focus was on time series of the Interferon regulatory factors Irf1, Irf2 and Irf3, since a gold standard network for the interactions between these factors can be derived from the literature (Darnell *et al.* (1994) and Raza *et al.* (2008)): Irf2 $\leftrightarrow$ Irf1 $\leftarrow$ Irf3.[26]

For the macrophage gene expression time series, the MCMC sampling scheme infers $\mathcal{K} = 2$ components for the conditions CMV and IFNγ, while for the third condition CMV+IFNγ most of the sampled states consist of $\mathcal{K} = 1$ component only, as shown in the top row of Figure 5.[27] The fraction of sampled states for which two observations i and j are allocated to the same component k ($1 \leq k \leq \mathcal{K}$) can be used as a connectivity measure $C(i, j)$. The bottom row of Figure 5 displays the

[26]The Interferon regulatory factors are the key regulators in the response of the macrophage cell to pathogens. They mediate the cellular signalling that leads to a transcriptional response to the initial binding events on the surface of the cell.

[27]Simulation details, such as hyperparameter settings and MCMC simulation lengths can be found in the original publication (Grzegorczyk *et al.*, 2008).



(a) CMV (b) IFNg

(c) CMV+IFNγ (d) AUROCs

Figure 6: **Reconstructing the regulatory network of the three Interferon regulatory factors Irf1, Irf2, and Irf3.** Panels (a)-(c): Mean posterior probabilities (vertical axis) of true and false edges in the Irf regulatory network, inferred with BDe, BGe, and BGM (horizontal axis) from the three macrophage gene expression time series: (a) CMV, (b) IFN$_\gamma$, and (c) CMV+IFN$_\gamma$. According to the biological literature the true edges are: $Irf1 \rightarrow Irf2$, $Irf2 \rightarrow Irf1$, and $Irf3 \rightarrow Irf1$, while the edges $Irf1 \rightarrow Irf3$, $Irf2 \rightarrow Irf3$, and $Irf3 \rightarrow Irf2$ are spurious. In panel (d) an AUROC histogram plot is given. For each of the three conditions the histogram shows bars of the BDe (white), BGe (grey) and BGM (black) AUROC values. It can be seen that BGM is never inferior to BDe or BGe in terms of AUROC scores, but BGM outperforms (i) BDe for the two conditions CMV and CMV+IFN$_\gamma$ and (ii) BGe for the two conditions IFN$_\gamma$ and CMV+IFN$_\gamma$.

resulting connectivity matrices graphically as heat matrices. From the heat matrices the same systematic trend can be observed for the three conditions. The first part (observations no. 2-6) and the last part of the three time series (observations no. 8-25) are allocated to different components. For condition CMV (IFNγ) the allocation of observation no. 7 (no. 9) is not fixed, that is, the allocation changes during the MCMC simulation. For CMV+IFNγ, whose number of components peaks at $\mathcal{K} = 1$ (see panel (c) in Figure 5), the separation between the two parts is less pronounced, though consistent with the other results. That is, there is a two-stage process, which reflects a state change in the host macrophage brought about by infection (CMV) or immune activation (IFNγ). Interestingly, this state change is less pronounced in the combined condition CMV+IFNγ. This observation is consistent with the known biological responses of macrophages to simultaneous infection by virus (CMV) and immune (IFNγ) activation. It suggests that upon dual challenge with both an infection and immune activation (CMV+IFNγ) signalling leads to a pronounced singular response. This is in agreement with observations of cooperation between viral and immune signalling in effective vigorous anti-viral state within the host macrophage, as discussed in Benedict *et al.* (2001).

To understand whether the BGM model also yields a better network reconstruction accuracy, the mean posterior probabilities of the true and false edges of the BGM model can be compared with those from the two standard Bayesian network models BGe and BDe; see Figure 6. For the IFNγ condition (panel (b)) BGM performs substantially better than BGe and BDe. For the other two conditions the difference

between the posterior means for the true and the false edges is also best for BGM, but the difference is less pronounced (BDe: 0.24, BGe: 0.24, BGM: 0.39 (CMV) and BDe: -0.42 , BGe: 0.09, BGM: 0.14 (CMV+IFNγ)).

3.3.3 Modeling non-homogeneous Bayesian networks with a multiple changepoint model

- **ORIGINAL PUBLICATION no. 1: Grzegorczyk, M.**, Husmeier, D., and Rahnenführer, J. (2011) Modelling non-stationary gene regulatory processes with the BGM model. *Computational Statistics*, **26(2)**, 199-218.
- **ORIGINAL PUBLICATION no. 2: Grzegorczyk, M.**, Husmeier, D., and Rahnenführer, J. (2010) Modelling non-stationary gene regulatory processes. *Advances in Bioinformatics*, **Volume 2010**, online article, 21 pages.

Summary:
In these two articles a modification of the Bayesian Gaussian Mixture (BGM) Bayesian network model from Subsection 3.3.2 was proposed, and the novel model was referred to as the dynamic variant BGM_D of the BGM model.[28] The novel BGM_D model is more suitable for modeling dynamic gene regulatory networks and gene expression time series, since it replaces the free mixture allocation model of the original BGM model from Subsection 3.3.2 by a multiple changepoint process. The assignment of data points to components is then done by a multiple changepoint process (Green (1995) and Suchard *et al.* (2003)), which divides the time series into segments, and therefore – different from the free allocation BGM model – the intuitive prior notion, that adjacent time points are likely to be governed by similar distributions, is taken into account.
The focus of the first publication (Grzegorczyk *et al.*, 2011) was on synthetic network data and the paper also contains a cross-method comparison with two other methods, namely a state-space model (Beal *et al.*, 2005) and a sparse linear regression approach (Rogers and Girolami, 2005). The second original publication (Grzegorczyk *et al.*, 2010) extends the first publication in three important aspects: First, the two models BGM and BGM_D were cross-compared on several gene expression time series from macrophages and *Arabidopsis thaliana*. Second, both models BGM and BGM_D were also implemented with the discrete BDe model to compare their performances on inferring the morphogenic stages of muscle development in *Drosophila melanogaster* from binary gene expression time series. Third, in addition to the application to three real biological systems, the second publication (Grzegorczyk *et al.*, 2010) also contains several theoretical comparisons between the two models BGM and BGM_D. These theoretical studies include a comparison of the a priori imposed temporal connectivity structures and a comparison of the prior probability ratios between the non-homogeneous and the homogeneous states of the two models. The theoretical results were found to be consistent with the empirical findings for real gene expression data, and thus give a deeper insight into the intrinsic difference between the free allocation BGM model and its changepoint variant BGM_D.

Motivation:
The Bayesian Gaussian Mixture (BGM) Bayesian network model from Subsection 3.3.2 provides the proper approximation of a non-linear regulation process by a

[28] Although the *Computational Statistics* paper (Grzegorczyk *et al.*, 2011) was printed later than the *Advances in Bioinformatics* online article (Grzegorczyk *et al.*, 2010), the *Computational Statistics* paper is theoretically orientated and can be seen as a precursor of the *Advances in Bioinformatics* paper, whose main focus is on biological applications. The principle idea of the BGM_D model was first presented at the WCSB workshop 2009 in Aarhus, Denmark and appeared in the conference proceedings (Grzegorczyk and Husmeier, 2009b).

piecewise linear process, since the assignment of observations to mixture components is done in the domain of the variables by a free mixture allocation of data points. Replacing the free mixture allocation model by a multiple changepoint process (e.g. see Green (1995) and Suchard *et al.* (2003)) yields the modified BGM$_D$ model, which in the first instance relaxes only the homogeneity assumption intrinsic to dynamic Bayesian network methodology. The novel BGM$_D$ model provides a proper approximation of a non-linear regulation process only under the assumption that the temporal processes are sufficiently smooth, since the assignment of observations to mixture components is now done in the temporal domain rather than the domain of variables. However, replacing the free allocation mixture model by a multiple changepoint process, reduces the complexity of the allocation space substantially and incorporates as prior knowledge that adjacent time points are likely to be assigned to the same component. As for the original BGM model from Subsection 3.3.2, the practical inference follows the Bayesian paradigm and samples the network, the number of changepoints and the changepoint locations from the posterior distribution with Markov Chain Monte Carlo (MCMC).

The mathematical model:
In the first instance, the novel BGM$_D$ model is similar to the original BGM model, described in Subsection 3.3.2. Given the N-by-m data matrix, $\mathcal{D}$, of temporal observations, let $\mathcal{D}_{n,t}$ and $\mathcal{D}_{\pi_n,t}$ denote the realizations of X_n and π_n at time point t. An allocation vector, $\vec{\mathcal{V}}$, of length m gives an allocation of the time points to $\mathcal{K}$ classes. In the BGM$_D$ model the classes correspond to temporal segments rather than mixture components. $\vec{\mathcal{V}}(j) = k$ means that the j-th observation is allocated to time segment k. The joint posterior probability of a graph, $\mathcal{G}$, $\mathcal{K}$ segments, and the allocation vector, $\vec{\mathcal{V}}$, factorizes exactly as in the BGM model:

$$P(\mathcal{G}, \mathcal{K}, \vec{\mathcal{V}}|\mathcal{D}) = \frac{P(\mathcal{G}, \mathcal{K}, \vec{\mathcal{V}}, \mathcal{D})}{P(\mathcal{D})} \propto P(\mathcal{G}, \mathcal{K}, \vec{\mathcal{V}}, \mathcal{D}) = P(\mathcal{K})P(\vec{\mathcal{V}}|\mathcal{K})P(\mathcal{G})P(\mathcal{D}|\mathcal{G}, \mathcal{K}, \vec{\mathcal{V}}) \tag{68}$$

where $P(\mathcal{D}|\mathcal{G}, \vec{\mathcal{V}}, \mathcal{K})$ was defined in Eq. (59) and can be computed in closed-form with Eqns. (60-61), as explained in Subsection 3.3.2. The posterior distribution for the BGM$_D$ model is then given by:

$$P(\mathcal{G}, \vec{\mathcal{V}}, \mathcal{K}|\mathcal{D}) \propto P(\mathcal{K})P(\vec{\mathcal{V}}|\mathcal{K})P(\mathcal{G}) \prod_{k=1}^{\mathcal{K}} \prod_{n=1}^{N} \Psi(\mathcal{D}_n^{(\vec{\mathcal{V}},k),\pi_n}) \tag{69}$$

where the local scores, $\Psi(.)$, were specified in Eq. (61). As for the original BGM model the prior on $\mathcal{K}$, $P(\mathcal{K})$, is assumed to be a truncated Poisson distribution with parameter $\lambda = 1$ restricted to $1 \leq \mathcal{K} \leq \mathcal{K}^{MAX}$. But different from the original BGM model the allocation vector, $\vec{\mathcal{V}}$, of the novel BGM$_D$ model is *not* implemented by a free mixture allocation model. In the BGM$_D$ model $\mathcal{K}$ segments are identified with $\mathcal{K} - 1$ changepoints: $b_1, \ldots, b_{\mathcal{K}-1}$ on the continuous interval $[2, m]$, where m is the number of observed time points, and the allocation is done by a multiple changepoint process (Green, 1995). That is, the observation at time point t is assigned to the k-th component, symbolically $\vec{\mathcal{V}}(t) = k$, if $b_{k-1} \leq t < b_k$, where for notational convenience the pseudo-changepoints $b_0 = 2$ and $b_{\mathcal{K}} = m$ have been introduced. That is, the set of changepoints, $\mathcal{B} = \{b_1, \ldots, b_{\mathcal{K}-1}\}$, consisting of $\mathcal{K} - 1$ changepoints, specifies the allocation vector, $\vec{\mathcal{V}} = \mathcal{V}(\mathcal{B})$, where the function, $\mathcal{V}(.)$, maps the changepoints onto the allocation vectors, as described above.
Following (Green, 1995) it is assumed that the changepoints in $\mathcal{B}$ are distributed as the even-numbered order statistics of $\mathcal{L} := 2(\mathcal{K} - 1) + 1$ points, $u_1, \ldots, u_{\mathcal{L}}$, uniformly

and independently distributed on the interval $[2, m]$. The motivation for this prior, $P(\mathcal{B}|\mathcal{K})$, instead of taking $\mathcal{K}-1$ uniformly distributed points, is to encourage *a priori* an equal spacing between the changepoints, i.e. to discourage segments (classes) that contain only a short compartment of the time series.
It is important to note that this prior, $P(\mathcal{B}|\mathcal{K})$, on $\mathcal{K}-1$ changepoint locations, $b_1, \ldots, b_{\mathcal{K}-1}$, induces a prior distribution, $P(\vec{\mathcal{V}}|\mathcal{K})$, on the allocation vector, $\vec{\mathcal{V}}$, conditional on the number of classes, $\mathcal{K}$, but there is *no* closed-form expression for $P(\vec{\mathcal{V}}|\mathcal{K})$. Therefore, it is more convenient and also appears more natural to sample indirectly from $P(\vec{\mathcal{V}}|\mathcal{K})$ by modifying the number and locations of changepoints rather than modifying the allocation vector, $\vec{\mathcal{V}}$, directly. The changepoint set, $\mathcal{B} = \{b_1, \ldots, b_{\mathcal{K}-1}\}$, specifies the number of segments, $\mathcal{K} = |\mathcal{B}| + 1$, where $|.|$ denotes the cardinality, and a unique allocation vector, $\vec{\mathcal{V}} = \mathcal{V}(\mathcal{B})$. But there is no one-to-one mapping; i.e. different changepoint sets, $\mathcal{B}$ and $\mathcal{B}^*$, can be mapped onto the same allocation vector, $\vec{\mathcal{V}} = \mathcal{V}(\mathcal{B}) = \mathcal{V}(\mathcal{B}^*)$. The MCMC sampling scheme, that will be discussed below, features moves that propose to change the set of changepoints from $\mathcal{B}$ to $\mathcal{B}^*$. And each move in the space of changepoint sets corresponds to a move in the space of the number of components and the allocation vectors: $[\mathcal{K}, \vec{\mathcal{V}}]$ to $[\mathcal{K}^*, \vec{\mathcal{V}}^*]$, where $\vec{\mathcal{V}} = \mathcal{V}(\mathcal{B})$, $\vec{\mathcal{V}}^* = \mathcal{V}(\mathcal{B}^*)$, $\mathcal{K} = |\mathcal{B}| + 1$, and $\mathcal{K}^* = |\mathcal{B}^*| + 1$. However, it should be noted that the BGM_D model could also be formulated in terms of changepoint sets rather than allocation vectors.

Inference:
Like the MCMC sampling scheme for the original BGM model, described in Subsection 3.3.2, inference for the BGM_D model can be done by combining the structure MCMC sampling algorithm from Subsection 2.6 with suitable moves that change the changepoint set, $\mathcal{B} = \{b_1, \ldots, b_{\mathcal{K}-1}\}$, where $\mathcal{K} = |\mathcal{B}| + 1$ is the number of (possibly empty) segments. For the latter types of moves changepoint birth, death and re-allocation moves (Green, 1995) appear to be most appropriate. In each MCMC step either a single-edge-operation structure MCMC move on the graph, $\mathcal{G}$, is performed and the set of changepoints, $\mathcal{B}$, is left unchanged, or a move on $\mathcal{B}$ is performed while the graph, $\mathcal{G}$, is left unchanged. The acceptance probability for a move on the graph, $\mathcal{G}$, is given by Eq. (64) with $\vec{\mathcal{V}} = \mathcal{V}(\mathcal{B})$ and $\mathcal{K} = |\mathcal{B}| + 1$.
The changepoint birth move adds one new changepoint to $\mathcal{B} = \{b_1, \ldots, b_{\mathcal{K}-1}\}$, and the new changepoint set, $\mathcal{B}^*$, consists of $\mathcal{K}$ changepoints. The changepoint death move removes one changepoint from $\mathcal{B}$ and the new changepoint set, $\mathcal{B}^*$, possesses $\mathcal{K}^* = \mathcal{K}-2$ changepoints. For birth and death moves the new allocation vector is given by: $\vec{\mathcal{V}}^* = \mathcal{V}(\mathcal{B}^*)$. The changepoint reallocation move substitutes one single changepoint in $\mathcal{B}$ for a new one. This gives a new changepoint set $\mathcal{B}^*$, with an unchanged cardinality, $\mathcal{K}-1$, and the novel allocation vector is given by: $\vec{\mathcal{V}}^* = \mathcal{V}(\mathcal{B}^*)$.[29] If a changepoint move is performed, the move types are selected with probabilities that depend on the current number of segments $\mathcal{K}$: $p_{b,\mathcal{K}}$ (for a birth), $p_{d,\mathcal{K}}$ (for a death), and $p_{r,\mathcal{K}}$ (for a re-allocation) move, where $p_{b,\mathcal{K}} + p_{d,\mathcal{K}} + p_{r,\mathcal{K}} = 1$ ($\mathcal{K} = 1, 2, \ldots$). These

[29] Note that the three changepoint moves ensure that the changepoint sets $\mathcal{B}$ and $\mathcal{B}^*$ are different, but the allocation vectors $\vec{\mathcal{V}} = \mathcal{V}(\mathcal{B})$ and $\vec{\mathcal{V}}^* = \mathcal{V}(\mathcal{B}^*)$ may be identical. The sampling scheme is defined in the space $[\mathcal{G}, \mathcal{B}]$ and under fairly mild regularity conditions (ergodicity), it converges to the posterior distribution $P(\mathcal{G}, \mathcal{B}|\mathcal{D})$ if the equation of detailed balance is fulfilled (Green, 1995). The condition of detailed balance implies that there is a unique complementary move for each move, and that the acceptance probability depends on the proposal probability of this complementary move. The moves presented here are designed such that there is a unique complementary death move for each birth move and vice-versa, and each re-allocation move can be reversed by one single (complementary) re-allocation move. Obviously, a Markov chain converging to the posterior distribution, $P(\mathcal{G}, \mathcal{B}|\mathcal{D})$, also converges to $P(\mathcal{G}, \vec{\mathcal{V}}, \mathcal{K}|\mathcal{D})$ in the space $[\mathcal{G}, \vec{\mathcal{V}}, \mathcal{K}]$, where $\vec{\mathcal{V}} = \mathcal{V}(\mathcal{B})$ and $\mathcal{K} = |\mathcal{B}| + 1$.

probabilities can be chosen as follows (Green, 1995):

$$p_{b,\mathcal{K}} = c \cdot \min\left\{1, \frac{P(\mathcal{K}+1)}{P(\mathcal{K})}\right\},\ p_{d,\mathcal{K}} = c \cdot \min\left\{1, \frac{P(\mathcal{K}-1)}{P(\mathcal{K})}\right\}, \tag{70}$$

where c is a constant that can be chosen as large as possible subject to the constraint $p_{b,\mathcal{K}} + p_{d,\mathcal{K}} \leq 0.9$ for $\mathcal{K} = 1, 2, \ldots$, and $p_{r,\mathcal{K}} = 1 - p_{b,\mathcal{K}} - p_{d,\mathcal{K}}$.
The acceptance probability for a move from $[\mathcal{G}, \mathcal{B}]$ to $[\mathcal{G}, \mathcal{B}^*]$ or $[\mathcal{G}, \mathcal{K}, \vec{\mathcal{V}}]$ to $[\mathcal{G}, \mathcal{K}^*, \vec{\mathcal{V}}^*]$, respectively, is then of the following functional form:

$$A = \min\left\{1, \frac{P(\mathcal{D}|\mathcal{G}, \vec{\mathcal{V}}^*, \mathcal{K}^*)}{P(\mathcal{D}|\mathcal{G}, \vec{\mathcal{V}}, \mathcal{K})} \times I \times P\right\} \tag{71}$$

where $\vec{\mathcal{V}} = \mathcal{V}(\mathcal{B})$, $\vec{\mathcal{V}}^* = \mathcal{V}(\mathcal{B}^*)$, $\mathcal{K} = |\mathcal{B}| + 1$, $\mathcal{K}^* = |\mathcal{B}^*| + 1$, I is the inverse proposal probability ratio, and

$$P = \frac{P(\vec{\mathcal{V}}^*|\mathcal{K}^*)P(\mathcal{K}^*)}{P(\vec{\mathcal{V}}|\mathcal{K})P(\mathcal{K})} \tag{72}$$

is the prior probability ratio. The exact form of the factors I and P depends on the move type.

(i) For a *changepoint re-allocation move*, one of the existing changepoints $b_j \in \mathcal{B} = \{b_1, \ldots, b_{\mathcal{K}-1}\}$ is randomly selected and removed. Subsequently, the replacement changepoint $b_j^\dagger$ is drawn from a uniform distribution on $[b_{j-1}, b_{j+1}]$ where $b_0 = 2$ and $b_{\mathcal{K}} = m$. The inverse proposal probability ratio I in Eq. (71) is then equal to 1. The prior probabilities $P(\mathcal{K}^*) = P(\mathcal{K})$ cancel out in the prior probability ratio, and the remaining prior probability ratio

$$P = \frac{P(\vec{\mathcal{V}}^*|\mathcal{K}^*)}{P(\vec{\mathcal{V}}|\mathcal{K})} \tag{73}$$

can be obtained from page 720 in Green's RJMCMC paper (Green, 1995):

$$P = \frac{(b_{j+1} - b_j^\dagger)(b_j^\dagger - b_{j-1})}{(b_{j+1} - b_j)(b_j - b_{j-1})} \tag{74}$$

If the changepoint set $\mathcal{B}$ is empty ($\mathcal{K} = 1$), the re-allocation move is rejected and the Markov chain is left unchanged.

(ii) If a *changepoint birth move* on $\mathcal{B} = \{b_1, \ldots, b_{\mathcal{K}-1}\}$ is proposed, the location of the new changepoint $b^\dagger$ is randomly drawn from a uniform distribution on the interval $[2, m]$. The proposal probability for this move is $p_{b,\mathcal{K}}/(m-2)$, where $p_{b,\mathcal{K}}$ is the $\mathcal{K}$-dependent probability of selecting a birth move. The reverse death move, which is selected with probability $p_{d,\mathcal{K}+1}$, consists in discarding randomly one of the $(\mathcal{K}-1)+1 = \mathcal{K}$ changepoints in $\mathcal{B}^*$. The inverse proposal probability ratio I in Eq. (71) is thus given by

$$I = \frac{p_{d,\mathcal{K}+1} \cdot (m-2)}{p_{b,\mathcal{K}} \cdot \mathcal{K}} \tag{75}$$

The prior probability ratio P is given by the expression at the bottom of page 720 in Green's RJMCMC paper (Green, 1995) slightly modified to allow for the fact that $\mathcal{K}$ components correspond to $\mathcal{K}-1$ changepoints in the BGM_D model:

$$P = \frac{P(\mathcal{K}+1)}{P(\mathcal{K})} \cdot \frac{2\mathcal{K}(2(\mathcal{K}+1))}{(m-2)^2} \cdot \frac{(b_{j+1} - b^\dagger)(b^\dagger - b_j)}{(b_{j+1} - b_j)} \tag{76}$$

where $b_j = \max\{b_i | b_i \leq b^\dagger\}$, $b_{j+1} = \min\{b_i | b_i \geq b^\dagger\}$, and $b^\dagger$ is the (proposed) new changepoint. The proposal probabilities $p_{b,\mathcal{K}}$ and $p_{d,\mathcal{K}+1}$ for birth and death moves have been chosen such that $p_{b,\mathcal{K}} \cdot P(\mathcal{K}) = p_{d,\mathcal{K}+1} \cdot P(\mathcal{K}+1)$ so that the ratio $p_{d,\mathcal{K}+1}/p_{b,\mathcal{K}}$ in Eq. (75) cancels out against the prior ratio $P(\mathcal{K}+1)/P(\mathcal{K})$ in Eq. (76). Thus, for birth moves the factor $I \times P$ in Eq. (71) simplifies to:

$$I \times P = \frac{2(2\mathcal{K}+1)}{(m-2)} \cdot \frac{(b_{j+1} - b^\dagger)(b^\dagger - b_j)}{(b_{j+1} - b_j)} \tag{77}$$

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

(iii) A *changepoint death move* randomly selects one of the changepoints in $\mathcal{B}$ and proposes to remove the selected changepoint $b^\dagger$. Since the changepoint death move is the reverse of the birth move, discussed above, it follows that the factor $I \times P$ in Eq. (71) is given by the inverse of Eq. (77) after having substituted $\mathcal{K}$ for $\mathcal{K} - 1$:

$$I \times P = \frac{(m-2)}{2(2\mathcal{K}-1)} \cdot \frac{(b_{j+1} - b_j)}{(b_{j+1} - b^\dagger)(b^\dagger - b_j)} \tag{78}$$

where $b_j = \max\{b_i | b_i \leq b^\dagger\}$, $b_{j+1} = \min\{b_i | b_i \geq b^\dagger\}$, and $b^\dagger$ is the changepoint whose removal is proposed. If the changepoint set $\mathcal{B}$ is empty ($\mathcal{K} = 1$), then there is no changepoint that can be removed, and the Markov chain is left unchanged.

Selected application(s):
Figure 2 in Subsection 2.4 shows a small dynamic network that consists of two nodes X_1 and X_2. The following non-linear state-space equations can be used for generating synthetic data from this network:

$$\begin{aligned} X_1(t) &= \sqrt{1-\varepsilon^2} \cdot X_1(t-1) + \varepsilon \cdot \phi_1(t) \\ X_2(t) &= \beta(t) \cdot X_1(t-1) + c \cdot \phi_2(t) \end{aligned} \tag{79}$$

where $\varepsilon \in [0,1]$, and $\phi_1(2), \phi_1(3), \ldots, \phi_2(2), \phi_2(3), \ldots$ are independently and identically (iid) standard Gaussian distributed variables. The first equation describes an autoregressive process, $\{X_1(t)\}_t$, and $\sqrt{1-\varepsilon^2} \in [0,1]$ is the (auto-)correlation between $X_1(t-1)$ and $X_1(t)$ for all t. The autocorrelation does not vary in time, and the autocorrelation can be tuned straightforwardly by varying ε. E.g. for $\varepsilon = 1$ the process $\{X_1(t)\}_t$ is a white noise process: $X_1(t) = \phi_1(t)$ for all t, and for $\varepsilon = 0$ the process $\{X_1(t)\}_t$ is a constant process $X_1(t) = X_1(t-1)$ for all t. Marginally, $X_1(t)$ is standard Gaussian distributed at each time point t. Accordingly, it can be set: $X_1(1) \sim N(0,1)$ and $X_2(1) \sim N(0,1)$.
From the second equation it can be seen that the relationship between X_1 and X_2 is implemented as a piece-wise linear function, since the regression coefficient, $\beta(t)$, is time-dependent. The noise level can be specified in terms of signal-to-noise ratios (SNRs). The standard deviation $\sigma(\beta(t)X_1(t))$ of the input signals $\beta(1)X_1(1), \beta(2)X_1(2), \ldots$ (before noise injections) can be estimated in advance by exhaustive data simulations. Having estimated $\sigma(\beta(t)X_1(t))$ by the empirical standard deviation, $\sigma(\widehat{\beta(t)X_1(t)})$, from pre-simulated data, c can be computed as follows:

$$c = \frac{\sigma(\widehat{\beta(t)X_1(t)})}{SNR} \tag{80}$$

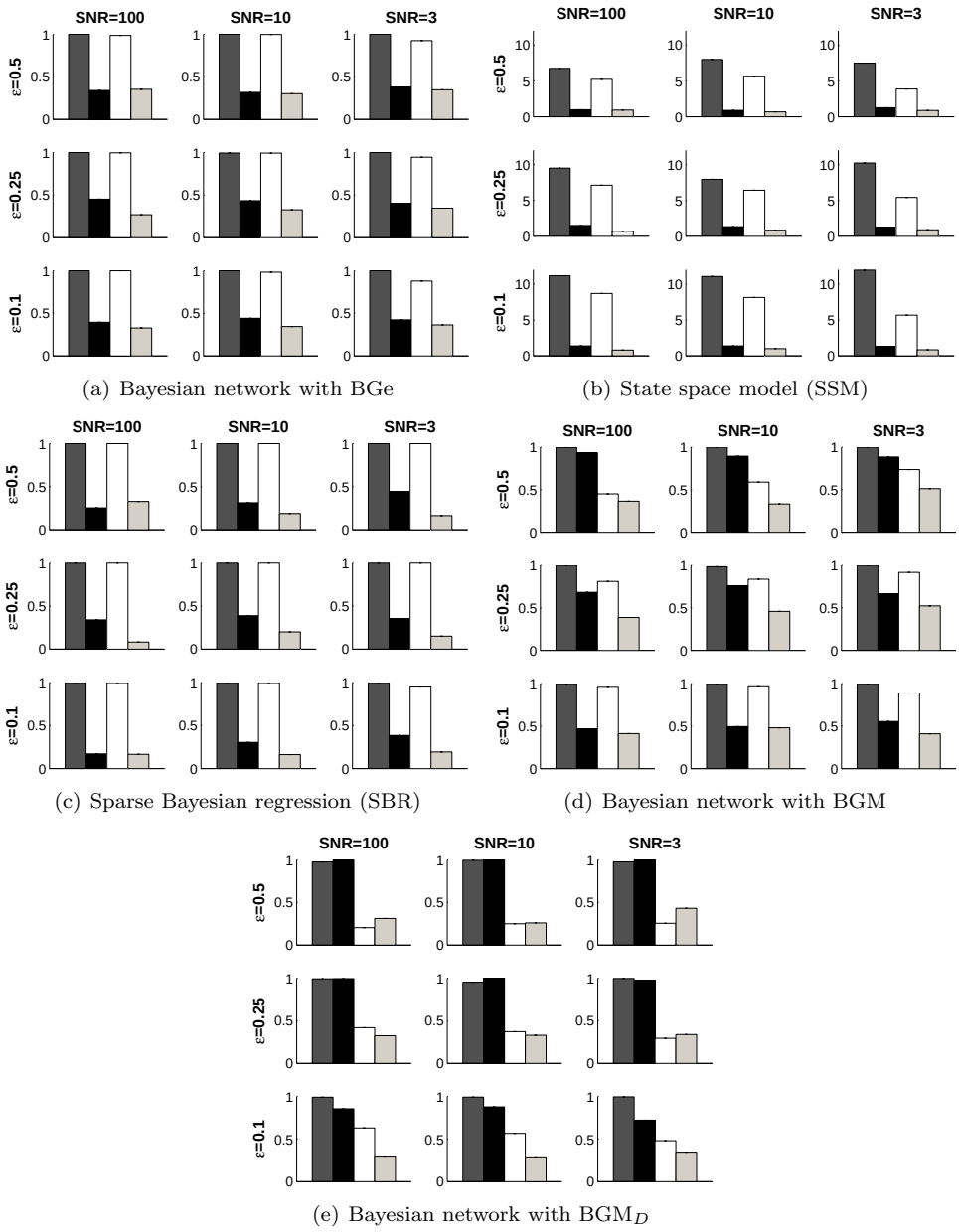
where SNR is the desired signal-to-noise ratio. For the study $m = 41$ data points were generated, and $\beta(t)$ was set to 1 for the first ($2 \leq t \leq 11$) and the

last ($32 \leq t \leq 41$) ten observations, while $\beta(t)$ was set to -1 for the 20 time points in between. In the study 9 combinations of ε ($\varepsilon = 0.5, 0.25, 0.1$) and SNR ($SNR = 100, 10, 3$) were considered, and 50 independent data sets were generated for each combination.

The mean AUROC values for assessing the reconstruction for the small network data, generated as described above, are represented as histograms in Figure 7. In this study a comparison with the sparse Bayesian regression (SBR) model of Rogers and Girolami (2005), and the Bayesian state-space model (SSM) of Beal *et al.* (2005) was included.[30] Fig. 7 shows histograms of the average marginal edge posterior probabilities of the four possible edges for the synthetic network with $N = 2$ nodes. Consistently, for all combinations of ε and SNR the five models under comparison: BGe (a), SSM (b), SBR (c), BGM (d), and BGM$_D$ (e) tend to assign the highest posterior probability to the true self-loop $X_1 \to X_1$ and the lowest posterior probability to the false edge $X_2 \to X_1$. But only the proposed BGM$_D$ model consistently suppresses the false self-feedback loop $X_2 \to X_2$ and assigns a higher edge posterior probability to the true edge $X_1 \to X_2$. The opposite behavior can be observed for the models BGe, SSM, and SBR. These three linear models systematically infer a higher posterior probability for the spurious self-loop $X_2 \to X_2$ and suppress the true edge $X_1 \to X_2$. Obviously, the $X_2(t)$'s tend to exhibit an autocorrelation by virtue of the autocorrelation of the $X_1(t)$'s and the influence of $X_1(t)$ on $X_2(t+1)$. That is, the stronger the autocorrelation of $X_1(t)$ (the lower ε), the stronger the autocorrelation of $X_2(t)$. The three linear models BGe, SSM, and SBR cannot approximate the non-linear relationship between $X_1(t)$ and $X_2(t+1)$. Consequently, they systematically infer the 'second-best' explanation of the $X_2(t)$ data: explaining the realizations of the $X_2(t)$'s via a direct modeling of the autocorrelation between the $X_2(t)$'s themselves. Like the proposed BGM$_D$ model the original BGM model from Subsection 3.3.2, in principle, can approximate the relationship between X_1 and X_2 by piece-wise linear functions. However, additional studies in the original paper (Grzegorczyk *et al.*, 2011) revealed that the approximation is less effective, especially if the autocorrelation of $X_1(.)$ is strong. Therefore, the posterior probabilities change in favor for the spurious feedback loop $X_2 \to X_2$ with decreasing parameter ε. Finally, for $\varepsilon = 0.1$ (strong autocorrelation) BGM turns out to be as ineffective as the three linear models BGe, SSM, and SBR.

In another study two gene expression time series from *Arabidopsis thaliana* were used for evaluating the dynamic variant (BGM$_D$) of the BGM Bayesian network model. The *Arabidopsis thaliana* cells were sampled at $m = 13 \times 2$ hour time intervals with Affymetrix microarray chips. The expressions were measured twice independently under experimentally generated constant light condition, but differed with respect to the pre-histories. In the first experimental scenario, T_{20}, the plants were entrained in a 10h:10h light/dark-cycle, while the plants in the second experimental setting, T_{28}, were entrained in 14h:14h light/dark-cycle. The analysis focuses on $N = 9$ genes, namely LHY, CCA1, TOC1, ELF4, ELF3, GI, PRR9, PRR5, and PRR3, which are known to be involved in circadian regulation (Salome and McClung (2004) and Más (2008)). From the graphical representation of the connectivity structures in Figure 8 it can be seen that both Bayesian network models BGM and BGM$_D$ infer a two-stage process for these two *Arabidopsis thaliana* gene expression time series. The two stages are likely to be related to the diurnal nature of the dark-light cycle

[30] Note that like the standard Gaussian BGe model for Bayesian networks both approaches: SBR and SSM are linear models, and the goal was to investigate to what extent the non-linear modeling capability of the BGM Bayesian network model (from Subsection 3.3.2) and the BGM$_D$ Bayesian network model leads to an improvement when applied to time series of a non-homogeneous nature. Both models SBR and SSM were implemented as described in Rogers and Girolami (2005) and Beal *et al.* (2005), respectively. For the computational analysis and inference, the authors' own MATLAB programs were used, as referenced in their original publications.



(a) Bayesian network with BGe

(b) State space model (SSM)

(c) Sparse Bayesian regression (SBR)

(d) Bayesian network with BGM

(e) Bayesian network with BGM_D

Figure 7: **Marginal edge scores for synthetic data with N=2 nodes.** Each panel is laid out as a matrix, where rows and columns correspond to different autocorrelation parameters ε (rows) and signal-to-noise ratios SNR (columns). The figures show histograms of the inferred marginal edge scores, averaged over 50 independent data instantiations, as inferred with BGe (a), SSM (b), SBR (c), BGM (d), and BGM_D (e). The four bars represent the four possible edges: Left: self-loop $X_1 \to X_1$ (true); center left: $X_1 \to X_2$ (true); center right: self-loop $X_2 \to X_2$ (false); right: $X_2 \to X_1$ (false). Note that edge scores refer to marginal edge posterior probabilities in panels (a), (d), and (e). For the state space model (SSM) in panel (b), edge scores are the average absolute values of the posterior expectation of the interaction matrix elements defined in Eq. (8) of Beal *et al.* (2005). For the sparse Bayesian regression approach (SBR) the iterative non-linear maximization algorithm described in Rogers and Girolami (2005) has been repeated 100 times independently (with different initializations) for each single data set and the edge scores are given by the fractions of non-zero regression coefficients.

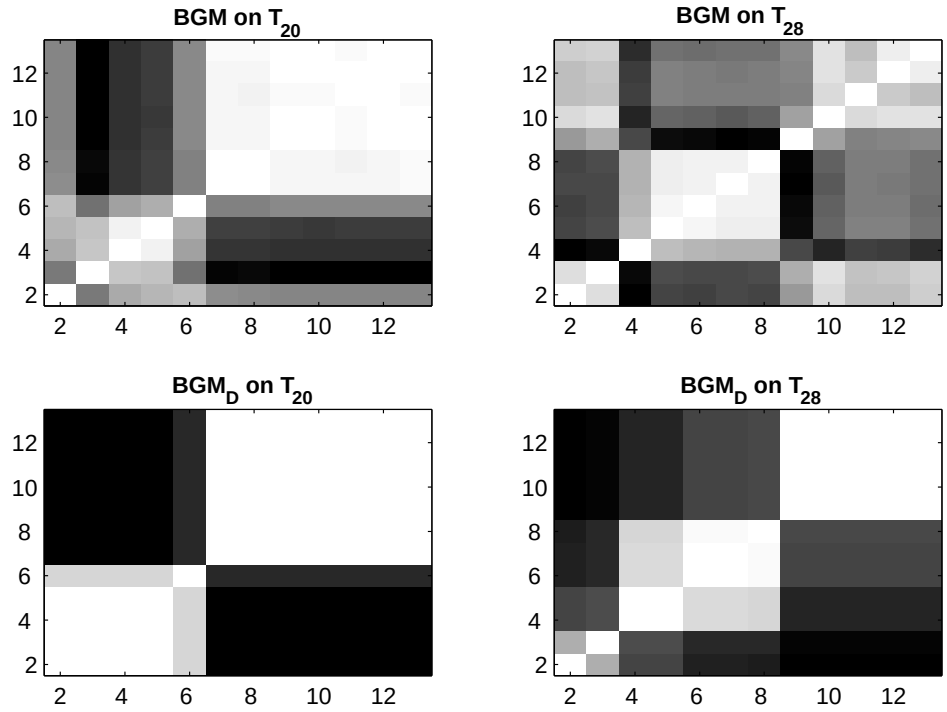


Figure 8: **Heat maps. Arabidopsis data.** Graphical heat map representations of the temporal connectivity structures for the two *Arabidopsis thaliana* time series T_{20} and T_{28}. *Top row:* Heat matrices for experiments T_{20} (top left) and T_{28} (top right) inferred with the BGM model from Subsection 3.3.2. *Bottom row:* Heat matrices for experiments T_{20} (bottom left) and T_{28} (bottom right) inferred with the novel BGM$_D$ model. In both Bayesian networks models self-feedback loops, such as $X \to X$, were excluded and it was ensured that both MCMC samplers converged; see original publications (Grzegorczyk *et al.*, 2010) for all details on the MCMC simulation lengths and the hyperparameter settings. Each heat map indicates the posterior probability of two time points being assigned to the same compartment (component/segment). The probabilities are represented by a grey shading, where white corresponds to a probability of 1, and black corresponds to a probability of 0. The numbers on the axes represent the time points of the time course experiment.

influencing the circadian genes and are therefore biologically plausible. The plants were subjected to different pre-histories, related to different lengths of the artificial, experimentally controlled light-dark cycle. The plants in experimental scenario T_{28} were entrained in an increased day length of 14 hours light followed by 14 hours darkness and in experiment T_{20} the plants were entrained in a decreased day length of 10 hours light followed by 10 hours darkness. As an effect of these two entrainments a phase shift in the gene regulatory processes between these two experiments was expected (Grzegorczyk *et al.*, 2008). The heat maps in the top row of Figure 8 show that the BGM Bayesian network model from Subsection 3.3.2 infers a certain trend for a phase shift of the changepoint (subjective day to subjective night) of about 4-6 hours as a consequence of the increased day length. A comparison of panel (a) and panel (b) in Figure 8 reveals that the connected blocks (compartments) of the time series are shifted along the diagonal by 2-3 time points (4-6 hours). From the bottom row of Figure 8 it can be seen that the novel BGM_D model infers the same trend but with a stronger separation score between these two biologically plausible compartments.[31]

[31] The BGM_D model is based on changepoints so that compartments once left cannot be re-visited. Therefore – different from the BGM model described in Subsection 3.3.2 – the BGM_D model has to allocate the last time points $(t_9, \ldots, t_{13})$ of time series T_{28} to an additional third component, since the first compartment (t_2, t_3) cannot be re-used after the transition to the second compartment $(t_4, \ldots, t_8)$.

3.3.4 Modeling non-homogeneous Bayesian networks with *node-specific* changepoints via Reversible Jump Markov Chain Monte Carlo

- **ORIGINAL PUBLICATION: Grzegorczyk, M.** and Husmeier, D. (2009) Non-stationary continuous Bayesian networks. **In:** *Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS2009)*, Bengio et al. (eds.), Curran Associates, 682-690.

Summary:
The objective of research was to generalize the BGM_D model from Subsection 3.3.3. A substantially greater model flexibility for dynamic Bayesian networks can be reached when the data segmentation is done individually for each single network node rather than assuming the same temporal segmentation for all network nodes. That is, instead of assuming one single multiple changepoint process with changepoints that are common to all nodes, the novel model, which was referred to as the *changepoint BGe* (cpBGe) Bayesian network model in the original publication (Grzegorczyk and Husmeier, 2009c), employs node-specific rather than network-wide changepoints. In the novel cpBGe Bayesian network model there is an independent multiple changepoint process for each individual network node. The application of the cpBGe model to gene expression time series from circadian clock-regulated genes in *Arabidopsis thaliana* leads to a plausible data segmentation, and the reconstructed *Arabidopsis thaliana* network shows features that are consistent with the biological literature. A precursor of the original publication (Grzegorczyk and Husmeier, 2009c) was presented at the conference on Pattern Recognition in Bioinformatics 2009 in Sheffield, England, and appeared in the conference proceedings (Grzegorczyk and Husmeier, 2009a).
Finally, it should be noted that the original focus of research was on using the concept of node-specific free allocation models to generalize the BGM model from Subsection 3.3.2. Using node-specific mixture models rather than node-specific changepoint processes would result in a proper *non-linear* rather than in a highly-flexible *non-homogeneous* Bayesian network model, namely the cpBGe model, presented here. While the algorithmic implementation of a generalized mixture BGM Bayesian network model is straightforward, the increased complexity of the latent variable configuration space – practically – turned out to introduce additional challenges for the mixing and convergence properties of the Markov Chain Monte Carlo (MCMC) sampling scheme. Since these problems could not be solved yet, so far only a generalization of the changepoint BGM_D model from Subsection 3.3.3 has been developed.

Motivation:
In many applications in systems biology the assumption of homogeneous gene regulatory processes is questionable, and models, such as the BGM_D Bayesian network model, presented in Subsection 3.3.3, which can deal with non-homogeneities appear to be more appropriate than the standard homogeneous Bayesian network model for discovering the regulatory interactions from data. But the BGM_D is based on the assumption that there are temporal changepoints at which all regulatory relationships of the network (i.e. *all* network parameter distributions) change. That is, only the network structure (the graph topology) is kept fixed among segments and allows for some information sharing, while the distributions of *all* network parameters have to be inferred independently for each segment. For many practical applications it seems

to be more realistic to assume that only the regulation of certain genes will change over time, and furthermore there may be different changepoint locations for different genes. That is, the changepoints may vary from gene to gene, and in particular, for short time series, the assumption that changepoints are always network-wide, i.e. always apply to all nodes, is therefore to restrictive. This restriction can lead to an immense loss of information, since the observations of each single gene, including those genes whose regulation does not vary over time, will be segmented according to those network-wide changepoints. This yields very short uninformative data segments, from which the distributions of the network parameters cannot be inferred properly. The motivation of the cpBGe model, presented here, is to avoid this information loss by using an independent multiple changepoint process for each individual gene.

The mathematical model:
To obtain a non-homogenous dynamic Bayesian network with node-specific changepoints, Eqns. (58-61) from Subsection 3.3.2 have to be generalized. Given the N-by-m data matrix, $\mathcal{D}$, of temporal observations, where $\mathcal{D}_{n,t}$ and $\mathcal{D}_{\pi_n,t}$ are the realizations of X_n and π_n at time point t, a latent allocation matrix, $\mathbf{V} = (\vec{\mathcal{V}}_1, \ldots, \vec{\mathcal{V}}_N)$, whose columns are node-specific allocation vectors, $\vec{\mathcal{V}}_n$, is used to allocate the observations of each node, X_n, individually to $\mathcal{K}_n$ segments. The numbers of segments required for the individual nodes, $\mathcal{K}_n$ $(n = 1, \ldots, N)$, can be summarized as a vector: $\vec{\mathcal{K}} = (\mathcal{K}_1, \ldots, \mathcal{K}_n)^T$. The n-th column, $\vec{\mathcal{V}}_n$, of $\mathbf{V}$ is the allocation vector for node X_n, and $\vec{\mathcal{V}}_n(t) = k$ means that the t-th observation of X_n, $\mathcal{D}_{n,t}$, is allocated to the k-th segment ($n = 1, \ldots, N$; and $k = 1, \ldots, \mathcal{K}_n$). In the novel cpBGe model the joint posterior probability of a graph, $\mathcal{G}$, the numbers of mixture components in $\vec{\mathcal{K}}$, and the allocation matrix, $\mathbf{V}$, factorizes as follows:

$$P(\mathcal{G}, \vec{\mathcal{K}}, \mathbf{V}|\mathcal{D}) = \frac{P(\mathcal{G}, \vec{\mathcal{K}}, \mathbf{V}, \mathcal{D})}{P(\mathcal{D})} \propto P(\mathcal{G}, \vec{\mathcal{K}}, \mathbf{V}, \mathcal{D}) = P(\vec{\mathcal{K}}, \mathbf{V})P(\mathcal{G})P(\mathcal{D}|\mathcal{G}, \mathbf{V}, \vec{\mathcal{K}}) \quad (81)$$

In the cpBGe model it is assumed that the pairs $(\vec{\mathcal{V}}_n, \mathcal{K}_n)$ $(n = 1, \ldots, N)$ are independently and identically (iid) distributed:

$$P(\mathbf{V}, \vec{\mathcal{K}}) = \prod_{n=1}^{N} P(\vec{\mathcal{V}}_n, \mathcal{K}_n) = \prod_{n=1}^{N} P(\vec{\mathcal{V}}_n|\mathcal{K}_n)P(\mathcal{K}_n) \quad (82)$$

For the prior probability distributions, $P(\mathcal{K}_n)$, on the node-specific numbers of segments, $\mathcal{K}_n$, iid truncated Poisson distributions with shape parameter $\lambda = 1$, restricted to $1 \leq \mathcal{K}_n \leq \mathcal{K}^{MAX}$, can be chosen $(n = 1, \ldots, N)$. The prior distributions, $P(\vec{\mathcal{V}}_n|\mathcal{K}_n)$, on the node-specific allocation vectors, $\vec{\mathcal{V}}_n$, conditional on the number of segments, $\mathcal{K}_n$, can be implicitly defined via a node-specific multiple changepoint process. That is, as described in Subsection 3.3.3, $\mathcal{K}_n$ segments are identified with $\mathcal{K}_n - 1$ changepoints: $b_{n,1}, \ldots, b_{n,\mathcal{K}_n-1}$ on the continuous interval $[2, m]$, and $\vec{\mathcal{V}}_n(t) = k$ if $b_{n,k-1} \leq t < b_{n,k}$, where $b_{n,0} = 2$ and $b_{n,\mathcal{K}_n} = m$. The node-specific changepoints, $b_{n,1}, \ldots, b_{n,\mathcal{K}_n-1}$, are again assumed to be distributed as the even-numbered order statistics of $\mathcal{L}_n := 2(\mathcal{K}_n - 1) + 1$ points $u_{n,1}, \ldots, u_{n,\mathcal{L}}$ uniformly and independently distributed on the interval $[2, m]$. See Subsection 3.3.3 for more details on the multiple changepoint process from Green (1995).
For a fixed set of parameters $\boldsymbol{\theta}$ the likelihood of the model is given by:

$$P(\mathcal{D}|\mathcal{G}, \mathbf{V}, \vec{\mathcal{K}}, \boldsymbol{\theta}) = \prod_{n=1}^{N} \prod_{k=1}^{\mathcal{K}_n} \prod_{t:\vec{\mathcal{V}}_n(t)=k} P(X_n = \mathcal{D}_{n,t}|\pi_n = \mathcal{D}_{\pi_n,t-1}, \boldsymbol{\theta}_n^k) \quad (83)$$

where $\mathcal{G} = \{\pi_1, \ldots, \pi_N\}$ is the graph and $\boldsymbol{\theta}_n^k$ $(n = 1, \ldots, N$; and $k = 1, \ldots, \mathcal{K}_n)$ is the node- and segment-specific parameter vector. Since $\vec{\mathcal{V}}_n(t) = k$ indicates that the

realization of node X_n at time point t has been generated by the k-th segment of a segmentation with $\mathcal{K}_n$ compartments, the latent 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 $\boldsymbol{\theta}_n^k$. Since the vectors, $\vec{\mathcal{V}}_n$, are node-specific, i.e. different nodes are modeled with different changepoint sets, the cpBGe model has a higher flexibility in modeling non-homogeneous regulatory relationships than the BGM_D model from Subsection 3.3.3. From Eq. (83) it follows that the marginal likelihood is given by:

$$P(\mathcal{D}|\mathcal{G},\mathbf{V},\vec{\mathcal{K}})=\int P(\mathcal{D}|\mathcal{G},\mathbf{V},\vec{\mathcal{K}},\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathcal{G})d\boldsymbol{\theta} \;=\; \prod_{n=1}^{N}\prod_{k=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n,(\vec{\mathcal{V}}_n,k)}) \tag{84}$$

$$\Psi(\mathcal{D}_n^{\pi_n,(\vec{\mathcal{V}}_n,k)})=\int \prod_{t:\vec{\mathcal{V}}_n(t)=k} P(X_n=\mathcal{D}_{n,t}|\pi_n=\mathcal{D}_{\pi_n,t-1},\boldsymbol{\theta}_n^k)P(\boldsymbol{\theta}_n^k|\pi_n)d\boldsymbol{\theta}_n^k \tag{85}$$

where $\mathcal{D}_n^{\pi_n,(\vec{\mathcal{V}}_n,k)}$ denotes the data subset, $\{(\mathcal{D}_{n,t},\mathcal{D}_{\pi_n,t-1}):\vec{\mathcal{V}}(t)=k\}$, consisting of all observations, $\mathcal{D}_{n,t}$ and $\mathcal{D}_{\pi_n,t-1}$, allocated to the k-th component by $\vec{\mathcal{V}}_n$ ($n=1,\ldots,N$; and $1\leq k\leq\mathcal{K}_n$). For each node, X_n ($n=1,\ldots,N$), and each node-specific segment, k ($k=1,\ldots,\mathcal{K}_n$), the $\Psi(\mathcal{D}_n^{\pi_n,(\vec{\mathcal{V}}_n,k)})$ term can be computed in closed-form when one of the standard Bayesian network models (e.g. the BGe model) is used. If no observation is allocated to the k-th component, symbolically $\mathcal{D}_n^{\pi_n,(\vec{\mathcal{V}}_n,k)}=\emptyset$, then $\Psi(\mathcal{D}_n^{\pi_n,(\vec{\mathcal{V}}_n,k)})$ is equal to 1. Eq. (85) is similar to Eq. (61) from Subsection 3.3.3 except that the allocation has become node-specific.

Inference:

The MCMC sampling scheme for the cpBGe model can be obtained by slightly modifying the sampling scheme for the BGM_D model, described in Subsection 3.3.3. It just has to be taken into account that the changepoints and allocation vectors in the cpBGe model are node-specific. A sample, $[\mathcal{G}_i,\mathbf{V}_i,\vec{\mathcal{K}}_i]_{i=1,\ldots,T}$, from the joint posterior distribution, $P(\mathcal{G},\mathbf{V},\vec{\mathcal{K}}|\mathcal{D})$, given in Eq. (81) can be obtained by combining the structure MCMC sampling scheme from Subsection 2.6 with moves that change the vector, $\vec{\mathcal{K}}$, and the latent allocation matrix, $\mathbf{V}$, via *node-specific* changepoint birth, changepoint death and changepoint re-allocation moves. That is, in each MCMC step either a structure MCMC single-edge-operation on the graph, $\mathcal{G}$, is performed while the allocation matrix, $\mathbf{V}$, and the vector, $\vec{\mathcal{K}}$, are left unchanged, or the move leaves the graph, $\mathcal{G}$, unchanged and proposes to change $[\mathbf{V},\vec{\mathcal{K}}]$ by a node-specific changepoint move.

If the graph prior, $P(\mathcal{G})$, in Eq. (83) can also be factorized: $P(\mathcal{G})=\prod_{n=1}^{N}P(\pi_n)$, where $\mathcal{G}=\{\pi_1,\ldots,\pi_N\}$, so that the local distributions, $P(\pi_n)$ ($n=1,\ldots,N$), are independent, then inference can be done independently for each single node X_n. Thus, different from the BGM_D model, where one single allocation vector applies to all nodes, the cpBGe model allows for parallel computing. This can be seen when inserting Eqns. (82) and (84) as well as the factorization of $P(\mathcal{G})$ into Eq. (83):

$$P(\mathcal{G},\mathbf{V},\vec{\mathcal{K}}|\mathcal{D})\propto\prod_{n=1}^{N}\left(P(\pi_n)P(\vec{\mathcal{V}}_n|\mathcal{K}_n)P(\mathcal{K}_n)\prod_{k=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n,(\vec{\mathcal{V}}_n,k)})\right) \tag{86}$$

where $\mathcal{G}=\{\pi_1,\ldots,\pi_N\}$, $\mathbf{V}=(\vec{\mathcal{V}}_1,\ldots,\vec{\mathcal{V}}_N)$, and $\vec{\mathcal{K}}=(\mathcal{K}_1,\ldots,\mathcal{K}_N)^T$. If follows from Eq. (86) that for each node, X_n, its parent set, π_n, its number of segments, $\mathcal{K}_n$, and its allocation vector, $\vec{\mathcal{V}}_n$, can be sampled according to X_n's contribution to the posterior

distribution:

$$P(\pi_n, \vec{\mathcal{V}}_n, \mathcal{K}_n|\mathcal{D}) \propto P(\pi_n)P(\vec{\mathcal{V}}_n|\mathcal{K}_n)P(\mathcal{K}_n)\prod_{k=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n,(\vec{\mathcal{V}}_n,k)}) \tag{87}$$

Independently for each node X_n a sample, $\left\{[\pi_{n_i}, \mathcal{K}_{n_i}, \vec{\mathcal{V}}_{n_i}]\right\}_{i=1,\ldots,T}$, can be obtained by sampling according to Eq. (87). And these node-specific samples of length T can be combined to a sample $\left\{[\mathcal{G}_i, \mathbf{V}_i, \vec{\mathcal{K}}_i]\right\}_{i=1,\ldots,T}$ from the posterior distribution in Eq. (86): For $i = 1, \ldots, T$ the graph can be built from the individual parent node sets, $\mathcal{G}_i := \{\pi_{1_i}, \ldots, \pi_{N_i}\}$, the allocation matrix, $\mathbf{V}_i$, is given by $\mathbf{V}_i := (\vec{\mathcal{V}}_{1_i}, \ldots, \vec{\mathcal{V}}_{N_i})$, and $\vec{\mathcal{K}}_i := (\mathcal{K}_{1_i}, \ldots, \mathcal{K}_{N_i})^T$.
To recapitulate this important feature of the cpBGe model: the node-specific samples $\left\{[\pi_{n_i}, \mathcal{K}_{n_i}, \vec{\mathcal{V}}_{n_i}]\right\}_{i=1,\ldots,T}$ can be generated in parallel rather than sequentially. If inference would be done sequentially, then in each MCMC step one single node, X_n, had to be randomly selected and a move on its parent set, π_n, or a move on its allocation vector, $\vec{\mathcal{V}}_n$, and number of segments, $\mathcal{K}_n$, would be performed. Assuming that parallel computing is utilized, then independently for each node, X_n, a new sample, $\left\{[\pi_{n_i}, \mathcal{K}_{n_i}, \vec{\mathcal{V}}_{n_i}]\right\}$, from Eq. (87) can be generated as follows: Given the current state of the Markov chain, $[\pi_n, \vec{\mathcal{V}}_n, \mathcal{K}_n]$, with probability $p = 0.5$ a structure MCMC move is performed and a new parent set, π_n^*, is proposed while $\vec{\mathcal{V}}_n$ and $\mathcal{K}_n$ are left unchanged. The new parent set is randomly chosen from the system $\mathcal{N}(\pi_n)$ of neighboring parent sets that can be reached from the current set, π_n, by adding one single node to π_n or removing one single node from π_n. The acceptance probability for the move from $[\pi_n, \vec{\mathcal{V}}_n, \mathcal{K}_n]$ to $[\pi_n^*, \vec{\mathcal{V}}_n, \mathcal{K}_n]$ is given by $A = \min\{1, R\}$, where

$$R = \frac{P(\pi_n^*, \mathcal{K}_n, \vec{\mathcal{V}}_n|\mathcal{D})}{P(\pi_n, \mathcal{K}_n, \vec{\mathcal{V}}_n|\mathcal{D})} \cdot \frac{|\mathcal{N}(\pi_n)|}{|\mathcal{N}(\pi_n^*)|} = \frac{P(\pi_n^*)\prod_{k=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n^*,(\vec{\mathcal{V}}_n,k)})}{P(\pi_n)\prod_{k=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n,(\vec{\mathcal{V}}_n,k)})} \cdot \frac{|\mathcal{N}(\pi_n)|}{|\mathcal{N}(\pi_n^*)|} \tag{88}$$

where the $\Psi(.)$ terms have been specified in Eq. (85), and $|.|$ is the cardinality operator. With probability $1 - p = 0.5$ the parent set π_n is left unchanged and a move on the changepoint set $\mathcal{B}_n$ of node X_n is performed. The node-specific changepoint set, $\mathcal{B}_n = \{b_{n,1}, \ldots, b_{n,\mathcal{K}_n-1}\}$, consists of $\mathcal{K}_n - 1$ changepoints, from which the node-specific allocation vector $\vec{\mathcal{V}}_n$ can be extracted: $\vec{\mathcal{V}}_n = \mathcal{V}(\mathcal{B}_n)$. The node-specific changepoint birth move adds one new changepoint to $\mathcal{B}_n$, the node-specific changepoint death move removes one changepoint from $\mathcal{B}_n$, and the node-specific changepoint reallocation move substitutes one single changepoint in $\mathcal{B}_n$ for a new one. The three move types can be selected with probabilities that depend on $\mathcal{K}_n$: $p_{b,\mathcal{K}_n}$ (for a birth), $p_{d,\mathcal{K}_n}$ (for a death), and $p_{r,\mathcal{K}_n}$ (for a re-allocation) move; see Eq. (70) for details. All three moves give a new changepoint set, $\mathcal{B}_n^*$, from which the new allocation vector, $\vec{\mathcal{V}}_n^* = \mathcal{V}(\mathcal{B}_n^*)$, can be extracted, while the new number of segments is given by $\mathcal{K}_n^* = |\mathcal{B}_n^*| + 1$.
The acceptance probability for a move from $[\pi_n, \mathcal{K}_n, \vec{\mathcal{V}}_n]$ to $[\pi_n, \mathcal{K}_n^*, \vec{\mathcal{V}}_n^*]$ is of the following form:

$$A = \min\left\{1, \frac{\prod_{k=1}^{\mathcal{K}_n^*}\Psi(\mathcal{D}_n^{\pi_n,(\vec{\mathcal{V}}_n^*,k)})}{\prod_{k=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n,(\vec{\mathcal{V}}_n,k)})} \times I_n \times P_n\right\} \tag{89}$$

where P_n is the prior probability ratio, and I_n is the inverse proposal probability ratio. The exact form of the factors, $I_n \times P_n$, depends on the move type. Deriving the factor, $I_n \times P_n$, as in Subsection 3.3.3 for the BGM_D model, yields for the three node-specific changepoint moves:

(i) For a *node-specific changepoint re-allocation*, one changepoint $b_{n,j} \in \mathcal{B}_n = \{b_{n,1}, \ldots, b_{n,\mathcal{K}_n-1}\}$ is randomly selected and removed. Subsequently, the replacement changepoint $b^{\dagger}_{n,j}$ is drawn from a uniform distribution on $[b_{n,j-1}, b_{n,j+1}]$ where $b_{n,0} = 2$ and $b_{n,\mathcal{K}_n} = m$. The factor $I_n \times P_n$ in Eq. (89) is then equal to:

$$I_n \times P_n = \frac{(b_{n,j+1} - b^{\dagger}_{n,j})(b^{\dagger}_{n,j} - b_{n,j-1})}{(b_{n,j+1} - b_j)(b_{n,j} - b_{n,j-1})} \tag{90}$$

If there is no changepoint in $\mathcal{B}_n$, the move is rejected and the Markov chain is left unchanged.

(ii) If a *node-specific changepoint birth move* on $\mathcal{B}_n = \{b_{n,1}, \ldots, b_{n,\mathcal{K}_n-1}\}$ is proposed, the location of the new changepoint $b^{\dagger}_n$ is randomly drawn from a uniform distribution on $[2, m]$. The proposal probability for this move is $p_{b,\mathcal{K}_n}/(m-2)$. The reverse death move is selected with probability $p_{d,\mathcal{K}_n+1}$ and consists in discarding randomly one of the $\mathcal{K}_n$ changepoints in $\mathcal{B}^*_n$. The factor $I_n \times P_n$ in Eq. (71) is then given by:

$$I_n \times P_n = \frac{2(2\mathcal{K}_n + 1)}{(m-2)} \cdot \frac{(b_{n,j+1} - b^{\dagger}_n)(b^{\dagger}_n - b_{n,j})}{(b_{n,j+1} - b_{n,j})} \tag{91}$$

where $b_{n,j} = \max\{b_{n,i} | b_{n,i} \leq b^{\dagger}_n\}$, and $b_{n,j+1} = \min\{b_{n,i} | b_{n,i} \geq b^{\dagger}_n\}$. For $\mathcal{K}_n = \mathcal{K}^{MAX}$ the birth of a new changepoint is invalid and the Markov chain is left unchanged.

(iii) A *node-specific changepoint death move* randomly selects one of the changepoints in $\mathcal{B}_n$ and proposes to remove the selected changepoint, $b^{\dagger}_n$. As the changepoint death move is the reverse of the birth move (ii), the factor $I_n \times P_n$ in Eq. (71) is given by:

$$I_n \times P_n = \frac{(m-2)}{2(2\mathcal{K}_n - 1)} \cdot \frac{(b_{n,j+1} - b_{n,j})}{(b_{n,j+1} - b^{\dagger}_n)(b^{\dagger}_n - b_{n,j})} \tag{92}$$

where $b_{n,j} = \max\{b_{n,i} | b_{n,i} \leq b^{\dagger}_n\}$, $b_{n,j+1} = \min\{b_{n,i} | b_{n,i} \geq b^{\dagger}_n\}$, and $b^{\dagger}_n$ is the changepoint, whose removal is proposed. For $\mathcal{K}_n = 1$ there is no changepoint in $\mathcal{B}_n$ that could be removed, and the Markov chain is left unchanged.

Selected application(s):

The cpBGe model was applied 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.[32] Four time series, which differed with respect to the pre-experiment entrainment condition and the time intervals: $T_e \in \{10h, 12h, 14h\}$ and $\tau \in \{2h, 4h\}$, were combined.[33] The focus of the analysis was on 9 circadian genes: LHY, TOC1, CCA1, ELF4, ELF3, GI, PRR9, PRR5, and PRR3, that are involved in circadian regulation. Having combined all four time series into a single set, the objective was to test whether the proposed cpBGe model would detect the different experimental phases.[34] The top panel of Figure 9 shows the marginal posterior probability of a changepoint for two selected genes

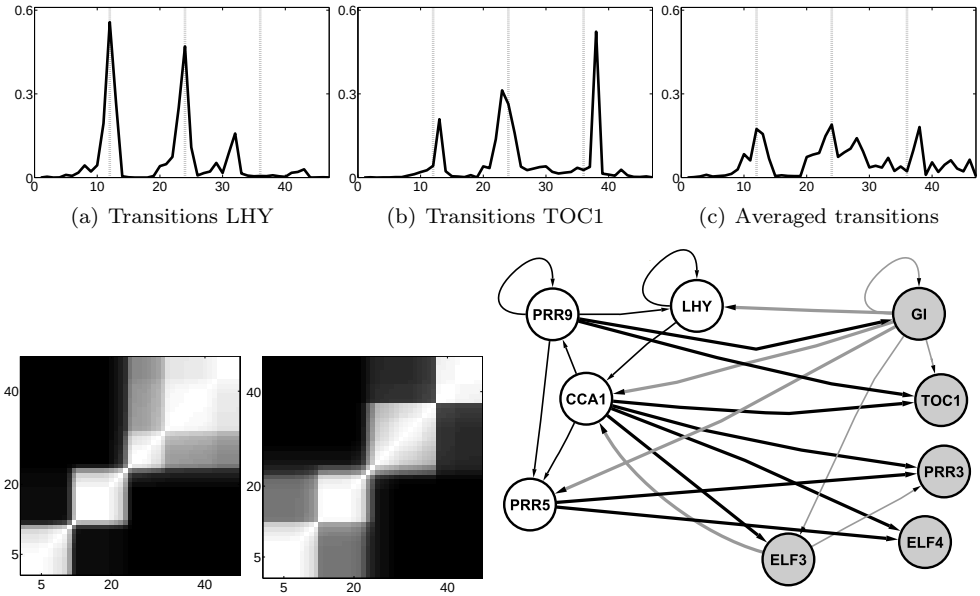
[32]The data were background-corrected and normalized according to standard procedures. RMA rather than GCRMA was used for reasons discussed in Lim *et al.* (2007).

[33]The data and more detailed information about the experimental protocols, can be obtained from Kieron *et al.* (2006), Mockler *et al.* (2007), and Grzegorczyk *et al.* (2008).

[34]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 bound-

(LHY and TOC1), and averaged over all genes. It is seen that the three concatenation points are clearly detected. There is a slight difference between the heights of the posterior probability peaks for LHY and TOC1. This behavior is also captured by the co-allocation matrices in the bottom row of Figure 9. This deviation indicates that the two genes are effected by the changing experimental conditions (entrainment, time interval) in different ways and thus provides a useful tool for further exploratory analysis. The bottom right panel of Figure 9 shows the extracted 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 being a central regulator. The reconstructed network also contains edges pointing into the opposite direction, from the evening genes back to the morning genes. This finding is also consistent with McClung (2006), where the evening genes were found to inhibit the morning genes via a negative feedback loop. In the reconstructed network, the connectivity within the group of evening genes is sparser than within the group of morning genes. This finding is consistent with the fact that following the light-dark cycle entrainment, the experiments were carried out in constant-light condition, resulting in a higher activity of the morning genes overall. Within the group of evening genes, the reconstructed network contains an edge between GI and TOC1. This interaction has been confirmed in Locke *et al.* (2005). Hence while a proper evaluation of the reconstruction accuracy is currently unfeasible – like Robinson and Hartemink (2009) and many related studies, there is no gold-standard owing to the unknown nature of the true interaction network – the study suggests that the essential features of the reconstructed network are biologically plausible and consistent with the literature.

ary time points were appropriately removed from the data. A proper mathematical treatment can be found in the original publication (Grzegorczyk and Husmeier, 2009c). 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.



(a) Transitions LHY (b) Transitions TOC1 (c) Averaged transitions

(d) Heat map LHY (e) Heat map TOC1 (f) Predicted network

Figure 9: **Results on the Arabidopsis gene expression time series.** *Top panels:* Average posterior probabilities (vertical axis) of transition times (horizontal axis) for two selected circadian genes (left: LHY, center: TOC1) and averaged over all 9 genes (right). The vertical dotted lines indicate the boundaries of the time series segments, which are related to different entrainment conditions and time intervals. *Bottom left and center panels:* Co-allocation matrices for the two selected genes LHY and TOC1. The axes represent time. The gray shading indicates the posterior probability of two time points being assigned to the same mixture component, ranging from 0 (black) to 1 (white). *Bottom right panel:* 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.

3.3.5 Modeling non-homogeneous Bayesian networks with *node-specific* changepoints via Gibbs sampling including dynamic programming for sampling changepoint configurations

- **ORIGINAL PUBLICATION: Grzegorczyk, M.** and Husmeier, D. (2011) Non-homogeneous dynamic Bayesian networks for continuous data. *Machine Learning*, **83(3)**, 355-419.

Summary:
The objective of research was to improve the inference for the cpBGe model from Subsection 3.3.4. Mixing and convergence of the MCMC sampling scheme, discussed in Subsection 3.3.4, can be very slow if the posterior landscape is highly peaked, and considerable efforts have been invested into improving mixing and convergence. It was found that the computational complexity can be substantially reduced by applying dynamic programming schemes, with which changepoints are sampled from the proper conditional distribution within a Gibbs sampling scheme, rather than pursuing Metropolis-Hastings MCMC inference based on birth, death, and re-allocation moves for individual changepoints. In the context of Bayesian mixture models two alternative dynamic programming schemes, based on different prior distributions for the changepoint processes, were presented by Fearnhead (2006). It was explored how to adopt these two schemes for the MCMC inference for the cpBGe Bayesian network model, and the performances of different Gibbs sampling variants were compared. For synthetic data sets and for data from *Arabidopsis thaliana* it could be shown that the computational costs, associated with MCMC inference for the cpBGe model, are drastically reduced when the novel Gibbs sampling schemes rather than the RJMCMC sampler from Subsection 3.3.4 is applied.

Motivation:
To sample from the posterior distribution, e.g. to sample from $P(\mathcal{G}, \mathbf{V}, \vec{\mathcal{K}}|\mathcal{D})$ in Eq. (81), all previous studies on non-homogeneous Bayesian networks (e.g. Robinson and Hartemink (2009), Grzegorczyk and Husmeier (2009c), Lèbre *et al.* (2010), and Grzegorczyk *et al.* (2011)) follow the same procedure: to sample the network, $\mathcal{G}$, they follow Madigan and York (1995) and apply the Metropolis-Hastings (MH) structure MCMC sampling scheme, based on single-edge operations; to sample the latent variables, $[(\mathbf{V}, \vec{\mathcal{K}})]$, they follow Green (1995) and apply Reversible Jump Markov Chain Monte Carlo (RJMCMC), based on changepoint birth, death, and reallocation moves. The objective of research in Grzegorczyk and Husmeier (2011b) was to develop Gibbs sampling schemes as alternatives to the RJMCMC sampling scheme. In Grzegorczyk and Husmeier (2011b) the advantage of the novel Gibbs sampling schemes was demonstrated for the non-homogeneous cpBGe Bayesian network model from Subsection 3.3.4, as one representative of the class of non-homogeneous Bayesian network models. The posterior distribution of the cpBGe model can be factorized according to Eq. (86):

$$P(\mathcal{G}, \mathbf{V}, \vec{\mathcal{K}}|\mathcal{D}) \propto \prod_{n=1}^{N} \left(P(\pi_n) P(\vec{\mathcal{V}}_n|\mathcal{K}_n) P(\mathcal{K}_n) \prod_{k=1}^{\mathcal{K}_n} \Psi(\mathcal{D}_n^{\pi_n,(\vec{\mathcal{V}}_n,k)}) \right) \tag{93}$$

where $\mathcal{G} = \{\pi_1, \ldots, \pi_N\}$, $\mathbf{V} = (\vec{\mathcal{V}}_1, \ldots, \vec{\mathcal{V}}_N)$, and $\vec{\mathcal{K}} = (\mathcal{K}_1, \ldots, \mathcal{K}_N)^T$. The goal is to infer the node-specific contributions to the posterior distribution:

$$P(\pi_n, \vec{\mathcal{V}}_n, \mathcal{K}_n|\mathcal{D}) \propto P(\pi_n)P(\vec{\mathcal{V}}_n|\mathcal{K}_n)P(\mathcal{K}_n)\prod_{k=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n,(\vec{\mathcal{V}}_n,k)}) \tag{94}$$

for each individual node X_n $(n = 1, \ldots, N)$ *independently.*

In the context of Bayesian mixture models, Fearnhead (2006) applies changepoint processes, for which changepoints occur at discrete time points, and Fearnhead considers two different priors for the numbers and locations of changepoints. These priors will be presented here in the context of node-specific changepoints:

(P1) The first prior is based on a Poisson prior $P(\mathcal{K}_n)$ for the number of segments, $\mathcal{K}_n$, and then a conditional prior on the positions of the $\mathcal{K}_n - 1$ changepoints, $\mathcal{B}_n = \{b_{n,1}, \ldots, b_{n,\mathcal{K}_n-1}\}$, where it is assumed that the changepoints, $b_{n,1}, \ldots, b_{n,\mathcal{K}_n-1}$, are distributed as the even-numbered order statistics of $\mathcal{L}_n := 2(\mathcal{K}_n - 1) + 1$ points $u_{n,1}, \ldots, u_{n,\mathcal{L}}$ uniformly and independently distributed on the *discrete* set $\{2 \ldots, m-1\}$. With $\vec{\mathcal{V}}_n = \mathcal{V}(\mathcal{B}_n)$ being the allocation vector implied by $\mathcal{B}_n$, mathematically $\vec{\mathcal{V}}_n(t) = k$ if $b_{n,k-1} < t \leq b_{n,k}$, where $b_{n,0} = 1$ is a pseudo changepoint. This corresponds to the prior distributions $P(\mathcal{K}_n)$ and $P(\vec{\mathcal{V}}_n|\mathcal{K}_n)$ from Subsection 3.3.4, except that a *discrete* rather than a continuous changepoint process is used.[35] In a pre-study on the cpBGe model (Grzegorczyk and Husmeier, 2011b) it was found that convergence and mixing of the original RJMCMC sampling scheme from Subsection 3.3.4 is approximately identical for the *discrete* variant and the *continuous* variant of the even-numbered order statistics prior for $P(\vec{\mathcal{V}}_n|\mathcal{K}_n)$.

(P2) The second prior is obtained from a point process on the positive and negative integers. The point process is specified by the probability mass function for the time between two successive points, for which a natural choice is the negative binomial distribution. The choice of this prior immediately imposes a prior distribution on the changepoint set, $\mathcal{B}_n$, and the vector of latent variables, $\vec{\mathcal{V}}_n = \mathcal{V}(\mathcal{B}_n)$, without any conditioning on the number of segments, $\mathcal{K}_n$, symbolically: $P(\vec{\mathcal{V}}_n|\mathcal{K}_n) \rightarrow P(\vec{\mathcal{V}}_n)$. Hence, the terms $\mathcal{K}_n$ in Eq. (87) and Eq. (94) become obsolete:

$$P(\pi_n, \vec{\mathcal{V}}_n|\mathcal{D}) \propto P(\pi_n)P(\vec{\mathcal{V}}_n)\prod_{k=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n,(\vec{\mathcal{V}}_n,k)}) \tag{95}$$

where $\mathcal{K}_n = |\mathcal{V}_n(m)| = |\mathcal{B}_n| + 1$, and m is the number of observations (columns) of the time series (data set), $\mathcal{D}$.

Given a Bayesian mixture model for which the latent variables are of the form of one of the two changepoint processes, discussed above, and the parameters can be integrated out in the (marginal) likelihood, as in Eqns. (84-85), Fearnhead (2006) shows that the changepoint set can be sampled from the proper posterior distribution *exactly*, with a dynamic programming scheme. To rephrase this: the changepoints can be sampled from the correct distribution directly, and no RJMCMC scheme (based on single changepoint moves, such as birth and death moves) with potential mixing and convergence problems is required. The computational complexity of this scheme is quadratic in the number of observations m.

The research idea was to adopt these two dynamic programming schemes for the non-homogeneous cpBGe dynamic Bayesian network model from Subsection 3.3.4. The

[35] The application of the corresponding dynamic programming scheme works with a discrete changepoint process only, since a closed form expression for $P(\vec{\mathcal{V}}_n|\mathcal{K}_n)$ is required; see original publication for details (Grzegorczyk and Husmeier, 2011b). The continuous changepoint process, which was originally employed for modeling $P(\vec{\mathcal{V}}|\mathcal{K})$ in the BGM_D model (see Subsection 3.3.3) and $P(\vec{\mathcal{V}}_n|\mathcal{K}_n)$ in the cpBGe model (see Subsection 3.3.4), cannot be used here, as it does not yield a closed form expression.

essential difference between the study in Fearnhead (2006) and the study in Grzegorczyk and Husmeier (2011b), presented here, is the following one: For the mixture models studied by Fearnhead (2006), the conditional distributions only depend 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; see Fearnhead (2006) for details. In the work presented here (Grzegorczyk and Husmeier, 2011b), the conditional distributions also depend on the network topology, $\mathcal{G}$, and thus, changing the network has a considerable impact on the conditional distributions so that the computational trick referred to above is no longer applicable. The implication is that the dynamic programming scheme comes with substantial computational overheads when applied for the cpBGe model, and it was therefore not clear from the outset whether it achieves any improvement over the RJMCMC scheme from Subsection 3.3.4.
However, the empirical evaluation revealed that the computational costs can be drastically reduced when the novel Gibbs sampling schemes rather than the RJMCMC sampler from Subsection 3.3.4 are applied for the cpBGe model. In particular, the study also revealed that the point process prior (P2) yields a faster converging Gibbs sampler than the prior (P1), whose continuous variant was originally used for the cpBGe model; see Subsection 3.3.4. Therefore, only the mathematical details of the dynamic programming scheme for the point process prior (P2) will be described in this subsection. The mathematical details for the alternative dynamic programming scheme, based on the prior (P1), can be found in the original publication (Grzegorczyk and Husmeier, 2011b).

The mathematical model:
Mathematically, the cpBGe model has been described in detail in Subsection 3.3.4. Here, in principle, the same model is considered, but the prior $P(\vec{\mathcal{V}}_n|\mathcal{K}_n)P(\mathcal{K}_n)$ is substituted for a point process prior, which indirectly specifies a joint prior on the number and positions of the changepoints. That is, instead of modeling $P(\mathcal{K}_n)$ explicitly, and the allocation vectors, $\vec{\mathcal{V}}_n$, conditional on $\mathcal{K}_n$, a point process prior is used to model the distances between successive changepoints. By adopting this point process prior the dynamic programming scheme becomes conceptually simpler and computationally much more efficient. This slightly modified version of the cpBGe model will therefore also be referred to as the *improved* cpBGe model.
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 a complete changepoint set, $\mathcal{B}_n = \{b_{n,1}, \ldots, b_{n,\mathcal{K}_n-1}\}$, with $b_{n,j} - b_{n,j-1} > 0$, $b_{n,1} \geq 2$ and $b_{n,\mathcal{K}_n} \leq m-1$ is:

$$P(\mathcal{B}_n) = g_0(b_{n,1} - b_{n,0}) \left(\prod_{j=2}^{\mathcal{K}_n - 1} g(b_{n,j} - b_{n,j-1}) \right) (1 - G(b_{n,\mathcal{K}_n} - b_{n,\mathcal{K}_n-1})) \qquad (96)$$

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

$$G(t) = \sum_{s=1}^{t} g(t); \quad G_0(t) = \sum_{s=1}^{t} g_0(t) \qquad (97)$$

are the cumulative distribution functions corresponding to $g(.)$ and $g_0(.)$. For $g(.)$ the probability mass function of the negative binomial distribution, NBIN(p,k), with parameters $p \in [0,1]$ and $k \in \mathbf{N}$ can be used:[36]

$$g(t) = \left(\begin{array}{c} t-1 \\ k-1 \end{array} \right) p^k (1-p)^{t-k} \quad (t = 1, 2, 3, \ldots) \tag{98}$$

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:

$$g_0(b_{n,1}) = \frac{1}{k} \sum_{i=1}^{k} \left(\begin{array}{c} (b_{n,1}-1)-1 \\ i-1 \end{array} \right) p^i (1-p)^{(b_{n,1}-1)-i} \tag{99}$$

Different from the continuous changepoint process, where segments can be empty and different changepoint sets, $\mathcal{B}_n$, can give the same allocation vector, $\mathcal{V}_n = \mathcal{V}(\mathcal{B}_n)$, the discrete point process prior avoids empty components and gives a one-to-one mapping between allocation vectors and changepoints: For $t = 2, \ldots, m$ and the changepoint set $\mathcal{B}_n = \{b_{n,1}, \ldots, b_{n,\mathcal{K}_n-1}\}$ it holds: $b_{n,k-1} < t \leq b_{n,k} \Leftrightarrow \vec{\mathcal{V}}_n(t) = k$. Hence, in addition to the notation $\vec{\mathcal{V}}_n = \mathcal{V}(\mathcal{B}_n)$, it can also be written $\mathcal{B}_n = \mathcal{V}^{-1}(\vec{\mathcal{V}}_n)$. Because of this isomorphism, the slightly modified cpBGe model can directly be formulated in terms of changepoint sets, $\mathcal{B}_n$, rather than in terms of latent allocation vectors, $\vec{\mathcal{V}}_n$. To this end, let $\mathcal{D}_n^{\pi_n}$ denote 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, and accordingly let $\mathcal{D}_n^{\pi_n}[s:t]$ denote the segment $\{(\mathcal{D}_{n,i}, \mathcal{D}_{\pi_n,i-1}) : s \leq i \leq t\}$ of adjacent observations. The local score of each data subset, $\mathcal{D}_n^{\pi_n}[s:t]$, is then

$$\Psi(\mathcal{D}_n^{\pi_n}[s:t]) = \int \prod_{i=s}^{t} P\Big(X_n = \mathcal{D}_{n,i} | \pi_n = \mathcal{D}_{\pi_n,i-1}, \boldsymbol{\theta}_n\Big) P(\boldsymbol{\theta}_n | \pi_n) d\boldsymbol{\theta}_n \tag{100}$$

and can be computed in closed-form, when a standard Bayesian network model is used. Instead of the allocation matrix, $\mathbf{V}$, the corresponding system of changepoint sets, $\mathbf{B} = \{\mathcal{B}_n | n = 1, \ldots, N\}$, which contains a changepoint set, $\mathcal{B}_n$, for each node, X_n, can be employed for the definition of the slightly modified cpBGe model. That is, as there is a one-to-one mapping between the allocation matrix, $\mathbf{V}$, and the system of changepoint sets, $\mathbf{B}$, the original representation:

$$P(\pi_n, \vec{\mathcal{V}}_n | \mathcal{D}) = P(\pi_n) P(\vec{\mathcal{V}}_n) \prod_{k=1}^{\mathcal{K}_n} \Psi(\mathcal{D}_n^{\pi_n, (\vec{\mathcal{V}}_n, k)}) \tag{101}$$

can be substituted for the following isomorphic representation:

$$P(\pi_n, \mathcal{B}_n | \mathcal{D}) \propto P(\pi_n) P(\mathcal{B}_n) \prod_{j=1}^{\mathcal{K}_n} \Psi(\mathcal{D}_n^{\pi_n}[(b_{n,j-1}+1) : b_{n,j}]) \tag{102}$$

where $\mathcal{B}_n = \mathcal{V}^{-1}(\vec{\mathcal{V}}_n)$ is the changepoint set that yields the allocation vector $\vec{\mathcal{V}}_n$, the $\Psi(.)$ terms have been specified in Eq. (100), the prior probability $P(\mathcal{B}_n)$ was defined in Eq. (96), and $b_{n,0} = 1$ and $b_{n,\mathcal{K}_n} = m$ are two pseudo changepoints. For each node X_n $(n = 1, \ldots, N)$ its contribution to the joint posterior distribution:

$$P(\mathcal{G}, \mathbf{B} | \mathcal{D}) \propto \prod_{n=1}^{N} P(\pi_n) P(\mathcal{B}_n) \prod_{j=1}^{\mathcal{K}_n} \Psi(\mathcal{D}_n^{\pi_n}[(b_{n,j-1}+1) : b_{n,j}]) \tag{103}$$

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

is given in Eq. (102).

Inference:
(i) Sampling parent sets, π_n, conditional on a fixed changepoint set, $\mathcal{B}_n$:
The structure MCMC sampling scheme from Subsection 2.6 is based on single-edge operations, and various studies, e.g. those in Friedman and Koller (2003) and Grzegorczyk and Husmeier (2008), have shown that the proposal scheme of the structure MCMC sampler leads to poor convergence and mixing, since simulations tend to get stuck in local optima. The RJMCMC sampling scheme for the cpBGe model, described in Subsection 3.3.4, infers graphs with the structure MCMC sampler and thus it is likely that the graph inference is suboptimal in terms of convergence. For standard homogeneous *dynamic* Bayesian networks an improvement can be achieved by sampling new parent sets $\pi_n^\star$ for each node X_n directly from the following (local) posterior distribution:

$$P(\pi_n^\star|\mathcal{D}) = \frac{\Psi(\mathcal{D}_n^{\pi_n^\star})P(\pi_n^\star)}{\sum_{\pi_n:|\pi_n|\leq\mathcal{F}}\Psi(\mathcal{D}_n^{\pi_n})P(\pi_n)} \tag{104}$$

where the $\Psi(.)$-scores of the standard (homogeneous) dynamic Bayesian network have been specified in Eq. (22), and the sum is over all valid parent node sets π_n subject to a fan-in restriction, $\mathcal{F}$, on the cardinality, $|\pi_n|$, symbolically: $|\pi_n| \leq \mathcal{F}$. For standard (homogeneous) dynamic Bayesian networks the local posterior distributions in Eq. (104) can be pre-computed and stored for each node X_n so that sampling parent sets from Eq. (104) tends to be computationally cheaper than sampling with the structure MCMC sampler, which is based on single edge operations. In the cpBGe model the parent node sets, $\pi_n^\star$, have to be sampled conditional on the changepoint set, $\mathcal{B}_n$, and the counterpart of Eq. (104) is given by:

$$P(\pi_n^\star|\mathcal{B}_n,\mathcal{D}) = \frac{P(\pi_n^\star)\prod_{j=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n^\star}[(b_{n,j-1}+1):b_{n,j}])}{\sum_{\pi_n:|\pi_n|\leq\mathcal{F}}P(\pi_n)\prod_{j=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n}[(b_{n,j-1}+1):b_{n,j}])} \tag{105}$$

where $\mathcal{K}_n = |\mathcal{B}_n|+1$ is the number of segments, and the $\Psi(.)$ terms have been specified in Eq. (100). Combing this sampling scheme for parent sets, $\pi_n^\star$, with changepoint birth, death, and reallocation moves is computationally expensive, since the distribution in Eq. (105) has to be re-computed after each single changepoint move from $\mathcal{B}_n$ to $\mathcal{B}_n^\star$. The bottleneck becomes obvious when taking into consideration that the three changepoint moves give relatively small steps in the configuration space of the possible changepoint sets, $\{\mathcal{B}_n\}$, so that a large amount of re-computation of Eq. (105) is required. But when combining this sampling scheme with a dynamic programming scheme for sampling changepoint sets, $\mathcal{B}_n$, directly from the conditional posterior distribution, $P(\mathcal{B}_n|\pi_n,\mathcal{D})$, the stepwise re-computation of Eq. (105) becomes computationally efficient. A dynamic programming scheme for sampling from $P(\mathcal{B}_n|\pi_n,\mathcal{D})$ will be described next.

(ii) Sampling changepoint sets, $\mathcal{B}_n$, conditional on a fixed parent set, π_n:
Let $Q(t|n,\pi_n)$ denote the probability of the observations $\mathcal{D}_{n,t:m} := \{\mathcal{D}_{n,t},\ldots,\mathcal{D}_{n,m}\}$ of node X_n given X_n's parent set, π_n, the parental observations, $\mathcal{D}_{\pi_n,(t-1):(m-1)} := \{\mathcal{D}_{\pi_n,t-1},\ldots,\mathcal{D}_{\pi_n,m-1}\}$, and a changepoint, $b^\dagger$, at time point $t-1$ ($t=2,\ldots,m$):

$$Q(t|n,\pi_n) = P(\mathcal{D}_{n,t:m}|\mathcal{D}_{\pi_n,(t-1):(m-1)}, b^\dagger = t-1) \tag{106}$$

$Q(t|n,\pi_n)$ can be computed by marginalizing over all possible changepoint subsets after the given changepoint, $b^\dagger = t-1$. E.g. for $t=2$ the given changepoint,

$b^\dagger = t-1$, corresponds to the pseudo changepoint, $b_{n,0} = 1$, so that the condition on $b^\dagger$ can be neglected:

$$\begin{aligned} Q(2|n,\pi_n) &= P(\mathcal{D}_{n,2:m}|\mathcal{D}_{\pi_n,1:(m-1)}) \\ &= \sum_{\mathcal{B}_n} P(\mathcal{B}_n) \prod_{j=1}^{\mathcal{K}_n} \Psi(\mathcal{D}_n^{\pi_n}[(b_{n,j-1}+1):b_{n,j}]) \end{aligned} \tag{107}$$

where the sum is over all possible changepoint sets, $\mathcal{B}_n = \{b_{n,1},\ldots,b_{n,\mathcal{K}_n-1}\}$, that divide the data, $\{(\mathcal{D}_{n,i},\mathcal{D}_{\pi_n,i-1}) : 2 \le i \le m\}$, pertaining to node X_n and its parent set π_n, into $\mathcal{K}_n \in \{1,\ldots,m-2\}$ disjunct segments, $\{\mathcal{D}_n^{\pi_n}[(b_{n,j-1}+1):b_{n,j}] : j = 1,\ldots,\mathcal{K}_n\}$, and $P(\mathcal{B}_n)$ was defined in Eq. (96).[37]
For $t = m$ there is a changepoint given at $b^\dagger = m-1$, and there cannot be any further (real) changepoints after $m-1$ so that

$$Q(m|n,\pi_n) = P(\mathcal{D}_{n,m:m}|\mathcal{D}_{\pi_n,(m-1):(m-1)}, b_\dagger = m-1) = \Psi(\mathcal{D}_n^{\pi_n}[m:m]) \tag{108}$$

and $\Psi(\mathcal{D}_n^{\pi_n}[m:m])$ can be computed in closed form with Eq. (100).
For $t = 3,\ldots,m-1$ a recursion can be used for the computation of the $Q(t|n,\pi_n)$ terms:

$$\begin{aligned} Q(t|n,\pi_n) &= \left(\sum_{s=t}^{m-1} \Psi(\mathcal{D}_n^{\pi_n}[t:s]) Q(s+1|n,\pi_n) g(s+1-t)\right) \\ &\quad + \Psi(\mathcal{D}_n^{\pi_n}[t:m])(1-G(m-t)) \end{aligned} \tag{109}$$

and

$$\begin{aligned} Q(2|n,\pi_n) &= \left(\sum_{s=2}^{m-1} \Psi(\mathcal{D}_n^{\pi_n}[2:s]) Q(s+1|n,\pi_n) g_0(s-1)\right) \\ &\quad + \Psi(\mathcal{D}_n^{\pi_n}[2:m])(1-G_0(m-2)) \end{aligned} \tag{110}$$

where $\Psi(\mathcal{D}_n^{\pi_n})[2:m] = \Psi(\mathcal{D}_n^{\pi_n})$, and $G_0(t) = \sum_{s=1}^{t} g_0(s)$. The proof of these relationships can be found in the original publication (Grzegorczyk and Husmeier, 2011b).
Having computed these probabilities, $Q(t|n,\pi_n)$ $(t = 2\ldots,m)$, via the recursion in Eqns. (109-110), another recursion[38] can be set up to sample a set of changepoints, $\mathcal{B}_n = \{b_{n,1},\ldots,b_{n,\mathcal{K}_n-1}\}$, from the correct posterior distribution:

$$P(\mathcal{B}_n|\pi_n,\mathcal{D}) = \frac{P(\mathcal{B}_n)\prod_{j=1}^{\mathcal{K}_n}\Psi(\mathcal{D}_n^{\pi_n}[(b_{n,j-1}+1):b_{n,j}])}{Q(2|n,\pi_n)} \tag{111}$$

where $b_{n,0} = 1$, and $Q(2|n,\pi_n)$ was defined in Eq. (107) and can be computed with Eqns. (109-110).
The posterior distribution of the first changepoint, $b_{n,1}$, given the parent set, π_n, is:

$$P(b_{n,1} = t|\pi_n,\mathcal{D}_n^{\pi_n}) = \Psi(\mathcal{D}_n^{\pi_n}[2:t]) Q(t+1|n,\pi_n) \frac{g_0(t)}{Q(2|n,\pi_n)} \tag{112}$$

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

$$P(\mathcal{K}_n = 1|\pi_n,\mathcal{D}_n^{\pi_n}) = \Psi(\mathcal{D}_n^{\pi_n}[2:m]) \frac{1-G_0(m-2)}{Q(2|n,\pi_n)} \tag{113}$$

[37] The number of segments, $\mathcal{K}_n$, varies between 1 and $m-1$, where $\mathcal{K}_n = 1$ means that there is no changepoint, $\mathcal{B}_n = \emptyset$, and the upper bound is given by $\mathcal{K}_n = m-1$, as there can be up to $m-2$ changepoints in between the two pseudo changepoints $b_{n,0} = 1$ and $b_{n,\mathcal{K}_n} = m$.
[38] Details on how to derive the second recursion can also be found in the original publication.

where $G_0(.)$ was defined in Eq. (97), and $[1 - G_0(m-2)]$ is the prior probability of the absence of any changepoint.[39]
The posterior distribution of the j-th changepoint, $b_{n,j}$, given the parent node set, π_n, and the previous changepoint, $b_{n,j-1} = s$, is:

$$\begin{aligned} P_t &:= P(b_{n,j} = t | b_{n,j-1} = s, \mathcal{D}_n^{\pi_n}) \\ &= \Psi(\mathcal{D}_n^{\pi_n}[(s+1):t]) Q(t+1|n,\pi_n) \frac{g(t-s)}{Q(s+1|n,\pi_n)} \end{aligned} \tag{114}$$

for $t = b_{n,j-1}+1, \ldots, m-1$. Thus, given a changepoint at $b_{n,j-1} = s$, the location of the next changepoint can be sampled from the discrete mass probability distribution $[P_{s+1}, \ldots, P_{m-1}, P_{\nabla(s)}]$ where $P_{\nabla(s)}$ is the probability for no further changepoints:

$$P_{\nabla(s)} := \Psi(\mathcal{D}_n^{\pi_n}[(s+1):m]) \frac{1 - G_0(m-s-1)}{Q(s+1|n,\pi_n)} \tag{115}$$

Having sampled the complete changepoint set, $\mathcal{B}_n = \{b_{n,1}, \ldots, b_{n,k-1}\}$, from these conditional distributions, the number of segments is $\mathcal{K}_n = k$.
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 Eqns. (109-110). This function is then used in Eqns. (114-115), where in a second sweep through the data the changepoint set, $\mathcal{B}_n$, is sampled from the conditional distribution $P(\mathcal{B}_n|\pi_n,\mathcal{D})$. The computational complexity is quadratic in the length of the time series, $\mathcal{O}(m^2)$. A sequence of changepoints, $\mathcal{B}_n$, uniquely determines the number of changepoints, $\mathcal{K}_n = |\mathcal{B}_n| + 1$, and the allocation vector, $\vec{\mathcal{V}}_n = \mathcal{V}(\mathcal{B}_n)$, since the two representations are isomorphic.

(iii) Sampling graphs, $\mathcal{G}$, and a system of node-specific changepoint sets, B, from the joint posterior distribution, $P(\mathcal{G}, \mathbf{B}|\mathcal{D})$, given in Eq. (103):
For each node, X_n, the parent node set, π_n, can be sampled conditional on the changepoint set, $\mathcal{B}_n$, from the conditional posterior distribution, $P(\pi_n|\mathcal{B}_n,\mathcal{D})$, given in Eq. (105), and a dynamic programming scheme can be used for each node X_n for sampling changepoint sets $\mathcal{B}_n$ conditional on the parent set, π_n, from the conditional posterior distribution, $P(\mathcal{B}_n|\pi_n,\mathcal{D})$, when a slightly modified prior distribution for the numbers and locations of changepoints is used. Bringing these two results together, a Gibbs MCMC sampling scheme for sampling from the joint posterior distribution, $P(\pi_n, \mathcal{B}_n|\mathcal{D})$, can be constructed by iteratively sampling from the two "full conditional distributions": $P(\pi_n|\mathcal{B}_n,\mathcal{D})$ and $P(\mathcal{B}_n|\pi_n,\mathcal{D})$. Using this Gibbs sampling schemes independently for each node X_n $(n = 1, \ldots, N)$ gives node-specific samples, $\{(\pi_{n_i}, \mathcal{B}_{n_i}) : i = 1, \ldots, T\}$, from Eq. (95), which can be merged into a sample, $\{(\mathcal{G}_i, \mathbf{B}_i) : i = 1, \ldots, T\}$, from the joint posterior distribution, $P(\mathcal{G}, \mathbf{B}|\mathcal{D})$, in Eq. (103), symbolically: $\mathbf{B}_i := \{\mathcal{B}_{1_i}, \ldots, \mathcal{B}_{N_i}\}$, and $\mathcal{G}_i := \{\pi_{1_i}, \ldots, \pi_{N_i}\}$. 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 can be reached in fewer MCMC steps.

Selected application(s):
To assess the improvement in convergence and mixing achieved with the Gibbs sampling/dynamic programming schemes the merged *Arabidopsis thaliana* gene expres-

[39] Recall that for a dynamic Bayesian network with time lag $\tau = 1$ 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 $t = m-1$.

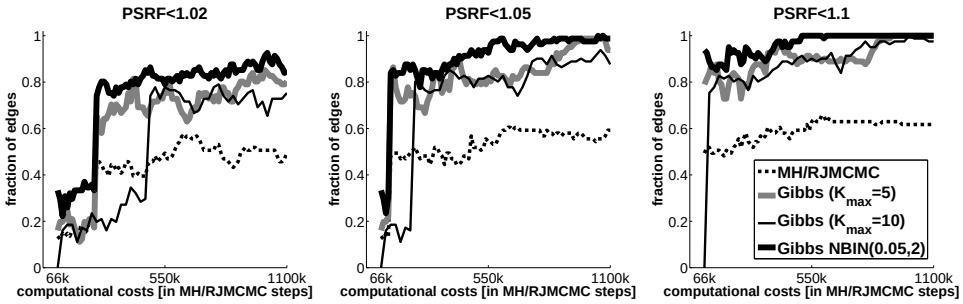


Figure 10: **Convergence diagnostics for four different sampling schemes for the cpBGe Bayesian network model.** The graphs show the proportion of edges for which the PSRF lies below the indicated threshold, satisfying the respective convergence criterion. The horizontal axes represent simulation time, measured in terms of the equivalent number of MH/RJMCMC steps. Four MCMC sampling schemes for the cpBGe model from Subsections 3.3.4 and 3.3.5 are compared: The standard MH/RJMCMC approach described in Subsection 3.3.4, and three variants of the Gibbs sampling/dynamic programming scheme described in this subsection. Gibbs($K_{max} = 10$) and Gibbs($K_{max} = 5$) employ a Poisson prior with truncation threshold $\mathcal{K}_{max}$ for the number of components, and use the discrete version of the even-numbered order statistics prior (conditional on the number of components) for the changepoint locations. The Gibbs-NBIN sampler follows Fearnhead (2006) and uses the prior imposed by the point process prior of Eq. (96) for the changepoint sets.

sion time series[40], described in Subsection 3.3.4, was re-analyzed with four different MCMC sampling schemes: **(i)** The standard Metropolis Hastings (MH) RJMCMC scheme (MH/RJMCMC), as described in Subsection 3.3.4 except that the discrete (P1) rather than the continuous even-numbered order statistics prior was used, and **(ii)-(iv)**: three variants of the Gibbs sampling/dynamic programming scheme. The three Gibbs sampling schemes differ with respect to the prior distribution on the changepoints. The first Gibbs sampling scheme **(ii)**, referred to as Gibbs($K_{max} = 10$), imposes a Poisson prior with truncation threshold $\mathcal{K}_n \leq 10$ on the number of components, $P(\mathcal{K}_n)$, and the discrete variant of the even-numbered order statistics prior (P1) is imposed on the segmentations, $P(\vec{\mathcal{V}}_n|\mathcal{K}_n)$. The second Gibbs sampling scheme **(iii)**, referred to as Gibbs($K_{max} = 5$), is identical to the Gibbs sampling scheme (ii), except that the truncation threshold has been lowered to $\mathcal{K}_n \leq 5$. The third Gibbs sampling scheme **(iv)**, referred to as Gibbs-NBIN, uses the prior (P2), imposed by the point process prior of Eq. (96), with hyperparameters $p = 0.05$ and $k = 2$ and was described in detail in this subsection.
To assess the degree of convergence, for each scheme **(i)-(iv)** five independent MCMC simulations with different initializations were performed and potential scale reduction factors (PSRFs) were computed for all potential edges, as described in Subsection 2.7. As convergence criterion the fraction of individual edges whose PSRF was lower than a pre-specified threshold was then used to assess convergence.[41] Due to different computational costs of the individual steps of the four sampling schemes – a Gibbs step

[40]The focus was on reverse-engineering a network consisting of 9 circadian genes: LHY, TOC1, CCA1, ELF4, ELF3, GI, PRR9, PRR5, and PRR3. Four time series, which differed with respect to the pre-experiment entrainment condition and the measured time intervals, were combined to one single time series. Boundary time points were appropriately removed to ensure for all pairs of consecutive time points that a proper conditional dependence relation is given. See Subsection 3.3.4 for a more detailed description.

[41]Recall that PSRF=1 indicates perfect convergence, and PSRF$\leq$1.1 is usually taken as an indication of sufficient convergence.

based on dynamic programming is substantially more expensive than a RJMCMC step – the PSRF scores were plotted against the simulation time, measured in terms of conventional RJMCMC steps.[42]
The results are shown in Figure 10. The three Gibbs sampling schemes based on dynamic programming significantly outperform the conventional RJMCMC sampling scheme (MH/RJMCMC). When comparing the three different dynamic programming schemes, Gibbs-NBIN performs slightly better than Gibbs($K_{max} = 10$) and Gibbs($K_{max} = 5$), in agreement with empirical results for Bayesian mixture models in Fearnhead (2006).

[42] 1100k RJMCMC (5500 Gibbs-NBIN) steps take 45 min with MATLAB code on a SunFire X4100M2 machine with MAD Opteron 2224 SE dual-core processor.

3.3.6 Regularization between Bayesian network models with network-wide and node-specific changepoints

- **ORIGINAL PUBLICATION: Grzegorczyk, M.** and Husmeier, D. (2011) Improvements in the reconstruction of time-varying networks: dynamic programming and regularization by information sharing among genes. *Bioinformatics*, **27(5)**, 693-699.

Summary:
Various different non-homogeneous dynamic Bayesian network models have been proposed and applied in the literature, and these models can be divided into two classes according to whether changepoints are common to the whole network (*class 1*), or varying from node to node (*class 2*). The approach of *class 1*, pursued in Grzegorczyk *et al.* (2008), Robinson and Hartemink (2009), and Grzegorczyk *et al.* (2011), is over-restrictive, as it does not allow for individual nodes to be affected by changing processes in different ways.[43] The approach of *class 2*, pursued in Grzegorczyk and Husmeier (2009c), Lèbre (2007), and Lèbre *et al.* (2010) is potentially over-flexible, as it does not provide any information sharing among the nodes. When an organism undergoes transitional changes, e.g. morphogenic transitions during embryogenesis, from larva to pupa or from pupa to adult fly in Drosophila, one would expect the majority of genes to be affected by these transitions in identical ways. However, there is no mechanism in the fully flexible model that incorporates this prior notion of commonality. The objective of research was therefore to explore a Bayesian clustering scheme akin to the weight sharing principle in neural computation (Nowlan and Hinton, 1992), by which nodes are assigned to clusters that are characterized by common changepoints. It has been demonstrated on synthetic data that this regularization scheme subsumes the aforementioned approaches as limiting cases, and that it automatically identifies the right trade-off between them in a data-driven manner. Finally, it has been demonstrated for gene expression profiles from a synthetically designed *Saccharomyces cerevisiae* strain under switching carbon metabolism that this regularization scheme yields a novel dynamic Bayesian network model that outperforms two alternative established network reconstruction methods from the biology literature.

Motivation:
The non-homogeneous BGM_D Bayesian network model, described in Subsection 3.3.3, is based on changepoints that are common to the whole network, i.e. the changepoints are *network-wide* and apply to all nodes. It has therefore been argued in Subsection 3.3.4 that the cpBGe Bayesian network model, which possesses node-specific changepoints, is more flexible and superior to the BGM_D model. But on the other hand, the cpBGe model may be potentially over-flexible, since there is no information sharing among nodes. That is, even if all nodes in the network share the same changepoints, these changepoints have to be inferred *independently* for each single node; especially for short time series (sparse data) the more flexible cpBGe model approach is then suboptimal too.
The research idea was to develop a novel Bayesian network model that regularizes between *network-wide* and *node-specific* changepoints; i.e. to develop a model that

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

regularizes between the BGM_D and the cpBGe model. The novel model uses a relatively simple Bayesian clustering scheme to subdivide the nodes into clusters and infers an independent changepoint set for each cluster of genes; all genes in a cluster share the same changepoints. Effectively, the novel model can be seen as a generalization of both models: BGM_D and cpBGe: It corresponds to the BGM_D model from Subsection 3.3.3 if there is only one single cluster that contains all nodes, as all genes share the same network-wide changepoints of this single cluster then. And the novel model corresponds to the cpBGe model from Subsection 3.3.4 and Subsection 3.3.5 if every single node possesses its own cluster, as cluster-specific changepoints are then effectively node-specific. Following the Bayesian paradigm the network, the clusters of genes and the cluster-specific changepoint sets are sampled from the posterior distribution using a mixture of Gibbs sampling and Metropolis-Hastings sampling steps.

The mathematical model:
The improved (slightly modified) cpBGe model from Subsection 3.3.5 can be extened by introducing a cluster function, $\mathcal{C}(.)$, that allocates the nodes $X_1, \ldots, X_N$ to $\tilde{c}$ $(1 \leq \tilde{c} \leq N)$ non-empty clusters, each characterized by its own changepoint set $\mathcal{B}_c^{\mathcal{C}}$ $(1 \leq c \leq \tilde{c})$.[44] Let $\mathcal{C}(n) = c$ indicate that the n-th node, X_n, is allocated to cluster c $(n = 1, \ldots, N$; and $1 \leq c \leq \tilde{c})$. Hence, for $\mathcal{C}(n) = c$ the changepoint set for the observations of node X_n is given by: $\mathcal{B}_{\mathcal{C}(n)}^{\mathcal{C}} = \mathcal{B}_c^{\mathcal{C}}$, and all nodes X_n with $\mathcal{C}(n) = c$ $(n = 1, \ldots, N)$ share this changepoint set, $\mathcal{B}_c^{\mathcal{C}}$. As explained in Subsection 3.3.5, there is an isomorphism between changepoint sets and allocation vectors if a point process prior for the cluster-specific changepoint locations is employed. Hence, the novel model could also be formulated in terms of cluster-specific allocation vectors $\vec{\mathcal{V}}_c^{\mathcal{C}}$, as done in the original publication (Grzegorczyk and Husmeier, 2011a). $\vec{\mathcal{V}}_c^{\mathcal{C}}(t) = k$ would then indicate that the t-th observations of all those nodes in cluster c are allocated to the k-th (cluster-specific) segment $(t = 2, \ldots, m$; and $k = 1, \ldots, \mathcal{K}_c)$. The cluster-specific changepoint set, $\mathcal{B}_c^{\mathcal{C}} = \{b_{c,1}, \ldots, b_{c,\mathcal{K}_c - 1}\}$, could then be extracted from $\vec{\mathcal{V}}_c^{\mathcal{C}}$: $\vec{\mathcal{V}}_c^{\mathcal{C}}(t) = k \Leftrightarrow b_{c,j-1} < t \leq b_{c,j}$, where $b_{c,0} = 1$ and $b_{c,\mathcal{K}_n} = m$ are two pseudo changepoints.
The changepoint sets for the $\tilde{c}$ clusters, induced by the cluster function, $\mathcal{C}(.)$, can be summarized as a system of changepoint sets, $\mathbf{B}^{\mathcal{C}} = \{\mathcal{B}_1^{\mathcal{C}}, \ldots, \mathcal{B}_{\tilde{c}}^{\mathcal{C}}\}$, where the cardinality of $\mathbf{B}^{\mathcal{C}}$ is identical to the number of clusters, $\tilde{c}$, induced by $\mathcal{C}(.)$. The joint probability distribution for the novel model, which might be referred to as the *regularized* cpBGe model, is given by:

$$\begin{aligned} P(\mathcal{G}, \mathcal{C}, \mathbf{B}^{\mathcal{C}}, \mathcal{D}) &= P(\mathcal{C}) P(\mathbf{B}^{\mathcal{C}}|\mathcal{C}) P(\mathcal{G}) P(\mathcal{D}|\mathcal{G}, \mathcal{C}, \mathbf{B}^{\mathcal{C}}) \\ &= P(\mathcal{C}) \left(\prod_{c=1}^{\tilde{c}} P(\mathcal{B}_c^{\mathcal{C}}) \right) P(\mathcal{G}) P(\mathcal{D}|\mathcal{C}, \mathbf{B}^{\mathcal{C}}, \mathcal{G}) \end{aligned} \tag{116}$$

where $\mathbf{B}^{\mathcal{C}} = \{\mathcal{B}_1^{\mathcal{C}}, \ldots, \mathcal{B}_{\tilde{c}}^{\mathcal{C}}\}$, and $\tilde{c}$ is the number of non-empty node clusters induced by $\mathcal{C}$. For the prior distribution $\text{P}(\mathcal{C})$ a uniform distribution on all clustering functions that give non-empty clusters can be used. It can be seen from Eq. (116) that the prior distributions $P(\mathcal{B}_c^{\mathcal{C}})$ $(c = 1, \ldots, \tilde{c})$ on the cluster-specific changepoint sets are assumed to be independent and identical. As for the modified cpBGe model from Subsections 3.3.5 these distributions can be specified via point process models on the

[44] The upper index $\mathcal{C}$ is added to avoid confusion. For the novel model $\mathcal{B}_c^{\mathcal{C}}$ denotes the *cluster-specific* changepoint set for the nodes in the c-th cluster, while $\mathcal{B}_n$ was used in Subsection 3.3.4 and Subsection 3.3.5 to denote the *node-specific* changepoint set for the n-th node in the (improved) cpBGe model.

positive *and* negative integers. This yields for $c = 1, \ldots, \tilde{c}$:

$$P(\mathcal{B}_c^{\mathcal{C}}) = g_0(b_{c,1}) \left(\prod_{j=2}^{\mathcal{K}_c - 1} g(b_{c,j} - b_{c,j-1}) \right) (1 - G(b_{c,\mathcal{K}_c} - b_{c,\mathcal{K}_c - 1})) \tag{117}$$

where $\mathcal{B}_c^{\mathcal{C}} = \{b_{c,1}, \ldots, b_{c,\mathcal{K}_c - 1}\}$, and the functions $g(.)$, $G(.)$, and $g_0(.)$ have been specified in Eqns. (97-99).
Different from the cpBGe model, nodes that are in the same cluster c ($1 \leq c \leq \tilde{c}$) share the same changepoint set, $\mathcal{B}_c^{\mathcal{C}}$, and will therefore be "penalized" only once via $P(\mathcal{B}_c^{\mathcal{C}})$, defined in Eq. (117).[45] Hence, the novel regularized cpBGe model encourages information sharing among nodes with respect to changepoint locations.
The marginal likelihood $P(\mathcal{D}|\mathcal{C}, \mathbf{B}^{\mathcal{C}}, \mathcal{G})$ is given by:

$$P(\mathcal{D}|\mathcal{C}, \mathbf{B}^{\mathcal{C}}, \mathcal{G}) = \prod_{n=1}^{N} \prod_{k=1}^{\mathcal{K}_{\mathcal{C}(n)}} \Psi(\mathcal{D}_n^{\pi_n}[(b_{\mathcal{C}(n),k-1} + 1) : b_{\mathcal{C}(n),k}]) \tag{118}$$

where $\mathcal{C} : \{1, \ldots, N\} \to \{1, \ldots, \tilde{c}\}$ is the cluster function, $\mathcal{K}_c = |\mathcal{B}_c^{\mathcal{C}}| + 1$ is the number of segments for the nodes in cluster c ($c = 1, \ldots, \tilde{c}$), $\mathcal{D}_n^{\pi_n}[(b_{c,k-1} + 1) : b_{c,k}])$ is the subset of adjacent observations, $\{(\mathcal{D}_{n,i}, \mathcal{D}_{\pi_n,i-1}) : b_{c,k-1} + 1 \leq i \leq b_{c,k}\}$, pertaining to node X_n and its parent set, π_n, and $b_{c,k-1}$ and $b_{c,k}$ are the $(k-1)$-th and the k-th changepoint in the cluster-specific set, $\mathcal{B}_c^{\mathcal{C}}$. The node- and segment-specific probabilities $\Psi(\mathcal{D}_n^{\pi_n}[(b_{c,k-1} + 1) : b_{c,k}])$ in Eq. (118) can be computed independently and in closed-form for each node X_n ($n = 1, \ldots, N$) and each segment k ($k = 1, \ldots, \mathcal{K}_c$) with one of the standard Bayesian network models; e.g. the Gaussian BGe model.
If the graph prior, $P(\mathcal{G})$, can also be factorized, $P(\mathcal{G}) = \prod_{n=1}^{N} P(\pi_n)$, the posterior probability distribution, $P(\mathcal{G}, \mathbf{B}^{\mathcal{C}}, \mathcal{C}|\mathcal{D})$, of the novel regularized cpBGe model is proportional to:

$$P(\mathcal{G}, \mathbf{B}^{\mathcal{C}}, \mathcal{C}|\mathcal{D}) \propto \left(\prod_{c=1}^{\tilde{c}} P(\mathcal{B}_c^{\mathcal{C}}) \right) \prod_{n=1}^{N} P(\pi_n) \prod_{k=1}^{\mathcal{K}_{\mathcal{C}(n)}} \Psi(\mathcal{D}_n^{\pi_n}[(b_{\mathcal{C}(n),k-1} + 1) : b_{\mathcal{C}(n),k}]) \tag{119}$$

For a clustering function $\mathcal{C}(.)$ that yields $\tilde{c} = N$ non-empty clusters, and therefore a one-to-one mapping between clusters and nodes, e.g. $\mathcal{C}(n) = n$ ($n = 1, \ldots, N$), Eq. (119) is exactly identical to the posterior distribution of the (slightly modified) cpBGe model, given in Eq. (103). This can be seen when defining node-specific changepoint sets as follows: $\mathcal{B}_n := \mathcal{B}_{\mathcal{C}(n)}^{\mathcal{C}}$ ($n = 1, \ldots, N$) and re-writing Eq. (119):

$$\begin{aligned} P(\mathcal{G}, \mathbf{B}^{\mathcal{C}}, \mathcal{C}|\mathcal{D}) \quad &\propto \quad \left(\prod_{c=1}^{\tilde{c}} P(\mathcal{B}_c^{\mathcal{C}}) \right) \prod_{n=1}^{N} P(\pi_n) \prod_{k=1}^{\mathcal{K}_{\mathcal{C}(n)}} \Psi(\mathcal{D}_n^{\pi_n}[(b_{\mathcal{C}(n),k-1} + 1) : b_{\mathcal{C}(n),k}]) \\ &= \left(\prod_{n=1}^{N} P(\mathcal{B}_n) \right) \prod_{n=1}^{N} P(\pi_n) \prod_{k=1}^{\mathcal{K}_n} \Psi(\mathcal{D}_n^{\pi_n}[(b_{n,k-1} + 1) : b_{n,k}]) \\ &= \prod_{n=1}^{N} P(\pi_n) P(\mathcal{B}_n) \prod_{k=1}^{\mathcal{K}_n} \Psi(\mathcal{D}_n^{\pi_n}[(b_{n,k-1} + 1) : b_{n,k}]) \end{aligned}$$

On the other hand, for a clustering function $\mathcal{C}(.)$ that clusters all nodes into one single cluster, $\tilde{c} = 1$ and $\mathcal{C}(n) = 1$ ($n = 1, \ldots, N$), there is only one single set of changepoints, $\mathcal{B}_1^{\mathcal{C}} = \{b_{1,1}, \ldots, b_{1,\mathcal{K}_1 - 1}\}$, that applies to all nodes, and it can be

[45] Rather than "penalizing" the changepoints for each individual node X_n ($n = 1, \ldots, N$) independently via $P(\mathcal{B}_n)$, defined in Eq. (96), even if there are nodes X_i and X_j ($i \neq j$) with exactly identical changepoint sets, symbolically $\mathcal{B}_i = \mathcal{B}_j$.

defined $\mathcal{B} := \{b_1, \ldots, b_{\mathcal{K}-1}\}$ where $\mathcal{K} := \mathcal{K}_1$ and $b_k := b_{1,k}$ $(k = 1, \ldots, \mathcal{K}-1)$, so that $\mathcal{B} = \mathcal{B}_1^{\mathcal{C}}$ and Eq. (119) becomes:

$$\begin{aligned} P(\mathcal{G}, \mathbf{B}^{\mathcal{C}}, \mathcal{C}|\mathcal{D}) \quad &\propto \quad \left(\prod_{c=1}^{\tilde{c}} P(\mathcal{B}_c^{\mathcal{C}})\right) \prod_{n=1}^{N} P(\pi_n) \prod_{k=1}^{\mathcal{K}_{\mathcal{C}(n)}} \Psi(\mathcal{D}_n^{\pi_n}[(b_{\mathcal{C}(n),k-1}+1) : b_{\mathcal{C}(n),k}]) \\ &= P(\mathcal{B}_1^{\mathcal{C}}) \prod_{n=1}^{N} P(\pi_n) \prod_{k=1}^{\mathcal{K}_1} \Psi(\mathcal{D}_n^{\pi_n}[(b_{1,k-1}+1) : b_{1,k}]) \\ &= P(\mathcal{B}) \prod_{n=1}^{N} P(\pi_n) \prod_{k=1}^{\mathcal{K}} \Psi(\mathcal{D}_n^{\pi_n}[(b_{k-1}+1) : b_k]) \\ &= P(\mathcal{B}) \left(\prod_{n=1}^{N} P(\pi_n)\right) \left(\prod_{n=1}^{N} \prod_{k=1}^{\mathcal{K}} \Psi(\mathcal{D}_n^{\pi_n}[(b_{k-1}+1) : b_k])\right) \\ &= P(\mathcal{B}) P(\mathcal{G}) \prod_{k=1}^{\mathcal{K}} \prod_{n=1}^{N} \Psi(\mathcal{D}_n^{\pi_n}[(b_{k-1}+1) : b_k]) \end{aligned}$$

As the point process prior on the changepoint locations yields a one-to-one mapping between changepoint sets, $\mathcal{B} = \{b_1, \ldots, b_{\mathcal{K}}\}$, and allocation vectors, $\vec{\mathcal{V}}$, symbolically: $\vec{\mathcal{V}}(t) = k \Leftrightarrow b_{k-1} < t \leq b_k$ $(t = 2, \ldots, m$; and $k = 1, \ldots, \mathcal{K}_c)$, this relationship can also be formulated in terms of the allocation vector, $\vec{\mathcal{V}}$:

$$P(\mathcal{G}, \mathbf{B}^{\mathcal{C}}, \mathcal{C}|\mathcal{D}) = P(\vec{\mathcal{V}}) P(\mathcal{G}) \prod_{k=1}^{\mathcal{K}} \prod_{n=1}^{N} \Psi(\mathcal{D}_n^{\pi_n,(\vec{\mathcal{V}},k)}) \tag{120}$$

where $\vec{\mathcal{V}} = \mathcal{V}(\mathcal{B})$. Eq. (120) is identical to Eq. (69) from Subsection 3.3.3 except that the original combination of a Poisson prior, $P(\mathcal{K})$, and a changepoint process prior for $P(\vec{\mathcal{V}}|\mathcal{K})$ of the original BGM_D model has been substituted for a simpler point process prior on the changepoint set, $\mathcal{B}$, which indirectly specifies the allocation vector, $\vec{\mathcal{V}} = \mathcal{V}(\mathcal{B})$, and the number of components, $\mathcal{K} = |\mathcal{B}| + 1$, in the regularized cpBGe model.
Consequently, the novel regularized cpBGe model can actually be seen as a generalized non-homogeneous Bayesian network model that subsumes both models BGM_D and cpBGe as limiting cases. It corresponds to a slightly modified BGM_D model (see Subsection 3.3.3) for $\tilde{c} = 1$, and it corresponds to the improved cpBGe model from Subsection 3.3.5 for $\tilde{c} = N$. The model complexity tuning in between these two extremes ($\tilde{c} = 1$ and $\tilde{c} = N$) is accomplished by Bayesian clustering and information sharing among nodes.

Inference:
Sampling from the posterior distribution of the regularized cpBGe model, given in Eq. (119), can be done by iteratively sampling from the following three "full) conditional" distributions: New graphs, $\mathcal{G}_\star$, can be sampled from $P(\mathcal{G}_\star|\mathcal{C}, \mathbf{B}^{\mathcal{C}}, \mathcal{D})$ $(n = 1, \ldots, N)$, new systems of (cluster-specific) changepoint sets, $\mathbf{B}_\star^{\mathcal{C}}$, can be sampled from $P(\mathbf{B}_\star^{\mathcal{C}}|\mathcal{G}, \mathcal{C}, \mathcal{D})$ $(c = 1, \ldots, \tilde{c})$, and a Reversible Jump Markov Chain Monte Carlo (RJMCMC) sampling scheme can be employed for sampling new cluster formations, $\mathcal{C}_\star$, along with new systems of changepoint sets, $\mathbf{B}_\star^{\tilde{\mathcal{C}}}$, from $P(\mathcal{C}_\star, \mathbf{B}_\star^{\tilde{\mathcal{C}}}|\mathcal{G}, \mathcal{C}, \mathbf{B}^{\mathcal{C}}, \mathcal{D})$ where $\mathcal{C}$ is the current cluster formation and $\mathbf{B}^{\mathcal{C}}$ is the current system of changepoints sets.

In this section these three sampling steps 1)-3) will be described in more detail.

Step 1: Sampling a new graph, $\mathcal{G}^\star$, conditional on the current cluster formation, $\mathcal{C}$, and the current system of cluster-specific changepoints, $\mathbf{B}^\mathcal{C}$:
Each graph, $\mathcal{G}$, is completely specified by the parent sets, π_n, of its nodes X_n ($n = 1, \ldots, N$), symbolically $\mathcal{G} = \{\pi_1, \ldots, \pi_N\}$. Hence, a new graph, $\mathcal{G}_\star$, can be sampled by sampling new parent sets $\pi_n^\star$ for each node, X_n, from

$$P(\pi_n^\star | \mathcal{B}^\mathcal{C}_{\mathcal{C}(n)}, \mathcal{D}) = \frac{P(\pi_n^\star) \prod_{k=1}^{\mathcal{K}_{\mathcal{C}(n)}} \Psi(\mathcal{D}_n^{\pi_n^\star}[(b_{\mathcal{C}(n),k-1}+1) : b_{\mathcal{C}(n),k}])}{\sum_{\pi_n : |\pi_n| \leq \mathcal{F}} P(\pi_n) \prod_{k=1}^{\mathcal{K}_{\mathcal{C}(n)}} \Psi(\mathcal{D}_n^{\pi_n}[(b_{\mathcal{C}(n),k-1}+1) : b_{\mathcal{C}(n),k}])} \quad (121)$$

where $\mathcal{B}^\mathcal{C}_{\mathcal{C}(n)} = \{b_{\mathcal{C}(n),1}, \ldots, b_{\mathcal{C}(n),\mathcal{K}_{\mathcal{C}(n)}-1}\}$, $\mathcal{K}_{\mathcal{C}(n)} = |\mathcal{B}^\mathcal{C}_{\mathcal{C}(n)}| + 1$, $\mathcal{F}$ is the fan-in restriction on the cardinality of the parent sets, and the $\Psi(.)$ terms have been specified in Eq. (100) and can be computed in closed form. $\mathcal{C}(n)$ is the cluster to which node X_n is allocated, so that Eq. (121) is identical to Eq. (105) except that the node-specific changepoint set in Eq. (105) has been substituted for the cluster-specific changepoint set $\mathcal{B}^\mathcal{C}_{\mathcal{C}(n)}$. The new parent sets, $\pi_n^\star$ ($n = 1, \ldots, N$), can be sampled independently from Eq. (121), and the composed graph, $\mathcal{G}_\star = \{\pi_1^\star, \ldots, \pi_N^\star\}$, has effectively been sampled from $P(\mathcal{G}^\star | \mathcal{C}, \mathbf{B}^\mathcal{C}, \mathcal{D})$.

Step 2: Sampling a new system of changepoint sets, $\mathbf{B}^\mathcal{C}_\star$, conditional on the current graph, $\mathcal{G}$, and the current cluster formation, $\mathcal{C}$:
Each system of changepoint sets, $\mathbf{B}^\mathcal{C}$, consists of cluster-specific changepoint sets $\mathcal{B}^\mathcal{C}_c$, symbolically $\mathbf{B}^\mathcal{C} = \{\mathcal{B}^\mathcal{C}_1, \ldots, \mathcal{B}^\mathcal{C}_{\tilde{c}}\}$, where $\tilde{c}$ is the number of non-empty node clusters induced by the cluster formation $\mathcal{C}$. Hence, a new system of changepoint sets, $\mathbf{B}^\mathcal{C}_\star = \{\mathcal{B}^\mathcal{C}_{1,\star}, \ldots, \mathcal{B}^\mathcal{C}_{\tilde{c},\star}\}$, can be sampled by independently sampling a new cluster-specific changepoint set $\mathcal{B}^\mathcal{C}_{c,\star} = \{b^\star_{c,1}, \ldots, b^\star_{c,\mathcal{K}^\star_c - 1}\}$ for each cluster $c = 1, \ldots, \tilde{c}$ from

$$P(\mathcal{B}^\mathcal{C}_{c,\star} | \mathcal{G}, \mathcal{C}, \mathcal{D}) = \frac{P(\mathcal{B}^\mathcal{C}_{c,\star}) \prod_{n:\mathcal{C}(n)=c} \left(\prod_{k=1}^{\mathcal{K}^\star_c} \Psi(\mathcal{D}_n^{\pi_n}[(b^\star_{c,k-1}+1) : b^\star_{c,k}]) \right)}{\sum_{\mathcal{B}^\mathcal{C}_c} \left\{ P(\mathcal{B}^\mathcal{C}_c) \prod_{n:\mathcal{C}(n)=c} \left(\prod_{k=1}^{\mathcal{K}_c} \Psi(\mathcal{D}_n^{\pi_n}[(b_{c,k-1}+1) : b_{c,k}]) \right) \right\}} \quad (122)$$

where the sum is over all valid changepoint sets $\mathcal{B}^\mathcal{C}_c = \{b_{c,1}, \ldots, b_{c,\mathcal{K}_c - 1}\}$ that divide the data into $\mathcal{K}_c = 1, \ldots, m-1$ disjunct segments, the prior probabilities $P(\mathcal{B}^\mathcal{C}_c)$ can be computed with Eq. (96), and the $\Psi(.)$ terms were specified in Eq. (100).
For each cluster c the new changepoint set, $\mathcal{B}^\mathcal{C}_{c,\star}$, can be sampled according to Eq. (122) with a slightly modified version of the dynamic programming scheme that was developed for the improved cpBGe model in Subsection 3.3.5. It just has to be taken into account that the changepoints are sampled for a cluster of nodes rather than one single node, so that the new changepoint set, $\mathcal{B}^\mathcal{C}_{c,\star} = \{b^\star_{c,1}, \ldots, b^\star_{c,\mathcal{K}^\star_c - 1}\}$, is common to all nodes in cluster c, $\{X_n : \mathcal{C}(n) = c\}$. Different from Subsection 3.3.5, let $Q(t|c, \mathcal{G}, \mathcal{C})$ therefore denote the probability of the observations for the nodes in the c-th cluster, $\{\mathcal{D}_{n,t:m} | n : \mathcal{C}(n) = c\}$, conditional on the corresponding realizations of their parent sets, $\{\mathcal{D}_{\pi_n,(t-1):(m-1)} | n : \mathcal{C}(n) = c\}$, *and* a changepoint $b^\dagger$ at time point $t-1$ ($t = 2, \ldots, m$). For the regularized cpBGe model, the node-specific terms $\Psi(\mathcal{D}_n^{\pi_n}[t : s])$ in Eqns. (109-110) just have to be substituted for (cluster-specific) products $\Psi^\mathcal{C}(.)$ of the (node-specific) $\Psi(.)$ terms:

$$\Psi^\mathcal{C}(\mathcal{D}_c^\mathcal{G}[t : s]) := \prod_{n:\mathcal{C}(n)=c} \Psi(\mathcal{D}_n^{\pi_n}[t : s]) \quad (123)$$

where $\mathcal{D}_c^\mathcal{G}[t : s])$ is the data subset $\{(\mathcal{D}_{n,i}, \mathcal{D}_{\pi_n,i-1}) : s \leq i \leq t, \mathcal{C}(n) = c\}$ consisting of the adjacent observations $\mathcal{D}_n^{\pi_n}[t : s]$ of all nodes in cluster c . The recursion for the

regularized cpBGe model is then for $t = 3, \ldots, m-1$ given by:

$$\begin{aligned} Q(t|c,\mathcal{G},\mathcal{C}) &= \left(\sum_{s=t}^{m-1} \Psi^{\mathcal{C}}(\mathcal{D}_c^{\mathcal{G}}[t:s]) Q(s+1|c,\mathcal{G},\mathcal{C}) g(s+1-t)\right) \\ &\quad + \Psi^{\mathcal{C}}(\mathcal{D}_c^{\mathcal{G}}[t:m])(1-G(m-t)) \end{aligned}$$

and

$$\begin{aligned} Q(2|c,\mathcal{G},\mathcal{C}) &= \left(\sum_{s=2}^{m-1} \{\Psi^{\mathcal{C}}(\mathcal{D}_c^{\mathcal{G}}[2:s])\} Q(s+1|c,\mathcal{G},\mathcal{C}) g_0(s-1)\right) \\ &\quad + \Psi^{\mathcal{C}}(\mathcal{D}_c^{\mathcal{G}}[2:m])(1-G_0(m-2)) \end{aligned}$$

where $Q(m|c,\mathcal{G},\mathcal{C}) = \Psi^{\mathcal{C}}(\mathcal{D}_c^{\mathcal{G}}[m:m])$ was defined as a product in Eq. (123), whose factors are $\Psi(.)$ terms that can be computed in closed-form with Eq. (100), so that $Q(m|c,\mathcal{G},\mathcal{C})$ can be used for initialization.
For each cluster, c, the posterior distribution of the first changepoint, $b_{c,1}^{\star}$, given the graph, $\mathcal{G}$, is given by:

$$P(b_{c,1}^{\star} = t|\mathcal{G},\mathcal{C},\mathcal{D}) = \Psi^{\mathcal{C}}(\mathcal{D}_c^{\mathcal{G}}[2:t]) Q(t+1|c,\mathcal{G},\mathcal{C}) \frac{g_0(t)}{Q(2|c,\mathcal{G},\mathcal{C})} \tag{124}$$

for $t = 2, \ldots, m-1$ and the probability of no changepoint is given by:

$$P(\mathcal{K}_c^{\star} = 1|\mathcal{G},\mathcal{C},\mathcal{D}) = \Psi^{\mathcal{C}}(\mathcal{D}_c^{\mathcal{G}}[2:m]) \frac{1-G_0(m-2)}{Q(2|c,\mathcal{G},\mathcal{C})} \tag{125}$$

where $G_0(.)$ was defined in Eq. (97).
The posterior distribution of the j-th changepoint, $b_{c,j}^{\star}$, for cluster c given the graph, $\mathcal{G}$, and the previous changepoint $b_{c,j-1}^{\star} = s$ is:

$$\begin{aligned} P_t &:= P(b_{c,j}^{\star} = t|b_{c,j-1}^{\star} = s,\mathcal{G},\mathcal{C},\mathcal{D}) \\ &= \Psi^{\mathcal{C}}(\mathcal{D}_c^{\mathcal{G}}[(s+1):t]) Q(t+1|c,\mathcal{G},\mathcal{C}) \frac{g(t-s)}{Q(s+1|c,\mathcal{G},\mathcal{C})} \end{aligned}$$

for $t = s+1, \ldots, m-1$. Therefore, given a changepoint at $b_{c,j-1}^{\star} = s$, the location of the next changepoint can be sampled from the discrete mass probability distribution $[P_{s+1}, \ldots, P_{m-1}, P_{\nabla(s)}]$ where $P_{\nabla(s)}$ is the probability for no further changepoints:

$$P_{\nabla(s)} := \Psi^{\mathcal{C}}(\mathcal{D}_c^{\mathcal{G}}[(s+1):m]) \frac{1-G_0(m-s-1)}{Q(s+1|c,\mathcal{G},\mathcal{C})} \tag{126}$$

Sampling sequentially from $[P_{s+1}, \ldots, P_{m-1}, P_{\nabla(s)}]$ until there is no further changepoint, yields the new changepoint set, $\mathcal{B}_{c,\star}^{\mathcal{C}} = \{b_{c,1}^{\star}, \ldots, b_{c,k-1}^{\star}\}$.
With regard to the third sampling step, described below, it is important to note that $Q(2|c,\mathcal{G},\mathcal{C})$ corresponds to the denominator in Eq. (122):

$$\begin{aligned} Q(2|c,\mathcal{G},\mathcal{C}) &= \sum_{\mathcal{B}_c^{\mathcal{C}}} P(\mathcal{B}_c^{\mathcal{C}}) \prod_{n:\mathcal{C}(n)=c} \left(\prod_{k=1}^{\mathcal{K}_c} \Psi(\mathcal{D}_n^{\pi_n}[(b_{c,k-1}+1):b_{c,k}])\right) \\ &= \sum_{\mathcal{B}_c^{\mathcal{C}}} P(\mathcal{B}_c^{\mathcal{C}}) \prod_{k=1}^{\mathcal{K}_c} \Psi^{\mathcal{C}}(\mathcal{D}_c^{\mathcal{G}}[(b_{c,k-1}+1):b_{c,k}]) \end{aligned} \tag{127}$$

where the sum is over all valid changepoint sets, $\mathcal{B}_c^{\mathcal{C}} = \{b_{c,1}, \ldots, b_{c,\mathcal{K}_c-1}\}$, that divide the data into $\mathcal{K}_c = 1, \ldots, m-1$ disjunct segments. Inserting Eq. (127) and Eq. (123) into Eq. (122) yields the compact representation:

$$P(\mathcal{B}_{c,\star}^{\mathcal{C}}|\mathcal{G},\mathcal{C},\mathcal{D}) = \frac{P(\mathcal{B}_{c,\star}^{\mathcal{C}}) \prod_{k=1}^{\mathcal{K}_c^{\star}} \Psi^{\mathcal{C}}(\mathcal{D}_c^{\mathcal{G}}[(b_{c,k-1}^{\star}+1):b_{c,k}^{\star}])}{Q(2|c,\mathcal{G},\mathcal{C})} \tag{128}$$

where $\mathcal{B}^{\mathcal{C}}_{c,\star} = \{b^{\star}_{c,1}, \ldots, b^{\star}_{c,\mathcal{K}^{\star}_c - 1}\}$. By introducing the new definition:

$$\begin{aligned} \Psi^{\mathcal{C}}(\mathcal{D}^{\mathcal{G}}_c[\mathcal{B}^{\mathcal{C}}_c]) &:= \prod_{k=1}^{\mathcal{K}_c} \Psi^{\mathcal{C}}(\mathcal{D}^{\mathcal{G}}_c[(b_{c,k-1}+1) : b_{c,k}]) \\ &= \prod_{k=1}^{\mathcal{K}_c} \prod_{n:\mathcal{C}(n)=c} \Psi(\mathcal{D}^{\pi_n}_n[(b_{c,k-1}+1) : b_{c,k}]) \end{aligned} \tag{129}$$

the conditional distribution of $\mathcal{B}^{\mathcal{C}}_{c,\star}$ from Eq. (128) and the likelihood of the regularized cpBGe model, given in Eq. (118), can be re-written more compactly:

$$P(\mathcal{B}^{\mathcal{C}}_{c,\star}|\mathcal{G},\mathcal{C},\mathcal{D}) = \frac{P(\mathcal{B}^{\mathcal{C}}_{c,\star})\Psi^{\mathcal{C}}(\mathcal{D}^{\mathcal{G}}_c[\mathcal{B}^{\mathcal{C}}_{c,\star}])}{Q(2|c,\mathcal{G},\mathcal{C})} \tag{130}$$

$$P(\mathcal{D}|\mathcal{C},\mathbf{B}^{\mathcal{C}},\mathcal{G}) = \prod_{c=1}^{\tilde{c}} \Psi^{\mathcal{C}}(\mathcal{D}^{\mathcal{G}}_c[\mathcal{B}^{\mathcal{C}}_c]) \tag{131}$$

Step 3: Sampling a new cluster formation, $\mathcal{C}^{\star}$, along with a new system of changepoint sets, $\mathbf{B}^{\mathcal{C}^{\star}}_{\star}$, by a sequence of Metropolis Hastings steps:
For the third step, sampling from $P(\mathcal{C}^{\star}, \mathbf{B}^{\mathcal{C}^{\star}}_{\star}|\mathcal{C},\mathbf{B}^{\mathcal{C}}\mathcal{G},\mathcal{D})$, a RJMCMC sampling scheme based on a sequence of cluster birth (b), death (d), and re-clustering (r) moves can be used. Sequentially, in each Metropolis Hastings step of the sequence, first, the move type is chosen randomly; e.g. from a uniform distribution: $p_{(b)} = p_{(d)} = p_{(r)} = 1/3$.

(i) Let there be $\tilde{c} < N$ non-empty clusters, induced by the current cluster formation $\mathcal{C}$. In a cluster birth move (b) a cluster c_1 that contains at least 2 nodes is randomly selected, and then one of those n_{c_1} nodes in cluster c_1 is randomly chosen. The cluster birth move tries to re-cluster this node from the c_1-th cluster to a new cluster $\tilde{c}+1$. Let $\mathcal{C}^{\star}$ denote the new cluster formation thus obtained. For the clusters c_1 and $\tilde{c}+1$, implied by the new cluster formation, $\mathcal{C}^{\star}$, new changepoints sets, $\mathcal{B}^{\mathcal{C}^{\star}}_{c_1,\star}$ and $\mathcal{B}^{\mathcal{C}^{\star}}_{\tilde{c}+1,\star}$, are sampled from the distributions $P(\mathcal{B}^{\mathcal{C}^{\star}}_{c_1,\star}|\mathcal{G},\mathcal{C}^{\star},\mathcal{D})$ and $P(\mathcal{B}^{\mathcal{C}^{\star}}_{\tilde{c}+1,\star}|\mathcal{G},\mathcal{C}^{\star},\mathcal{D})$, defined in Eq. (130), with Fearnhead's dynamic programming scheme; see **Step 2** for details.

(ii) Let there be $\tilde{c} > 1$ non-empty clusters, induced by the current cluster formation $\mathcal{C}$, including at least one cluster that contains only one single node. In a cluster death move (d) one of the clusters that contain only one single node is selected, and this single node is re-allocated to one of the other non-empty clusters, chosen randomly. The first selected cluster c_1 disappears and cluster c_2 absorbs the node from cluster c_1. Let $\mathcal{C}^{\star}$ denote the new cluster formation thus obtained. For the absorbing cluster c_2 a new changepoint set $\mathcal{B}^{\mathcal{C}^{\star}}_{c_2,\star}$ is sampled from $P(\mathcal{B}^{\mathcal{C}^{\star}}_{c_2,\star}|\mathcal{G},\mathcal{C}^{\star},\mathcal{D})$, defined in Eq. (130), with Fearnhead's dynamic programming scheme; see **Step 2** for details.

(iii) Let there be $1 < \tilde{c} < N$ non-empty clusters induced by the current cluster formation $\mathcal{C}$. In a re-clustering move two clusters c_1 and c_2 ($c_1 \neq c_2$) are chosen as follows. First, cluster c_1 is randomly selected among those clusters that contain at least 2 nodes. Next, cluster c_2 is randomly selected among the $\tilde{c}-1$ remaining clusters. Afterwards, one of the nodes from cluster c_1 is randomly chosen and re-allocated to cluster c_2. Let $\mathcal{C}^{\star}$ denote the new cluster formation thus obtained. Since cluster c_1 contains at least 2 nodes, this does not affect the number of clusters $\tilde{c}$.

For both clusters, c_1 and c_2, new changepoint sets, $\mathcal{B}_{c_1,\star}^{\mathcal{C}^\star}$ and $\mathcal{B}_{c_2,\star}^{\mathcal{C}^\star}$, are sampled from $P(\mathcal{B}_{c_1,\star}^{\mathcal{C}^\star}|\mathcal{G},\mathcal{C}^\star,\mathcal{D})$ and $P(\mathcal{B}_{c_2,\star}^{\mathcal{C}^\star}|\mathcal{G},\mathcal{C}^\star,\mathcal{D})$, defined in Eq. (130), with Fearnhead's dynamic programming scheme; see **Step 2** for details.

The acceptance probabilities of these three RJMCMC moves (i-iii), described above, are given by the product of the likelihood ratio (LR), the prior ratio (PR), the inverse proposal probability ratio or Hastings factor (HR), and the Jacobian (J) in the standard way (Green (1995)): $A = min\{1, R\}$, where

$$R = LR \times PR \times HR \times J \tag{132}$$

Since this is a discrete problem, the Jacobian J is 1 for all three move types, and R reduces to $R = LR \times PR \times HR$. In this thesis only the mathematical details of birth moves (b) are given and it is referred to the original publication (Grzegorczyk and Husmeier, 2011a) for cluster death and cluster re-allocation moves.
For each cluster birth move (b), symbolically $(\mathcal{C}, \mathbf{B}^{\mathcal{C}}) \rightarrow (\mathcal{C}^\star, \mathbf{B}_\star^{\mathcal{C}^\star})$, there is a unique complementary cluster death move (d), symbolically $(\mathcal{C}^\star, \mathbf{B}_\star^{\mathcal{C}^\star}) \rightarrow (\mathcal{C}, \mathbf{B}^{\mathcal{C}})$, where $\mathbf{B}^{\mathcal{C}} = \{\mathcal{B}_1, \ldots, \mathcal{B}_{\tilde{c}}\}$, $\mathbf{B}_\star^{\mathcal{C}^\star} = \{\mathcal{B}_{1,\star}, \ldots, \mathcal{B}_{\tilde{c}+1,\star}\}$, $\mathcal{B}_c = \{b_{c,1}, \ldots, b_{c,\mathcal{K}_c}\}$ $(c = 1, \ldots, \tilde{c})$, $\mathcal{B}_{c,\star} = \{b_{c,1}^\star, \ldots, b_{c,\mathcal{K}_c^\star}^\star\}$ $(c = 1, \ldots, \tilde{c}+1)$, and $\mathcal{B}_{c,\star} = \mathcal{B}_c$ for $c \in \{1, \ldots, c_1 - 1, c_1 + 1, \ldots, \tilde{c}\}$. Thus, with Eq. (131) it follows for the likelihood ratio:

$$\begin{aligned} LR_{(b)} &= \frac{P(\mathcal{D}|\mathcal{G}, \mathbf{B}_\star^{\mathcal{C}^\star}, \mathcal{C}^\star)}{P(\mathcal{D}|\mathcal{G}, \mathbf{B}^{\mathcal{C}}, \mathcal{C})} = \frac{\prod_{c=1}^{\tilde{c}+1} \Psi^{\mathcal{C}^\star}(\mathcal{D}_c^{\mathcal{G}}[\mathcal{B}_{c,\star}^{\mathcal{C}^\star}])}{\prod_{c=1}^{\tilde{c}} \Psi^{\mathcal{C}}(\mathcal{D}_c^{\mathcal{G}}[\mathcal{B}_c^{\mathcal{C}}])} \\ &= \frac{\Psi^{\mathcal{C}^\star}(\mathcal{D}_{c_1}^{\mathcal{G}}[\mathcal{B}_{c_1,\star}^{\mathcal{C}^\star}]) \cdot \Psi^{\mathcal{C}^\star}(\mathcal{D}_{\tilde{c}+1}^{\mathcal{G}}[\mathcal{B}_{\tilde{c}+1,\star}^{\mathcal{C}^\star}])}{\Psi^{\mathcal{C}}(\mathcal{D}_{c_1}^{\mathcal{G}}[\mathcal{B}_{c_1}^{\mathcal{C}}])} \end{aligned}$$

where the $\Psi^{\mathcal{C}}(\mathcal{D}_c^{\mathcal{G}}[\mathcal{B}_c^{\mathcal{C}}])$ terms were specified in Eq. (129) and can be computed in closed form.
The prior ratio for birth moves (b) is given by:

$$PR_{(b)} = \frac{P(\mathcal{C}^\star)P(\mathbf{B}_\star^{\mathcal{C}^\star})}{P(\mathcal{C})P(\mathbf{B}^{\mathcal{C}})} = \frac{P(\mathbf{B}_\star^{\mathcal{C}^\star})}{P(\mathbf{B}^{\mathcal{C}})} = \frac{P(\mathcal{B}_{c_1,\star}^{\mathcal{C}^\star})P(\mathcal{B}_{\tilde{c}+1,\star}^{\mathcal{C}^\star})}{P(\mathcal{B}_{c_1}^{\mathcal{C}})} \tag{133}$$

since the prior distribution on the cluster formations is assumed to be uniform, $P(\mathcal{C}^\star) = P(\mathcal{C})$, and $\mathcal{B}_c^{\mathcal{C}} = \mathcal{B}_{c,\star}^{\mathcal{C}^\star}$ for $c = 1, \ldots, c_1 - 1, c_1 + 1, \ldots, \tilde{c}$.
The Hastings-Ratio depends on the designs of the birth and the complementary death move, which were both described above: With probability $p_{(b)}$ a birth move is performed, and a cluster c_1 out of the set of $c^\dagger$ clusters that contain at least 2 nodes is selected with probability $1/c^\dagger$. Afterwards, one of the n_{c_1} nodes in cluster c_1 is selected with probability $1/n_{c_1}$ and moved to a new cluster $\tilde{c}+1$. This yields a new cluster formation $\mathcal{C}^\star$. Finally, new changepoint sets, $\mathcal{B}_{c_1,\star}^{\mathcal{C}^\star}$ and $\mathcal{B}_{\tilde{c}+1,\star}^{\mathcal{C}^\star}$, are sampled for both clusters c_1 and $\tilde{c}+1$. In the complementary death move, which is performed with probability $p_{(d)}$, the new 'born' cluster $\tilde{c}+1$ has to be selected out of the $c^\ddagger$ clusters, induced by the new formation $\mathcal{C}^\star$, that contain only one single node. The node $\tilde{c}+1$ is selected with probability $1/c^\ddagger$, and as the node in cluster $\tilde{c}+1$ has to be moved back to cluster c_1, the cluster c_1 has to be selected as absorbing cluster. The cluster c_1 is selected with probability $1/\tilde{c}$ out of the set of all $\tilde{c}$ clusters that were originally induced by $\mathcal{C}$. Finally, the original changepoint set $\mathcal{B}_{c_1}^{\mathcal{C}}$ has to be re-sampled for the absorbing cluster c_1. The Hastings-Ratio is therefore given by:

$$HR_{(b)} = \frac{p_{(d)}(1/c^\ddagger)(1/\tilde{c})P(\mathcal{B}_{c_1}^{\mathcal{C}}|\mathcal{G},\mathcal{D},\mathcal{C})}{p_{(b)}(1/c^\dagger)(1/n_{c_1})P(\mathcal{B}_{\tilde{c}+1,\star}^{\mathcal{C}^\star}|\mathcal{G},\mathcal{D},\mathcal{C}^\star)P(\mathcal{B}_{c_1,\star}^{\mathcal{C}^\star}|\mathcal{G},\mathcal{D},\mathcal{C}^\star)} \tag{134}$$

where $c^\dagger$ is the number of clusters induced by $\mathcal{C}$ with at least two nodes, n_{c_1} is the number of nodes in the c_1-th cluster (that was selected from those $c^\dagger$ clusters), $c^\ddagger$

is the number of clusters induced by $\mathcal{C}^\star$ that contain only one single node, and $\tilde{c}$ is number of clusters induced by $\mathcal{C}$. The conditional distributions of cluster-specific changepoint sets were defined in Eq. (130). The factors $p_{(d)}$ and $p_{(b)}$ in $HR_{(b)}$ cancel out, since each move type was set equally likely: $p_{(d)} = p_{(b)} = 1/3$. Furthermore, by inserting Eq. (130), the Hastings ratio can be transformed to:

$$HR_{(b)} = \frac{c^\dagger n_{c_1}}{c^\ddagger \tilde{c}} \frac{P(\mathcal{B}^{\mathcal{C}}_{c_1}) \Psi^{\mathcal{C}}(\mathcal{D}^{\mathcal{G}}_{c_1}[\mathcal{B}^{\mathcal{C}}_{c_1}]) Q(2|c_1, \mathcal{G}, \mathcal{C}^\star) Q(2|\tilde{c}+1, \mathcal{G}, \mathcal{C}^\star)}{P(\mathcal{B}^{\mathcal{C}^\star}_{c_1,\star}) \Psi^{\mathcal{C}^\star}(\mathcal{D}^{\mathcal{G}}_{c_1}[\mathcal{B}^{\mathcal{C}^\star}_{c_1,\star}]) P(\mathcal{B}^{\mathcal{C}^\star}_{\tilde{c}+1,\star}) \Psi^{\mathcal{C}^\star}(\mathcal{D}^{\mathcal{G}}_{\tilde{c}+1}[\mathcal{B}^{\mathcal{C}^\star}_{\tilde{c}+1,\star}]) Q(2|c_1, \mathcal{G}, \mathcal{C})} \tag{135}$$

It follows that the product, $R_{(b)} = LR_{(b)} \times PR_{(b)} \times HR_{(b)}$, in Eq. (132) reduces to:

$$R_{(b)} = \frac{c^\dagger n_{c_1}}{c^\ddagger \tilde{c}} \frac{Q(2|c_1, \mathcal{G}, \mathcal{C}^\star) Q(2|\tilde{c}+1, \mathcal{G}, \mathcal{C}^\star)}{Q(2|c_1, \mathcal{G}, \mathcal{C})} \tag{136}$$

The term $R_{(b)}$ and the acceptance probability, $A_{(b)} = min\{1, R_{(b)}\}$, for birth moves can be computed efficiently, since the $Q(.)$ terms in Eq. (136) can be computed with the dynamic programming scheme of Fearnhead, described in **Step 2**.
The acceptance probabilities for cluster death (d) and re-clustering (r) moves from $(\mathcal{C}, \mathbf{B}^{\mathcal{C}})$ to $(\mathcal{C}^\star, \mathbf{B}^{\mathcal{C}^\star}_\star)$, whose designs were described above, can be derived analogously; see original publication for details (Grzegorczyk and Husmeier, 2011a).

Selected application(s):
The first application is a simulation study that compares the network reconstruction accuracies of the four Bayesian network models, described in this thesis. **(i)** the standard homogeneous dynamic Bayesian network from Subsection 2.5, **(ii)** the non-homogeneous BGM_D Bayesian network model with network-wide changepoints from Subsection 3.3.3, **(iii)** the non-homogeneous cpBGe Bayesian network model with node-specific changepoints from Subsection 3.3.5, and **(iv)** the non-homogeneous regularized cpBGe Bayesian network model with node-cluster-specific changepoints, described in this subsection. To ensure a fair cross-method comparison, the three non-homogeneous network models **(ii)-(iv)** were implemented with the point process prior, see Eqns. (97-99) in Subsection 3.3.5, and the Gibbs sampling/dynamic programming scheme, described in Subsection (3.3.5), was used for model inference.[46]
Therefore, and also to emphasize that the goal of the simulation study in Grzegorczyk and Husmeier (2011a) was to compare the generic concepts of *network-wide*, *gene-specific*, and *cluster-specific* changepoints rather than to compare the performance of the concrete model instantiations, it will be distinguished between the following classes of non-homogeneous models: A *class 1* non-homogeneous network model possesses network-wide changepoints (here: represented by the BGM_D model), a *class 2* model has gene-specific changepoints (here: represented by the cpBGe model), and a *regularized class 2* model regularizes between a *class 1* and a *class 2* model and possesses gene-cluster-specific changepoints (here: represented by the regularized cpBGe model). In the simulation study the network reconstruction accuracy was evaluated with the area under the precision-recall curve with larger values indicating a better performance, as explained in Subsection 2.7.
The RAF protein signalling transduction pathway, shown in panel (f) of Figure 11, plays a pivotal role in the mammalian immune response and has hence been widely studied in the literature (e.g. Sachs *et al.* (2005)). For the simulation study, synthetic network data were generated from a slightly modified version of the RAF-pathway, in which an extra self-feedback loop has been added to node 'PIP3':

[46] In consistency with the findings for the cpBGe model, reported in Subsection 3.3.6, the implementation of the point process prior for the distances between changepoints and the usage of the corresponding Gibbs sampling scheme turned out to improve convergence and mixing for the BGM_D model inference.

$PIP3(t+1) = \sqrt{1-\varepsilon^2}PIP3(t) + \varepsilon\phi_{PIP3}(t+1)$. For the study a moderate autocorrelation ($\varepsilon = 0.25$) was used. The realizations of the other ten nodes are linear combinations of the realizations of their parents at the preceding time points plus iid standard Gaussian $\mathcal{N}(0,1)$ distributed noise injections. E.g. for 'PIP2': $PIP2(t+1) = \beta_{PIP3}(t)PIP3(t) + \beta_{PLCG}(t)PLCG(t) + c_{PIP2}\phi_{PIP2}(t+1)$, where the variables $\phi_{\cdot}(.)$ are iid standard Gaussian distributed, and the coefficient, c_{PIP2}, can be used to vary the signal-to-noise ratio (SNR). The regression coefficients, β_{PIP3} and β_{PLCG}, were sampled from continuous uniform distributions on the interval $[0.5, 2]$ with a random sign, and time series of length $m = 21$ were generated. The noise level can be specified in terms of signal-to-noise ratios (SNRs). To this end, the standard deviation σ_{PIP2} of the input signals $\{\sigma(\beta_{PIP3}(t)PIP3(t) + \beta_{PLCG}(t)PLCG(t))|t = 1, \ldots, 21\}$ (before noise injections) can be estimated by exhaustive data simulation. Having estimated σ_{PIP2} from pre-simulated data, the coefficient c_{PIP2} can be computed as follows:

$$c_{PIP2} = \frac{\sigma_{PIP2}}{SNR} \tag{137}$$

where SNR is the desired signal-to-noise ratio.
In the simulation study it was distinguished between five different scenarios: **1)** homogeneous dynamic Bayesian network (DBN) data with regression coefficients that are constant in time, e.g. $\beta_{PIP3}(t) = \beta_{PIP3}$ and $\beta_{PLCG}(t) = \beta_{PLCG}$; **2)** non-homogeneous *class 1* DBN data where *all* regression coefficients of the domain are re-sampled after $t = 11$; **3)** non-homogeneous *class 2* DBN data with each node having 1-2 randomly chosen node-specific changepoints, where the corresponding regression coefficients are re-sampled; **4)** non-homogeneous regularized *class 2* data generated from a DBN where the coefficients of five nodes are re-sampled after $t = 11$, and the coefficients of the other 5 nodes are re-sampled twice independently, after $t = 8$ and after $t = 13$. In addition a fifth scenario was considered: **5)** non-homogeneous regularized data *without* any autocorrelation (no AC), i.e. $\varepsilon = 1$. Each of the five scenarios **1)-5)** was combined with two signal-to-noise ratios (SNR=3 and SNR=10), and 10 independent data instantiations were generated for each of the ten combinations, i.e. 100 data sets in total.
The network reconstruction accuracies were quantified with the mean area under the precision-recall curves (AUC) and the dependence on the hyperparameter p of the negative binomial point process prior of Eq. (98) was investigated. For the three non-homogeneous models **(ii)-(iv)** a low (high) value of the hyperparameter p implies a high (low) prior penalty for changepoints. The variation of p can therefore be interpreted as the variation of the prior knowledge about the number of underlying changepoints. Different from the non-homogeneous models **(ii)-(iv)**, the homogeneous Bayesian network model **(i)** does not allow for changepoints and its performance is therefore independent of the hyperparameter p. Figure 11 summarizes the empirical results of the simulation study.

1) Homogeneous data: Except for the highest setting of the hyperparameter p, the three non-homogeneous dynamic Bayesian network models never perform worse than the homogeneous DBN model. On the other hand for non-homogeneous data (see scenarios 2)-5)), the homogeneous model is inappropriate and performs substantially worse. The superiority of the homogeneous model for the combination $p = 0.1$ and SNR=3 demonstrates that the hyperparameter p must be selected carefully; for high p all three non-homogeneous models tend to infer spurious changepoints and therefore tend to over-fit the data.
2) Class 1 data: The *class 1* model and the proposed regularized *class 2* model perform equally well. Both outperform the *class 2* model, except for high values of p. Since a high value of the hyperparameter p implies a low prior penalty for changepoints, it appears that the *class 1* model with network-wide changepoints can

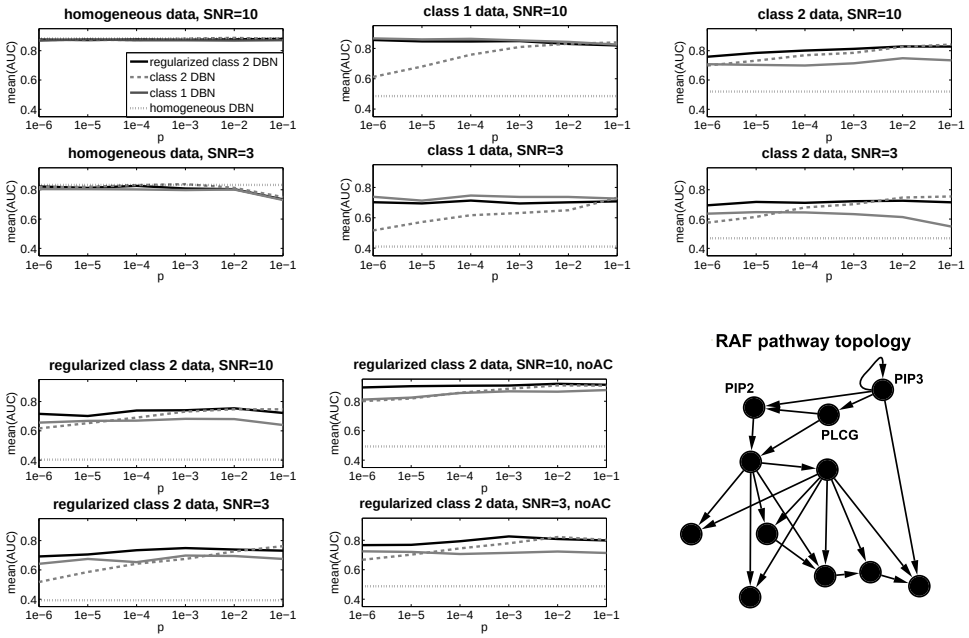


Figure 11: **Network reconstruction accuracy on synthetic data.** The figure shows the mean area under the precision-recall curves (AUC) in dependence on the hyperparameter p of the negative binomial point process prior of Eq. (98). For the RAF pathway (bottom right panel) five scenarios of non-homogeneity were implemented. For each scenario there is a panel for SNR=3 and SNR=10. The data sets were inferred using the following models each representing one particular class: **(i)** homogeneous model: the standard Gaussian DBN model from Subsection 2.4),**(ii)** the *class 1* BGM_D model from Subsection 3.3.3 in a slightly modified version, **(iii)** the *class 2* cpBGe model from Subsection 3.3.4, implement as described in Subsection 3.3.5, and **(iv)** the regularized *class 2* model described in this subsection. The mean Precision-Recall AUC scores were computed from 10 independent data instantiations.

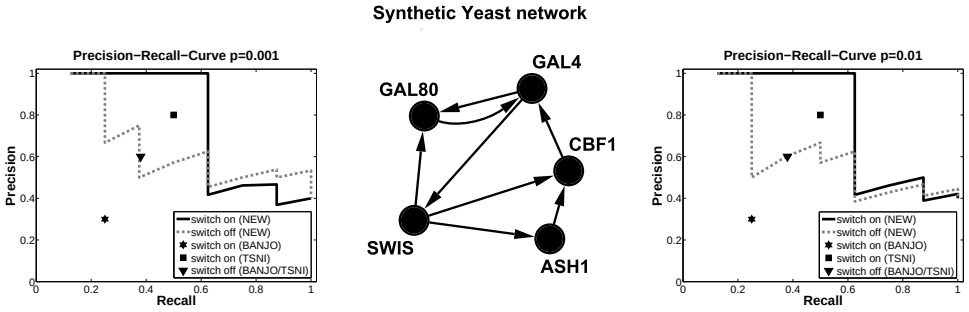


Figure 12: **Network reconstruction accuracy evaluated with synthetic biology.** The center panel shows the true gene regulatory network in *Saccharomyces cerevisiae*, designed in Cantone *et al.* (2009). The outer panels show the precision-recall curves for the proposed regularized *class 2* cpBGe model (NEW). Results were obtained for both experimental conditions: the "switch on" and the "switch off" time series described in Cantone *et al.* (2009). The symbols at fixed positions (triangle, star and square) mark the precision/recall results reported in Cantone *et al.* (2009) for two state-of-the-art network reconstruction methods: BANJO (conventional homogeneous DBN) and TSNI (ODE based approach).

approximate the node-specific changepoints by setting a higher number of network-wide changepoints.
3) Class 2 data: The *class 1* model cannot accommodate the node-specific changepoints and is outperformed by the proposed regularized *class 2* model (the "NEW" model). Interestingly, the latter also shows more stability than the *class 2* model with respect to a variation of the hyperparameter p, indicating increased robustness as a consequence of the node clustering.
4) Regularized class 2 data: The results are comparable to those for the *class 2* data. The *class 1* model is consistently inferior to the *class 2* model, and the *class 2* model is, once again, substantially more susceptible to a variation of p. The mean AUC values are – overall – lower than for the previous case, the *class 2* data. This seems to be a consequence of spurious interactions resulting from chance correlations.
5) Regularized class 2 data without autocorrelation: It can be seen that setting the autocorrelation of node $PIP3$ to zero ($\varepsilon = 1$, no AC), noticeably increases the mean AUC values compared to those obtained for scenario **4)**.
In summary, this study shows that the regularized *class 2* model, which implements cluster-specific changepoints and allows for information sharing method is always among the best-scoring Bayesian network models. The regularized cpBGe model shows more robustness than the competing DBN models both with respect to a variation of the type of data, and a variation of the prior knowledge, inherent in Eqns. (97-99) via the hyperparameter p.

To demonstrate that the regularized cpBGe model, described in this subsection, also performs well for real-world data it was also applied to gene expression data from a synthetically designed yeast strain.[47] Expression data for a synthetically generated network of five genes in *Saccharomyces cerevisiae* (yeast), depicted in Figure 12, were taken and made available from Cantone *et al.* (2009). Cantone *et al.* (2009) collected data from synthetically designed yeast cells grown with different carbon

[47]While *systems biology* aims to develop a formal understanding of biological processes via the development of quantitative mathematical models, *synthetic biology* aims to use such models to design unique biological circuits (synthetic networks) in the cell able to perform specific tasks. Conversely, data from synthetic biology can be utilized to assess the performance of models from systems biology.

sources: galactose ("switch on") or glucose ("switch off"). Quantitative real-time RT-PCR was used in intervals of 20 minutes up to 5 hours for the first, and in intervals of 10 minutes up to 3 hours for the second condition (see Cantone *et al.* (2009) for details). For the study, presented here, the raw data, available from Cantone *et al.* (2009), were standardized via a log and a z-score transformation.
Applying the regularized cpBGe model to the data yields marginal edge posterior probabilities, and for different thresholds on these probabilities the precision and the recall scores can be computed (see Subsection 2.7). Plotting the precision scores against the recall scores gives the precision-recall curve shown in Figure 12.[48] In agreement with Cantone *et al.* (2009) it was found that the "switch on" data are more informative than their "switch off" counterpart. The scores for two alternative network reconstruction methods, namely BANJO[49] and TSNI[50], which could be taken from Cantone *et al.* (2009), lie clearly and consistently below the "switch on" Precision Recall curve, for different choices of the changepoint process prior – defined by p in Eqns. (97-99). This suggests that the regularized cpBGe model, presented here, achieves a genuine and significant improvement over state-of-the-art schemes reported in the recent systems biology literature.

[48] Larger areas under the Precision Recall curve are indicative of a better reconstruction accuracy. See Subsection 2.7 and Davis and Goadrich (2006) for details.

[49] BANJO is the "Bayesian Network Inference with Java Objects" software that was implemented by the group of Alexander Hartemink (Duke University). BANJO provides various algorithms for static and dynamic Bayesian networks and has been developed over the years. BANJO employs the discrete BDe metric for scoring Bayesian networks, and a simulated annealing approach is used for finding the best scoring network. For example see Smith *et al.* (2006) for more details on BANJO.

[50] TSNI is the "Time-Series Network Identification" algorithm (Bansal *et al.* (2006) and Bansal and di Bernardo (2007)). The TSNI algorithm is based on ordinary differential equations; see Cantone *et al.* (2009) for a detailed description.

4 Discussion and outlook

In Subsection 3.3 several novel non-homogeneous dynamic Bayesian network models, that have been developed recently, were presented. Compared to standard classical (homogeneous) Bayesian networks these novel models improve the modeling flexibility, and for various real-world examples from the topic field of systems biology research it could be shown that the increased flexibility actually yields a higher network reconstruction accuracy.

Relaxing the homogeneity assumption in dynamic Bayesian networks is a recent research topic and various authors have proposed relaxing the homogeneity assumption by complementing the traditional homogeneous dynamic Bayesian network with a Bayesian multiple changepoint process (e.g. see Robinson and Hartemink (2009), Lèbre *et al.* (2010), and Grzegorczyk and Husmeier (2011b) among others). Each time series segment defined by two demarcating changepoints is associated with separate node-specific network parameters, and in this way the conditional probability distributions are allowed to vary from segment to segment. An attractive feature of this approach is that under certain regularity conditions, most notably parameter independence and conjugacy of the prior, the parameters can be integrated out in closed form in the likelihood. The inference task thus reduces to sampling the network structure as well as the number and location of changepoints from the posterior distribution, which can be effected with Reversible Jump Markov Chain Monte Carlo (RJMCMC) (Green, 1995), as in Robinson and Hartemink (2009) and Lèbre *et al.* (2010), or with dynamic programming (Fearnhead, 2006), as in Grzegorczyk and Husmeier (2011b).

In all these non-homogeneous dynamic Bayesian network models the marginal likelihood is computed from a typically uninformative parameter prior that is the same for all time series segments. These models ignore the fact that many systems, such as regulatory processes and signalling pathways in the cell, evolve gradually. For example, consider the cellular processes during an organism's development (morphogenesis) or its adaptation to changing environmental conditions. The assumption of a homogeneous process with constant parameters is obviously over-restrictive in that it fails to allow for the non-stationary nature of the processes. However, complete parameter independence is over-flexible in that it ignores the evolutionary aspect of developmental and adaptation processes. Given a regulatory network at a given time interval in an organism's life cycle, it is unrealistic to assume that at an adjacent time interval, nature has potentially reinvented the regulatory circuits from scratch. Instead, it is more realistic to assume that cognizance of the interaction strengths at the previous time interval provides prior knowledge about the interaction strengths at the next (adjacent) time interval. Currently, "*we*"[51] are developing another non-homogeneous dynamic Bayesian network model with node-specific changepoints in which the parameters associated with separate time series segments are *a priori* encouraged to be similar. Avoiding the fallacies of a Bayesian filter, a coupling hyperparameter that is shared among the segments is introduced and this hyperparameter is itself inferred from the data in a Bayesian sense. That is, the new approach allows for systematic information sharing between parameters associated with adjacent time series segments, and the overall strength of this coupling is controlled by a global hyperparameter. Simulation studies on non-homogeneous synthetic network data with systematically evolving network parameters have led to promising results. It could be shown that incorporating this information sharing scheme into models, such as the cpBGe model

[51] The manuscript in preparation, entitled "A Regularized Non-homogeneous Dynamic Bayesian Network for Applications in Systems and Synthetic Biology" by Marco Grzegorczyk and Dirk Husmeier, will be submitted very soon.

from Subsection 3.3.4 or the model from Lèbre *et al.* (2010), significantly improves the network reconstruction accuracy.
However, from a more global perspective, the collection of cellular data for systems biology is still associated with experimental efforts and financial costs, and thus, the available data sets tend to be sparse with respect to the experimental replicates (static steady state data) or lengths of the time series (dynamic data). E.g. gene expression data can be measured with modern biotechnologies, such as Microarray Gene Chips, and these technologies enable "wet-lab" researchers to measure thousands of genes simultaneously, but usually the number of measurements that are taken is very small relative to the number of genes that are measured. Consequently, even if the focus is only on a small subset of the measured genes, the regulatory network between these genes cannot be modeled in arbitrary detailedness. It is the sparseness of the available data sets that imposes an upper bound onto the detailedness that can be reached. In the context of Bayesian network methodology, networks/graphs are scored in light of the available data and following the Bayesian paradigm the posterior probability of a network serves as "scoring metric". The posterior probability of a graph is proportional to the marginal likelihood times the graph prior distribution, symbolically:

$$P(\mathcal{G}|\mathcal{D}) \propto P(\mathcal{D}|\mathcal{G})P(\mathcal{G}) \tag{138}$$

as explained in more detail in Section 2.1. Models, such as the BGM model from Subsection 3.3.2, the BGM_D model from Subsection 3.3.3, the cpBGe model from Subsection 3.3.4 and 3.3.5, or the regularized cpBGe model from Subsection 3.3.6 improve the modeling flexibility by introducing segmentations $\mathcal{S}$ of the data that allow for modeling non-linear and non-homogeneous interactions between the network nodes. Independently of whether the segmentations are induced by changepoints or free allocation mixture models, or whether segmentations are node-specific or network-wide, each of the expanded models is of the following form:

$$P(\mathcal{G},\mathcal{S}|\mathcal{D}) \propto P(\mathcal{D}|\mathcal{S},\mathcal{G})P(\mathcal{G})P(\mathcal{S}) \tag{139}$$

where $\mathcal{S}$ is the (network-)model-specific data segmentation, whose prior distribution is given by $P(\mathcal{S})$. These network-model expansions yield more flexible marginal likelihoods $P(\mathcal{D}|\mathcal{S},\mathcal{G})$ that for example can take non-homogenities into account, and the original posterior probability $P(\mathcal{G}|\mathcal{D})$ is substituted for the *marginal* graph posterior probability of the expanded model (NEW):

$$P_{NEW}(\mathcal{G}|\mathcal{D}) = \sum_{\mathcal{S}} P(\mathcal{G},\mathcal{S}|\mathcal{D}) \tag{140}$$

where the sum is over all valid data segmentations. Inserting Eq. (139) into Eq. (140) yields:

$$P_{NEW}(\mathcal{G}|\mathcal{D}) \propto P_{NEW}(\mathcal{D}|\mathcal{G})P(\mathcal{G}) \tag{141}$$

where

$$P_{NEW}(\mathcal{D}|\mathcal{G}) := \sum_{\mathcal{S}} P(\mathcal{D}|\mathcal{S},\mathcal{G})P(\mathcal{S}) \tag{142}$$

It can be seen from Eq. (141) that the more complex modeling frameworks do *not* change the essential form of the original Bayesian network posterior probability, given in Eq. (139). The only difference is that the modeling flexibility – inherent in the functional form of the marginal likelihood $P_{NEW}(\mathcal{D}|\mathcal{G})$ – is improved. Taking into account that the sparseness of the data imposes an upper bound on the flexibility that can be modeled, it is questionable whether the development of more flexible network models, which in the absence of sufficiently "rich" data sets cannot be learned properly, will

be useful for practical applications in the field of systems biology.
On the other hand, throughout this thesis the prior distribution on the graph structures, $P(\mathcal{G})$, was assumed to be a uniform distribution or a distribution that just penalizes graphs according to their complexities (usually quantified in the number of network edges). Even though uninformative priors can be justified in the absence of any genuine prior knowledge about the true underlying graph topology, for most applications in the field of systems biology there is a huge amount of available knowledge from other data sources, which should *not* be ignored. In particular, building more appropriate graph prior distributions from other "rich" data sources is likely to yield a more substantial improvement in the network reconstruction accuracy than developing more and more flexible Bayesian network models, which cannot be inferred properly from sparse data anyway. First approaches, which systematically exploit biological knowledge to improve the network reconstruction accuracy, have been proposed by various authors (Tamada *et al.* (2003), Tamada *et al.* (2005), Nariai *et al.* (2005), Imoto *et al.* (2006), and Werhli and Husmeier (2007)).[52] E.g. Werhli and Husmeier (2007) combine biological prior knowledge from the KEGG[53] data base with Markov Chain Monte Carlo (MCMC) sampling of Bayesian networks, and the authors demonstrate that the integration of this knowledge yields a significantly better network reconstruction accuracy for the RAF protein activation pathway (Sachs *et al.*, 2005) than Bayesian networks with uninformative graph prior distributions.
To reach the optimal network reconstruction accuracy, even the most flexible Bayesian network models have to be combined with appropriate graph prior distributions, e.g. extracted and generated from other sources of data, unless sufficiently large expression data sets are available for computational systems biology. It is the full exploitation of both aspects: (i) available biological prior knowledge and (ii) Bayesian modeling capacity that is required to ensure that Bayesian networks stay an important probabilistic Machine Learning method for the elucidation of regulatory network structures in future systems biology research.

[52] E.g. (i) knowledge about transcription factor binding motifs in promoter sequences, (ii) knowledge about known protein-protein interactions, (iii) knowledge about evolutionary information, or (iv) knowledge about known pathways from databases, such as the Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways database.

[53] The Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways database contains pathway information for metabolism and other cellular processes. Details on the KEGG data base can for example be found in Kanehisa (1997), Kanehisa and Goto (2000), and Kanehisa *et al.* (2006).

References

Akaike, H. (1983) Information measures and model selection. *Bull. Int. Stat. Inst.*, **50**, 277–290.

Andrieu, C. and Doucet, A. (1999) Joint Bayesian model selection and estimation of noisy sinusoids via reversible jump MCMC. *IEEE Transactions on Signal Processing*, **47**, 2667–2676.

Bansal, M. and di Bernardo, D. (2007) Inference of gene networks from temporal gene expression profiles. *IET Systems Biology*, **5**.

Bansal, M., Gatta, G. and di Bernardo, D. (2006) Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics*, **22**, 815–822.

Basso, K., Margolin, A., Stolovitzky, G., Klein, U., Dalla-Favera, R. and Califano, A. (2005) Reverse engineering of regulatory networks in human B cells. *Nature Genetics*, **37**, 382–390.

Beal, M., Falciani, F., Ghahramani, Z., Rangel, C. and Wild, D. (2005) A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*, **21**, 349–356.

Beinlich, I., Suermondt, R., Chavez, R. and Cooper, G. (1989) The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In Hunter, J. (ed.), *Proceedings of the Second European Conference on Artificial Intelligence and Medicine*. Berlin: Springer-Verlag.

Benedict, C. A., Banks, T. A., Senderowicz, L., Ko, M., Britt, W., Angulo, A., Ghazal, P. and Ware, C. (2001) Lymphotoxins and cytomegalovirus cooperatively induce interferon-b establishing host-virus détente. *Immunity*, **15**, 617–626.

Bishop, C. (2006) *Pattern Recognition and Machine Learning*. Springer, Singapore.

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

Butte, A. and Kohane, I. (2000) Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing*, **2000**, 418–429.

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

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

Chickering, D. (2002) Learning equivalence classes of Bayesian network structures. *Journal of Machine Learning Research*, **2**, 445–498.

Chickering, D. M. (1995) A transformational characterization of equivalent Bayesian network structures. *International Conference on Uncertainty in Artificial Intelligence (UAI)*, **11**, 87–98.

Chickering, D. M. (1996) Learning Bayesian networks is NP-complete. In Fisher, D. and Lenz, H. J. (eds.), *Learning from Data: Artificial Intelligence and Statistics*, volume 5, pp. 121–130. Springer, New York.

Cooper, G. F. and Herskovits, E. (1992) A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, **9**, 309–347.

Darnell, J., Kerr, I. and Stark, G. (1994) Jak-STAT pathways and transcriptional activation in response to IFNs and other extracellular signaling proteins. *Science*, **264**, 1415–1421.

Davis, J. and 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. ACM, New York, NY, USA.

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

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

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

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

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

Gelman, A., Carlin, J., Stern, H. and Rubin, D. (2004) *Bayesian data analysis*. Chapman and Hall/CRC, London, England.

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

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

Grzegorczyk, M. (2006) *Comparative Evaluation of different Graphical Models for the Analysis of Gene Expression Data*. Ph.D. thesis, Department of Statistics, Dortmund University. http://hdl.handle.net/2003/22855.

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

Grzegorczyk, M. and Husmeier, D. (2009a) Avoiding spurious feedback loops in the reconstruction of gene regulatory networks with dynamic Bayesian networks. In Kadirkamanathan, V., Sanguinetti, G., Girolami, M., Niranjan, M. and Noirel, J. (eds.), *Pattern Recognition in Bioinformatics*, pp. 113–124. Springer, Berlin, Heidelberg.

Grzegorczyk, M. and Husmeier, D. (2009b) Modelling non-stationary gene regulatory processes with a non-homogeneous dynamic Bayesian network and the change point process. In Manninen, T., Wiuf, C., Lähdesmäki, H., Grzegorczyk, M., Rahnenführer, J., Ahdesmäki, M., M.-L., L. and Yli-Harja, O. (eds.), *Sixth International Workshop on Computational Systems Biology (WCSB)*, pp. 51–54. TICSP series 48, Tampere, Finland.

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

Grzegorczyk, M. and Husmeier, D. (2011a) Improvements in the reconstruction of time-varying networks: dynamic programming and regularization by information sharing among genes. *Bioinformatics*, **27**, 693–699.

Grzegorczyk, M. and Husmeier, D. (2011b) Non-homogeneous dynamic Bayesian networks for continuous data. *Machine Learning*, **83**, 355–419.

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

Grzegorczyk, M., Husmeier, D. and Rahnenführer, J. (2010) Modelling non-stationary gene regulatory processes. *Advances in Bioinformatics*, **Volume 2010**, (online article, 20 pages).

Grzegorczyk, M., Rahnenführer, J. and Husmeier, D. (2011) Modelling non-stationary dynamic gene regulatory processes with the BGM model. *Computational Statistics*, **26**, 199–218.

Honda, K., Takaoka, A. and Taniguchi, T. (2006) Type I Inteferon gene induction by the Interferon regulatory factor family of transcription factors. *Immunity*, **25**, 349–360.

Husmeier, D., Dondelinger, F. and Lèbre, S. (2010) Inter-time segment information sharing for non-homogeneous dynamic Bayesian networks. In et al., L. J. (ed.), *Advances in Neural Information Processing Systems (NIPS) 23*, pp. 901–909.

Husmeier, D., Dybowski, R. and Roberts, S. (2005) *Probabilistic Modeling in Bioinformatics and Medical Informatics*. Advanced Information and Knowledge Processing. Springer, New York.

Ickstadt, K., Bornkamp, B., Grzegorczyk, M., Wieczorek, J., Sheriff, M., Grecco, H. and Zamir, E. (2010) Nonparametric Bayesian networks. In Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, F. and West, M. (eds.), *Bayesian Statistics 9*, pp. 283–316. Oxford University Press.

Imoto, S., Higuchi, T., Goto, T. and Miyano, S. (2006) Error tolerant model for incorporating biological knowledge with expression data in estimating gene networks. *Statistical Methodology*, **3**, 1–16.

Imoto, S., Kim, S., Goto, T., , Aburatani, S., Tashiro, K., Kuhara, S. and Miyano, S. (2003) Bayesian networks and nonparametric heteroscedastic regression for nonlinear modeling of genetic networks. *Journal of Bioinformatics and Computational Biology*, **1**, 231–252.

Kanehisa, M. (1997) A database for post-genome analysis. *Trends Genet*, **13**, 375376.

Kanehisa, M. and Goto, S. (2000) KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, **28**, 27–30.

Kanehisa, M., Goto, S., Hattori, M., Aoki-Kinoshita, K., Itoh, M., Kawashima, S., Katayama, T., Araki, M. and Hirakawa, M. (2006) From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research*, **34**, 354–357.

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

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

Kolar, M., Song, L. and Xing, E. (2009) Sparsistent learning of varying-coefficient models with structural changes. *NIPS*, **22**, 1006–1014.

Larget, B. and Simon, D. L. (1999) Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees. *Molecular Biology and Evolution*, **16**, 750–759.

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

Lèbre, S., Becq, J., Devaux, F., Stumpf, M. and Lelandais, G. (2010) Statistical inference of the time-varying structure of gene-regulation networks. *BMC Systems Biology*, **4**.

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

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

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

Margolin, A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R. and Califano, A. (2006) ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, **7**.

Más, P. (2008) Circadian clock function in Arabidopsis thaliana: time beyond transcription. *Trends in Cell Biology*, **18**, 273–281.

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

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

Nariai, N., Tamada, Y., Imoto, S. and Miyano, S. (2005) Estimating gene regulatory networks and protein-protein interactions of saccharomyces cerevisiae from multiple genome-wide data. *Bioinformatics*, **21**, ii206–ii212.

Neal, R. M. (2001) Annealed importance sampling. *Statistics and Computing*, **11**, 125–139.

Newman, D. J., Hettich, S., Blake, C. L. and Merz, C. J. (1998) UCI repository of machine learning databases. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

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

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

Pearl, J. (2000) *Causality: Models, Reasoning and Intelligent Systems*. Cambridge University Press, London, UK.

Pournara, I. and Wernisch, L. (2004) Reconstruction of gene networks using Bayesian learning and manipulation experiments. *Bioinformatics*, **20**, 2934–2942.

Raza, S., Robertson, K., Lacaze, P., Page, D., Enright, A., Ghazal, P. and Freeman, T. (2008) A logic based diagram of signalling pathways central to macrophage activation. *BMC Systems Biology*, **2**. Article 36.

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

Rogers, S. and 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. and Nolan, G. P. (2005) Protein-signaling networks derived from multiparameter single-cell data. *Science*, **308**, 523–529.

Salome, P. and McClung, C. (2004) The Arabidopsis thaliana clock. *Journal of Biological Rhythms*, **19**, 425–435.

Schäfer, J. and Strimmer, K. (2005a) An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, **21**, 754–764.

Schäfer, J. and Strimmer, K. (2005b) A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, **4**, Article 32.

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

Smith, V., Yu, J., Smulders, T., Hartemink, A. and Jarvis, E. (2006) Computational inference of neural information flow networks. *PLoS Computational Biology*, **2**, (online).

Spirtes, P., Glymour, C. and Scheines, R. (2001) *Causation, Prediction, and Search*. Springer Verlag, New York.

Suchard, M., Weiss, R., Dorman, K. and Sinsheimer, J. (2003) Inferring spatial phylogenetic variation along nucleotide sequences: a multiple change-point model. *Journal of the American Statistical Association*, **98**, 427–437.

Talih, M. and Hengartner, N. (2005) Structural learning with time-varying components: Tracking the cross-section of financial time series. *Journal of the Royal Statistical Society B*, **67**, 321–341.

Tamada, Y., Bannai, H., Imoto, S., Katayama, T., Kanehisa, M. and Miyano, S. (2005) Utilizing evolutionary information and gene expression data for estimating gene networks with Bayesian network models. *Journal of Bioinformatics and Computational Biology*, **3**, 1295–1313.

Tamada, Y., Kim, S., Bannai, H., Imoto, S., Tashiro, K., Kuhara, S. and Miyano, S. (2003) Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. *Bioinformatics*, **19**, ii227–ii236.

Verma, T. and Pearl, J. (1990) Equivalence and synthesis of causal models. In *International Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 6, pp. 255–270.

Vyshemirsky, V. and Girolami, M. (2008) Bayesian ranking of biochemical system models. *Bioinformatics*, **24**, 833–839.

Wang, Y., Cao, J. and Li, L. (2010) Global robust power-rate stability of delayed genetic regulatory networks with noise perturbation. *Cognitive Neurodynamics*, **4**, 81–90.

Werhli, A., Grzegorczyk, M. and Husmeier, D. (2006) Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical Gaussian models and Bayesian networks. *Bioinformatics*, **22**, 2523–2531.

Werhli, A. and Husmeier, D. (2007) Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Statistical Applications in Genetics and Molecular Biology*, **6**.

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

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

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