Mach Learn (2008) 71: 265–305
DOI 10.1007/s10994-008-5057-7

Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move

Marco Grzegorczyk · Dirk Husmeier

Received: 27 March 2007 / Revised: 11 January 2008 / Accepted: 28 March 2008 / Published online: 17 April 2008

Abstract Applications of Bayesian networks in systems biology are computationally demanding due to the large number of model parameters. Conventional MCMC schemes based on proposal moves in structure space tend to be too slow in mixing and convergence, and have recently been superseded by proposal moves in the space of node orders. A disadvantage of the latter approach is the intrinsic inability to specify the prior probability on network structures explicitly. The relative paucity of different experimental conditions in contemporary systems biology implies a strong influence of the prior probability on the posterior probability and, hence, the outcome of inference. Consequently, the paradigm of performing MCMC proposal moves in order rather than structure space is not entirely satisfactory. In the present article, we propose a new and more extensive edge reversal move in the original structure space, and we show that this significantly improves the convergence of the classical structure MCMC scheme.

Keywords Bayesian networks · Structure learning · MCMC sampling

1 Introduction

The concept of Bayesian networks is a cornerstone for modern statistical inference on regulatory networks. Following up on the seminal paper by Friedman et al. (2000), it has recently enjoyed considerable popularity in systems biology research. However, the large number of

Editor: Kevin P. Murphy.

M. Grzegorczyk (✉) · D. Husmeier
Centre for Systems Biology at Edinburgh (CSBE), Darwin Building, The King's Buildings, Edinburgh, UK
e-mail: Marco@bioss.sari.ac.uk

D. Husmeier
e-mail: Dirk@bioss.sari.ac.uk

M. Grzegorczyk · D. Husmeier
Biomathematics and Statistics Scotland (BioSS), JCMB, The King's Buildings, Edinburgh, UK

model parameters and the relative paucity of different experimental conditions renders inference computationally demanding, since the need to capture inference uncertainty calls for expensive bootstrapping or MCMC simulations. The latter are indispensable when adopting a proper Bayesian approach to inference, which tends to be computationally less expensive than the frequentest approach of bootstrapping (Larget and Simon 1999). However, the traditional structure MCMC sampling scheme of proposing MCMC moves in the space of network structures, as discussed in Madigan and York (1995) and Giudici and Castelo (2003), turns out to be rather slow in mixing and convergence, and it is not a viable approach to the analysis of high-throughput data in systems biology. Mixing and convergence of the Markov chain can be considerably improved with the alternative order MCMC sampling scheme in the space of node orders, as proposed by Friedman and Koller (2003). The disadvantage of this approach is that the prior on graph structures cannot be defined explicitly. Contemporary applications in systems biology are characterized by a paucity of different experimental conditions relative to the complexity of the employed models. This implies that the prior has usually a substantial influence on the posterior and the outcome of inference; hence, an approach that is intrinsically unable to specify it explicitly is not entirely satisfactory.

In the present article, we therefore stick to the classical structure MCMC concept of devising a Markov chain in the space of network structures rather than node orders, and we show that mixing and convergence in this space can be substantially improved by the introduction of a new edge reversal move type.

The main advantage of upgrading the structure MCMC sampling scheme by a new edge reversal move is that its computational costs are not increased over those of the original order MCMC sampler of Friedman and Koller (2003). That is, the bias of order MCMC can be avoided without incurring extra computational costs. Alternative approaches to correct this bias within the framework of sampling or marginalizing over node orders have been proposed, but come at substantially increased computational costs. Kovisto and Sood (2004) and Kovisto (2006) propose a dynamic programming algorithm to analytically marginalize over node orders to compute the marginal posterior probabilities of the edges. The approach is akin to a method proposed by Ott et al. (2004) in the context of optimization. Dynamic programming reduces the computational complexity from super-exponential to exponential in the number of nodes. However, these computational costs are still too expensive for large networks with more than typically 20 nodes. Eaton and Murphy (2007) propose a hybrid technique that is based on the dynamic programming approach of Kovisto and Sood (2004) and Kovisto (2006). Dynamic programming is used to analytically marginalize over node orders for the design of refined and more efficient MCMC proposal moves in structure space. While this approach has achieved substantially improved convergence and mixing results for small networks, it is still of exponential complexity in the number of nodes and, hence, not applicable for larger networks with more than typically 20 nodes. Alternatively, Ellis and Wong (2006) suggest the application of order MCMC followed by a correction step based on importance sampling. Ellis and Wong (2006) demonstrate that this correction step successfully reduces the bias intrinsic to the original order MCMC scheme of Friedman and Koller (2003). However, computing the exact correction is NP-hard, and even computing the approximate correction is too expensive for large networks with many nodes. Consequently, all these schemes are not practically viable for inferring large networks with more than 20 to 30 nodes, as is typically of interest in computational systems biology.

Another idea for Bayesian network structure learning is the inclusion-driven MCMC technique of Castelo and Kočka (2003), which is based on the concept of the inclusion boundary of a directed acyclic graph. The inclusion boundary of a graph is the set of all the graphs that can be reached from any graph in the equivalence class of the current graph by

a single-edge operation (edge addition, edge deletion, or non-covered edge reversal, where a non-covered edge is an edge that, on reversal, leads to a graph in a different equivalence class). Inclusion-driven MCMC, in its strict sense, is a modified classical structure MCMC simulation where at each step a new graph from the approximated inclusion boundary neighborhood of the current graph is proposed. As the inclusion boundary is not computationally efficient to handle in practice, Castelo and Kočka (2003) propose an alternative approach that is based on the reversal of covered edges. An edge in a graph is said to be covered if its reversal leads to a new graph in the same equivalence class as the old graph. Consequently, the reversal of covered edges provides an efficient way to traverse an equivalence class. The algorithm of Castelo and Kočka (2003) precedes a standard single-edge operation by a randomly chosen number of covered edge reversals, where the authors have devised a computationally cheap local method for identifying covered edges (an edge is covered if on its removal the parent sets of the two connected nodes are identical). In classical structure MCMC a move within an equivalence class has the same computational cost as any other move: it requires an acyclicity check to be carried out and the score of the proposed DAG to be computed. Inclusion-driven MCMC reduces the cost of this type of move by identifying and reverting covered edges. This renders both the computation of the score and the acyclicity check obsolete. The essence of inclusion-driven MCMC, thus, is an accelerated traversal of the equivalence classes. While this approach potentially speeds up the MCMC simulation to some extent, our experimental studies presented in Appendix 6 demonstrate that this method does not solve convergence and mixing problems of structure MCMC in principle. To understand this failure, note that mixing and convergence problems are caused by high-probability regions in configuration space being separated by low-probability regions that are difficult to traverse. Graphs of the same equivalence class lie along a ridge with the same probability score. While inclusion-driven MCMC provides a mechanism for a faster movement along these ridges, it does not provide any mechanism for bridging low-probability valleys. Hence, if the latter cause a structure MCMC simulation to stall, inclusion-driven MCMC will be unlikely to achieve any improvement. Additionally, we found that the potential improvement of inclusion-driven MCMC over structure MCMC was not significant in our simulations when the MCMC trajectory lengths were corrected for the different computational costs of the MCMC proposal steps. Finally, the inclusion-driven MCMC approach of Castelo and Kočka (2003) does not allow the Hastings factor to be computed properly, leading to a systematic bias that is akin to the one incurred with order MCMC. We therefore use the classical structure MCMC sampling scheme as an unbiased reference MCMC sampling scheme in the graph space throughout the present paper. A comparison between the inclusion-driven MCMC approach and our proposed new MCMC approach can be found in Appendix 4.

Mansinghka et al. (2006) introduce a hierarchical Bayesian framework that captures structural knowledge using novel edge priors, and hence improves network structure recovery when the underlying true graph possesses systemicity (block structured edge sets), or when the underlying true DAG is sparse, that is exhibits few edges. But when the true graph does not match these features their computationally more expensive hierarchical approach does not have superior performance than the classical structure MCMC sampling scheme.

In the present paper we therefore propose a novel method that improves the poor mixing of the classical structure MCMC sampler in general, without incurring the bias intrinsic to the order MCMC scheme of Friedman and Koller (2003), and without an inflation of the computational costs that would render large-scale applications inviable.

2 Bayesian network methodology

This section gives an introduction to Bayesian network methodology and introduces some graph theoretic notations we will use throughout this article.

Bayesian networks (BNs) are interpretable and flexible models for representing probabilistic relationships between interacting variables. At a qualitative level, the graph of a BN describes the relationships between the domain variables in the form of conditional independence relations. At a quantitative level, local relationships between variables are described by conditional probability distributions. Formally, a BN is defined by a graph $\mathcal{M}$, a family of conditional probability distributions F, and their parameters q, which together specify a joint distribution over the domain variables.

The graph $\mathcal{M}$ 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. The *directed edges* indicate dependence relations. If there is a directed edge pointing from node X_i to node X_j, 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, that is, the set of nodes from which an edge points to X_n in $\mathcal{M}$. We say that a node X_n is *orphaned* if it has an empty parent set: $\pi_n = \emptyset$. If a node X_k can be reached by following a *path* of directed edges starting at node X_i, then X_k is called a *descendant* of X_i. The structure of a Bayesian network is defined to be a *directed acyclic graph*, that is, a directed graph in which no node can be its own descendant. Graphically this means that there are no cycles of directed edges (loops) in DAGs. It is due to the acyclicity that the joint probability distribution in BNs can be factorized as follows:

$$P(X_1, \ldots, X_N) = \prod_{n=1}^{N} P(X_n|\pi_n). \tag{1}$$

For further details, see Jensen (1996). Thus, DAGs imply sets of conditional independence assumptions for BNs, and so factorizations of the joint probability distribution in which each node depends on its parent nodes only. But more than one DAG can imply exactly the same set of conditional independences, and if two DAGs assert the same set of conditional independence 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 directed acyclic graph (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).

Although Bayesian networks (BNs) are based on DAGs, it is important to note that not all directed edges in a BN can be interpreted causally. Like a BN, 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 BN in that it satisfies the corresponding Markov independences. However, the reverse does not hold.

The probability models for BNs we will consider in this paper lead to the same scores for equivalent DAGs, so that only equivalence classes can be learnt from data. Chickering (1995) shows that equivalence classes of DAGs can be uniquely represented using *completed*

partially directed acyclic graphs (CPDAGs). A CPDAG contains the same skeleton as the original DAG, but possesses both directed and undirected edges. Every directed edge $X_i \rightarrow X_j$ of a CPDAG denotes that all DAGs of this class contain this edge, while every undirected edge $X_i - X_j$ in this CPDAG-representation denotes that some DAGs contain the directed edge $X_i \rightarrow X_j$, while others contain the oppositely orientated edge $X_i \leftarrow X_j$. An algorithm that takes as input a DAG, and outputs the CPDAG representation of the equivalence class to which that DAG belongs, can be found in Chickering (2002).

Stochastic models for Bayesian networks (Friedman et al. 2000) specify the distributional form F and the parameters q of the local probability distributions $P(X_n|\pi_n)$ $(n = 1, \ldots, N)$. They assert a distribution to each domain node X_n conditional on its parent set π_n, whereby the parent sets are specified through the underlying DAG. The local probability distributions together specify the joint probability distribution of all domain variables $P(X_1, \ldots, X_N)$ (see (1)). Consequently, given data $\mathcal{D}$ these parametric models can be used to score DAGs $\mathcal{M}$ with respect to their posterior probabilities $P(\mathcal{M}|\mathcal{D}, F, q)$. Neglecting F and q, we have:

$$P(\mathcal{M}|\mathcal{D}) = \frac{P(\mathcal{M}, \mathcal{D})}{P(\mathcal{D})} = \frac{P(\mathcal{D}|\mathcal{M}) \cdot P(\mathcal{M})}{\sum_{\mathcal{M}^\star \in \Omega} P(\mathcal{D}|\mathcal{M}^\star) \cdot P(\mathcal{M}^\star)}, \tag{2}$$

whereby $P(\mathcal{M})$ $(\mathcal{M} \in \Omega)$ is the prior probability over the space Ω of all possible DAGs over the domain $X_1, \ldots, X_N$. $P(\mathcal{D}|\mathcal{M})$ is the marginal likelihood, that is the probability of the graph $\mathcal{M}$ given the data $\mathcal{D}$. A commonly used graph prior $P(\mathcal{M})$ $(\mathcal{M} \in \Omega)$ that we will use in our data applications except those where explicit prior knowledge is included is given by:

$$P(\mathcal{M}) = \frac{1}{\Pi} \prod_{n=1}^{N} \binom{N-1}{|\pi_n|}^{-1} \tag{3}$$

where Π is a normalization constant, and $|\pi_n|$ is the cardinality of the parent set π_n.

There are two major stochastic models for which certain regularity conditions can be satisfied, so that a closed form solution can be derived for the likelihood $P(\mathcal{D}|\mathcal{M})$ by analytical integration. See Cooper and Herskovits (1992) and Geiger and Heckerman (1994) for further details. The posterior probability $P(\mathcal{M}|\mathcal{D})$ (see (2)) has a modular form:

$$P(\mathcal{M}|\mathcal{D}) = \frac{1}{Z_c} \prod_{n=1}^{N} \exp(\psi[X_n, \pi_n|\mathcal{D}]). \tag{4}$$

Here, Z_c is a normalization factor, and $\psi[X_n, \pi_n|\mathcal{D}]$ are local scores that are computed from the data $\mathcal{D}$ and depend on the parent sets π_n implied through the DAG $\mathcal{M}$. The local scores $\psi[.]$ are defined by the employed probability model. The two major stochastic models, leading to a closed form solution, are (1) the linear Gaussian model with a Normal-Wishart distribution as the conjugate prior (BGe-model, see Geiger and Heckerman 1994), and (2) the multinomial distribution with a Dirichlet prior (BDe-model, see Cooper and Herskovits 1992). A comparison of these models in the context of reverse engineering gene regulatory networks can be found in Friedman et al. (2000).

Two different Markov chain Monte Carlo (MCMC) methods can be used for sampling directed acyclic graphs (DAGs) $\mathcal{M}$ from the posterior distribution $P(\mathcal{M}|\mathcal{D})$. The structure MCMC approach of Madigan and York (1995) generates a sample of DAGs $\mathcal{M}_1, \ldots, \mathcal{M}_T$ from the posterior distribution by a Metropolis Hastings sampler in the space of DAGs. Given a DAG $\mathcal{M}_i$, in a first step a new DAG $\mathcal{M}_{i+1}$ is proposed with the following proposal

probability $Q(\mathcal{M}_{i+1}|\mathcal{M}_i)$:

$$Q(\mathcal{M}_{i+1}|\mathcal{M}_i) = \begin{cases} \frac{1}{|\mathcal{N}(\mathcal{M}_i)|}, & \mathcal{M}_{i+1} \in \mathcal{N}(\mathcal{M}_i), \\ 0, & \mathcal{M}_{i+1} \notin \mathcal{N}(\mathcal{M}_i), \end{cases} \tag{5}$$

where $\mathcal{N}(\mathcal{M}_i)$ denotes the *neighborhood* of $\mathcal{M}_i$, that is the collection of all DAGs that can be reached from $\mathcal{M}_i$ by deletion, addition or reversal of one single edge of the current graph $\mathcal{M}_i$, and $|\mathcal{N}(\mathcal{M}_i)|$ is the cardinality of this collection. We note that the new graph $\mathcal{M}_{i+1}$ has to be acyclic, so it has to be checked which edges can be added to $\mathcal{M}_i$ and which edges can be reversed in $\mathcal{M}_i$ without violating the acyclicity-constraint. In the Metropolis Hastings algorithm the proposed graph $\mathcal{M}_{i+1}$ is accepted with the acceptance probability:

$$A(\mathcal{M}_{i+1}|\mathcal{M}_i) = \min\{1, R(\mathcal{M}_{i+1}|\mathcal{M}_i)\} \tag{6}$$

where

$$\begin{aligned} R(\mathcal{M}_{i+1}|\mathcal{M}_i) &:= \frac{P(\mathcal{M}_{i+1}|\mathcal{D})}{P(\mathcal{M}_i|\mathcal{D})} \cdot \frac{Q(\mathcal{M}_i|\mathcal{M}_{i+1})}{Q(\mathcal{M}_{i+1}|\mathcal{M}_i)} \\ &= \frac{P(\mathcal{D}|\mathcal{M}_{i+1}) \cdot P(\mathcal{M}_{i+1})}{P(\mathcal{D}|\mathcal{M}_i) \cdot P(\mathcal{M}_i)} \cdot \frac{|\mathcal{N}(\mathcal{M}_i)|}{|\mathcal{N}(\mathcal{M}_{i+1})|} \end{aligned} \tag{7}$$

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

$$\mathcal{K}(\tilde{\mathcal{M}}|\mathcal{M}) = Q(\tilde{\mathcal{M}}|\mathcal{M}) \cdot A(\tilde{\mathcal{M}}|\mathcal{M}) \tag{8}$$

and

$$\mathcal{K}(\mathcal{M}|\mathcal{M}) = 1 - \sum_{\tilde{\mathcal{M}} \in \mathcal{N}(\mathcal{M})} Q(\tilde{\mathcal{M}}|\mathcal{M}) \cdot A(\tilde{\mathcal{M}}|\mathcal{M}). \tag{9}$$

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

$$\frac{P(\tilde{\mathcal{M}}|\mathcal{D})}{P(\mathcal{M}|\mathcal{D})} = \frac{\mathcal{K}(\tilde{\mathcal{M}}|\mathcal{M})}{\mathcal{K}(\mathcal{M}|\tilde{\mathcal{M}})}. \tag{10}$$

Under ergodicity, that is a sufficient condition for the Markov chain $\{\mathcal{M}_i\}$ to converge, the posterior distribution $P(\mathcal{M}|\mathcal{D})$ is the stationary distribution:

$$P(\tilde{\mathcal{M}}|D) = \sum_{\mathcal{M}} \mathcal{K}(\tilde{\mathcal{M}}|\mathcal{M}) \cdot P(\mathcal{M}|\mathcal{D}). \tag{11}$$

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*. The practical advantage of the restriction on the maximum number of edges converging on a node is a reduction of the computational complexity, which improves the convergence. 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 DAG's neighborhood: DAGs that contain nodes with too many parents, that is more than the fan-in value, have to be removed from the respective neighborhoods.

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 $\prec_1, \ldots, \prec_T$ from the posterior distribution $P(\prec | \mathcal{D})$ over node orders $\prec$, so that the state space is the set of all $N!$ possible orders of the domain nodes.

Each node order $\prec = (X_{\sigma(1)}, \ldots, X_{\sigma(N)})$ can be seen as implied through a permutation σ of the indices $\{1, \ldots, N\}$. The meaning of such an order $\prec = (X_{\sigma(1)}, \ldots, X_{\sigma(N)})$ is that it represents all DAGs that are consistent with it in the following sense: A DAG $\mathcal{M}$ is consistent with $\prec$ if and only if the parent sets $\pi_{\sigma(n)}$ of the domain nodes $X_{\sigma(n)}$ are restricted to nodes that are standing to the left in $\prec$. That is, it has to hold: $X_{\sigma(i)} \notin \pi_{\sigma(j)}$ if $\sigma(j)$ precedes $\sigma(i)$ in $\prec$. We note that a fan-in restriction can be realized by additionally restricting the cardinalities of the parent sets $\pi_{\sigma(n)}$.

Friedman and Koller (2003) assume a uniform prior over node orders, that is $P(\prec) = \frac{1}{N!}$, and recommend to use a simple *flip-operator* which 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:

$$Q(\prec_{i+1} \mid \prec_i) = \begin{cases} \frac{2}{N \cdot (N-1)}, & \prec_{i+1} \in \Pi(\prec_i), \\ 0, & \prec_{i+1} \notin \Pi(\prec_i). \end{cases} \tag{12}$$

Thereby $\Pi(\prec_i)$ is the set of all node orders that can be reached from $\prec_i$ by exchanging in $\prec_i$ two nodes for each other, and leaving the positions of all other nodes in $\prec_i$ unchanged. To guarantee convergence to the posterior probability $P(\prec | \mathcal{D})$ 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:

$$R(\prec_{i+1} \mid \prec_i) = \frac{P(\prec_{i+1} | \mathcal{D})}{P(\prec_i | \mathcal{D})} \cdot \frac{Q(\prec_i \mid \prec_{i+1})}{Q(\prec_{i+1} \mid \prec_i)} = \frac{P(\mathcal{D} | \prec_{i+1})}{P(\mathcal{D} | \prec_i)} \tag{13}$$

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

$$P(\mathcal{D} | \prec) = \prod_{n=1}^{N} \sum_{\pi \in \mathcal{U}_{\sigma(n)}^{\prec}} \exp(\psi[X_{\sigma(n)}, \pi | \mathcal{D}]) \tag{14}$$

where $\mathcal{U}_{\sigma(n)}^{\prec}$ is the set of all possible parent sets for node $X_{\sigma(n)}$ that contain exclusively nodes standing to the left of $X_{\sigma(n)}$ in the node order $\prec$. As the orders $\prec_i$ and $\prec_{i+1}$ differ by the position of two nodes only, the likelihood ratio in (13) can be computed more efficiently, as explained in Friedman and Koller (2003).

We note that it is useful to precompute and store the $\psi[.]$ scores of all the domain nodes and their parent sets at the beginning of the process. The sums of local scores in the likelihoods (see (13)) can then be computed from these cached lists and do not have to be recomputed everytime when needed. Furthermore Friedman and Koller (2003) introduce a pruning approach which further reduces the computational complexity of the summations.

In a second step, given the sample of node orders $\prec_1, \ldots, \prec_T$ a sample of DAGs can be obtained by a simple sampling approach (Friedman and Koller 2003). Given the order $\prec$ for each domain node X_n its parent set π_n can be sampled independently from the following distribution:

$$P(\pi_n) = \frac{\exp(\psi[X_n, \pi_n | \mathcal{D}]) \cdot I(X_n, \pi_n, \prec)}{\sum_{\pi} \exp(\psi[X_n, \pi | \mathcal{D}]) \cdot I(X_n, \pi, \prec)} \tag{15}$$

where 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 preceded node X_n in the order $\prec$, and 0 otherwise. Sampling a parent set π_n for each domain node X_n yields a complete DAG. 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 influence 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 $P(\mathcal{M})$ (see (3) with regard to our applications). DAGs that are consistent with more orders are more likely than DAGs consistent with fewer orders. For instance, the DAG without any edge can be sampled out of all $N!$ node orders, while a DAG of the type $X_{\sigma(1)} \to X_{\sigma(2)} \to \cdots \to X_{\sigma(N)}$ can be sampled out of one single node order, namely $\prec = (X_{\sigma(1)}, \ldots, X_{\sigma(N)})$, only. To recapitulate this: While we can specify the prior on the graphs given the node order, $P(\mathcal{M}|\prec)$, the explicit computation of the prior over graphs requires a marginalization over orders: $P(\mathcal{M}) = \sum_{\prec} P(\mathcal{M}|\prec) \cdot P(\prec)$. The distortion inherent in the marginalization means that we are effectively unable 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 likelihood. 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 likelihood, is relatively low. In what follows, we therefore investigate an alternative approach.

3 The new reversal move

3.1 Non-mathematical exposition

In this section, we will explain the concept of a new edge reversal move for structure MCMC without any equations. The mathematical details will be provided in Sect. 3.2.

It is known that structure MCMC is slow in mixing and convergence, as it is based on single edge operations, so that the modifications of the graph are small and the sampler tends to get trapped in local maxima (Castelo and Kočka 2003; Friedman and Koller 2003). The structure MCMC approach allows the reversal of an edge, only if the reversal leads to a new valid DAG. Therefore, the first obvious problem is that it depends on the overall structure of the current graph which edges can be reversed. We want to avoid this shortcoming by changing the parent sets of both nodes that are connected by the edge in a more involved way. That is, we sample completely new parent sets for both nodes, so that (i) the corresponding edge will point into the opposite direction in the new graph, and (ii) the overall graph structure is valid (acyclic).

Even for those edges that can be reversed by the classical structure MCMC edge 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 the single edge only, the acceptance probability can be increased substantially. This is due to the fact that 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 of the two nodes. While reversing the edge, the new reversal move guarantees that both parent sets are merely resampled according to their score, so that the new sets can be expected to be 'higher-scoring' on average. More precisely, for both nodes the parent sets become adopted to the

new direction of the edge. Thus, the key idea of the new reversal move is to render possible a bigger modification of the current DAG which is tailor-made to the new direction of the edge to be reversed.

We note that the key idea of this new edge reversal move is similar to the optimal reinsertion operator introduced by Moore and Wong (2003). This reinsertion operator first deletes all edges connected to a selected node, and then reconnects the selected node to new child and parent nodes according to a local optimization criterion subject to the acyclicity constraint. Moore and Wong (2003) demonstrate that this move improves the efficiency of a greedy search in structure space. However, while viable in the context of *optimization*, e.g. to find the maximum a posteriori network structure, the proposed scheme does not seem useful for *sampling* network structures from the posterior distribution with MCMC. This is because the reinsertion operation does not allow the clear and unambiguous definition of a one-to-one mapping between complementary forward and backward moves, so that the computation of the acceptance probabilities of the resulting MCMC sampling scheme would require a computationally expensive search for all possible transitions between networks.

Fortunately, this problem does not occur when employing our new edge reversal move, where the mapping between complementary moves is straightforward, as discussed in Sect. 3.2.

3.2 Mathematical details of the new reversal move

We introduce the following definitions. $\mathcal{M}$ denotes a graph over the domain $\{X_1, \ldots, X_N\}$ that consists of directed edges only, whereas $\mathcal{M}$ does not necessarily have to be an acyclic graph. For a domain node X_n we define $\mathcal{M}^{\{X_n\}\leftarrow\emptyset}$ to be the graph obtained by setting the parent set of X_n to the empty set, that is, we obtain the new graph by removing from $\mathcal{M}$ all edges pointing to X_n. We also say that $\mathcal{M}^{X_n\leftarrow\emptyset}$ is obtained by *orphaning* node X_n. Correspondingly, for two domain nodes X_i and X_j we define $\mathcal{M}^{\{X_i,X_j\}\leftarrow\emptyset}$ to be the graph obtained by *orphaning* both nodes X_i and X_j. We refer to the graph in which the old parent set of a node X_n is replaced by a new parent set $\tilde{\pi}_n$ as $\mathcal{M}^{X_n\leftarrow\tilde{\pi}_n}$, and correspondingly we refer to 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 $\tilde{\pi}_j$ as $\mathcal{M}^{X_i\leftarrow\tilde{\pi}_i,X_j\leftarrow\tilde{\pi}_j}$. For clarity, we note that

$$(\mathcal{M}^{X_i\leftarrow\tilde{\pi}_i,X_j\leftarrow\tilde{\pi}_j})^{X_i\leftarrow\pi_i} = \mathcal{M}^{X_i\leftarrow\pi_i,X_j\leftarrow\tilde{\pi}_j} \tag{16}$$

where π_i and $\tilde{\pi}_i$ denote two different parent sets for node X_i.

Furthermore, we introduce the following indicator function in the space $\{\mathcal{M}\}$ of (directed) graphs:

$$\delta : \{\mathcal{M}\} \to \{0, 1\} \tag{17}$$

with $\delta(\mathcal{M}) = 1$ if the graph $\mathcal{M}$ is a directed *acyclic* graph (DAG), and $\delta(\mathcal{M}) = 0$ if $\mathcal{M}$ is a cyclic graph, that is, if $\mathcal{M}$ contains a directed cycle. Finally, we define the following two partition functions: Given a DAG $\mathcal{M}$, the first partition function is a sum of local scores over all those parent sets π of X_n for which $\mathcal{M}^{X_n\leftarrow\pi}$ is a valid DAG.

$$Z(X_n|\mathcal{M}) := \sum_{\pi:\delta(\mathcal{M}^{X_n\leftarrow\pi})=1} \exp(\psi[X_n, \pi|\mathcal{D}]). \tag{18}$$

Given a DAG $\mathcal{M}$ and a domain node X_m, the second partition function is a sum of local scores over all those parent sets π of X_n which contain X_m, symbolically: $X_m \in \pi$, and for

which $\mathcal{M}^{X_n \leftarrow \pi}$ is a valid DAG.

$$Z^{\star}(X_n|\mathcal{M}, X_m) := \sum_{\substack{\pi:\delta(\mathcal{M}^{X_n \leftarrow \pi})=1 \\ X_m \in \pi}} \exp(\psi[X_n, \pi|\mathcal{D}]). \tag{19}$$

The new edge reversal move works as follows:

First step: Given a DAG $\mathcal{M}$, we randomly select one of its (directed) edges $X_i \rightarrow X_j$ from a uniform distribution over all edges in $\mathcal{M}$. By orphaning both nodes X_i and X_j we obtain a new DAG which we denote: $\mathcal{M}_{\odot} := \mathcal{M}^{\{X_i, X_j\} \leftarrow \emptyset}$.

Second step: In the second step we sample a new parent set $\tilde{\pi}_i$ for 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 $\mathcal{M}_{\odot}$, symbolically: $\delta(\mathcal{M}_{\odot}^{X_i \leftarrow \tilde{\pi}_i}) = 1$. That is, $\tilde{\pi}_i$ is sampled from the following modified Boltzmann distribution:

$$Q(\tilde{\pi}_i|\mathcal{M}_{\odot}, X_j) = \frac{\exp(\psi[X_i, \tilde{\pi}_i|\mathcal{D}]) \cdot \delta(\mathcal{M}_{\odot}^{X_i \leftarrow \tilde{\pi}_i}) \cdot I(\tilde{\pi}_i, X_j)}{Z^{\star}(X_i|\mathcal{M}_{\odot}, X_j)}, \tag{20}$$

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 (19). Having sampled the new parent set $\tilde{\pi}_i$ for node X_i, we set $\mathcal{M}_{\oplus} := \mathcal{M}_{\odot}^{X_i \leftarrow \tilde{\pi}_i}$, so that $\mathcal{M}_{\oplus}$ is a valid DAG that contains the oppositely oriented edge $X_i \leftarrow X_j$.

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

$$Q(\tilde{\pi}_j|\mathcal{M}_{\oplus}) = \frac{\exp(\psi[X_j, \tilde{\pi}_j|\mathcal{D}]) \cdot \delta(\mathcal{M}_{\oplus}^{X_j \leftarrow \tilde{\pi}_j})}{Z(X_j|\mathcal{M}_{\oplus})}, \tag{21}$$

whereby the partition function $Z(.)$ was defined in (18). That is, we sample a new parent set $\tilde{\pi}_j$ for X_j, so that the graph $\mathcal{M}_{\oplus}^{X_j \leftarrow \tilde{\pi}_j}$ which possesses the reversed edge $X_j \leftarrow X_i$ remains acyclic, symbolically: $\delta(\mathcal{M}_{\oplus}^{X_j \leftarrow \tilde{\pi}_j}) = 1$. This DAG $\tilde{\mathcal{M}} := \mathcal{M}_{\oplus}^{X_j \leftarrow \tilde{\pi}_j}$ is proposed by the new reversal move.

An example illustrating the new edge reversal move is given in Fig. 1. For the remainder of this article we will denote this new edge reversal move as the *REV* move. A brief and concise summary of the new edge reversal (REV) move algorithm can be found in Appendix 1. The proposal probability $Q^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M})$ of the REV move from $\mathcal{M}$ to $\tilde{\mathcal{M}}$ by reversing the edge $X_i \rightarrow X_j$ is then given by:

$$\begin{aligned} Q^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M}) = \frac{1}{N^{\dagger}} \cdot & \frac{\exp(\psi[X_i, \tilde{\pi}_i|\mathcal{D}]) \cdot \delta(\mathcal{M}_{\odot}^{i \leftarrow \tilde{\pi}_i}) \cdot I(\tilde{\pi}_i, X_j)}{Z^{\star}(X_i|\mathcal{M}_{\odot}, X_j)} \\ & \times \frac{\exp(\psi[X_j, \tilde{\pi}_j|\mathcal{D}]) \cdot \delta(\mathcal{M}_{\oplus}^{j \leftarrow \tilde{\pi}_j})}{Z(X_j|\mathcal{M}_{\oplus})}, \end{aligned} \tag{22}$$

where $N^{\dagger}$ is the number of edges in $\mathcal{M}$, $\mathcal{M}_{\odot} := \mathcal{M}^{\{X_i, X_j\} \leftarrow \emptyset}$, $\mathcal{M}_{\oplus} := \mathcal{M}_{\odot}^{X_i \leftarrow \tilde{\pi}_i}$, and the partition functions were defined in (18) and (19). The indicator function $I(\tilde{\pi}_i, X_j)$ is equal to 1 if $\tilde{\pi}_i$ contains X_j and 0 otherwise. $\delta(.)$ indicates acyclic graphs.

We note that we have the following relationship between the DAGs $\mathcal{M}$ and $\tilde{\mathcal{M}}$:

$$\begin{aligned} \tilde{\mathcal{M}} = \mathcal{M}_{\oplus}^{X_j \leftarrow \tilde{\pi}_j} &= (\mathcal{M}_{\odot}^{X_i \leftarrow \tilde{\pi}_i})^{X_j \leftarrow \tilde{\pi}_j} = ((\mathcal{M}^{\{X_i, X_j\} \leftarrow \emptyset})^{X_i \leftarrow \tilde{\pi}_i})^{X_j \leftarrow \tilde{\pi}_j} \\ &= \mathcal{M}^{X_i \leftarrow \tilde{\pi}_i, X_j \leftarrow \tilde{\pi}_j}. \end{aligned} \tag{23}$$

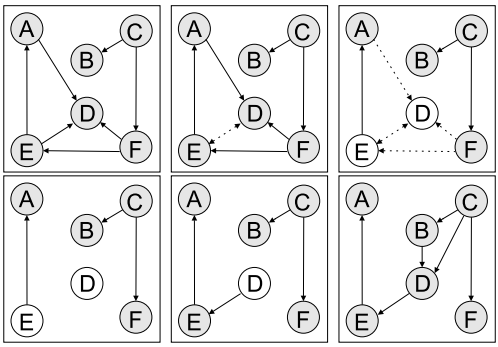


Fig. 1 Illustration of the new edge reversal (REV) move. Step 1 (*top left*): $\mathcal{M}$ is a directed acyclic graph (DAG) over the domain $\{A, \ldots, F\}$. Step 2 (*top centre*): We select an edge of $\mathcal{M}$ at random; in this example, it is the edge $E \to D$. This edge will be reversed by the move. We note that this edge could not be reversed by the classical edge reversal operation of structure MCMC, as the reversal of this edge without any further modification would lead to the directed cycle $A \to D \to E \to A$ and so to an invalid (cyclic) DAG. Step 3 (*top right*): The selected edge connects the domain nodes E and D. Orphaning these two nodes by deleting all edges feeding into them, we obtain the graph $\mathcal{M}_\odot$. In this example the dotted edges have to be deleted. Step 4 (*bottom left*): This is the DAG $\mathcal{M}_\odot = \mathcal{M}^{\{E,D\}\leftarrow\emptyset}$. The nodes E and D have been orphaned, and we will sample and assign new parent sets to them in the next two steps. First, we sample a new parent set for node E from $Q(\tilde{\pi}_E|\mathcal{M}_\odot, D)$ (see (20)). That is, we sample a parent set $\tilde{\pi}_E$ for E which contains D, so that the graph $\mathcal{M}_\odot^{E\leftarrow\tilde{\pi}_E}$ is acyclic and contains the oppositely oriented edge $E \leftarrow D$. In this example the set $\tilde{\pi}_E = \{D\}$ is sampled. Step 5 (*bottom centre*): Assigning this new parent set $\tilde{\pi}_E$ to node E gives the DAG $\mathcal{M}_\oplus = \mathcal{M}_\odot^{E\leftarrow\{D\}}$ shown here. Subsequently, we sample a new parent set for node D from $Q(\tilde{\pi}_D|\mathcal{M}_\oplus)$ (see (21)). That is, we sample a new parent set $\tilde{\pi}_D$ for node D, so that the graph $\mathcal{M}_\oplus^{D\leftarrow\tilde{\pi}_D}$ is acyclic. In this example the set $\tilde{\pi}_D = \{B, C\}$ is sampled. Step 6 (*bottom right*): Assigning this new parent set $\tilde{\pi}_D$ to node D gives the final DAG $\tilde{\mathcal{M}} = \mathcal{M}_\oplus^{D\leftarrow\{B,C\}}$ shown here. The inverse new edge reversal (REV) move leading back from $\tilde{\mathcal{M}}$ to $\mathcal{M}$ by reversing the edge $E \leftarrow D$ is illustrated in Fig. 9 in Appendix 3

Each REV move changes the parent sets of two nodes and leaves the parent sets of all other domain nodes unchanged. With regard to the computation of the acceptance probability of the Metropolis Hastings algorithm we have to design a complementary REV move leading backward from $\tilde{\mathcal{M}}$ to $\mathcal{M}$. To this end we state the following theorem, whose proof can be found in Appendix 2:

Theorem *For each REV move leading from a DAG $\mathcal{M}$ to a DAG $\tilde{\mathcal{M}}$ by reversing the edge $X_i \to X_j$, there is exactly one inverse REV move leading back from $\tilde{\mathcal{M}}$ to $\mathcal{M}$. The inverse move selects the edge $X_j \to X_i$ in $\tilde{\mathcal{M}}$ for edge reversal, and orphaning both nodes X_i and X_j in the first step yields the DAG $\tilde{\mathcal{M}}_\odot$. In the second step the parent set π_j of X_j in $\mathcal{M}$ is sampled and assigned as new parent set for X_j in $\tilde{\mathcal{M}}_\odot$, which gives the DAG $\tilde{\mathcal{M}}_\oplus$. Finally, in the third step the parent set π_i of X_i in $\mathcal{M}$ is sampled and assigned as the new parent set for X_i in $\tilde{\mathcal{M}}_\oplus$, which gives the DAG $\mathcal{M}$.*

We have the following relationships between the DAGs: First, as $\mathcal{M}$ and $\tilde{\mathcal{M}}$ differ by the parent sets of X_i and X_j only, we have:

$$\mathcal{M}_\odot := \mathcal{M}^{\{X_i,X_j\}\leftarrow\emptyset} = \tilde{\mathcal{M}}^{\{X_i,X_j\}\leftarrow\emptyset} =: \tilde{\mathcal{M}}_\odot \tag{24}$$

and we can derive:

$$\mathcal{M}_{\oplus} := \mathcal{M}_{\odot}^{X_i \leftarrow \tilde{\pi}_i} = \tilde{\mathcal{M}}_{\odot}^{X_i \leftarrow \tilde{\pi}_i} = (\tilde{\mathcal{M}}^{\{X_i, X_j\} \leftarrow \emptyset})^{X_i \leftarrow \tilde{\pi}_i} = \tilde{\mathcal{M}}^{X_j \leftarrow \emptyset}, \tag{25}$$

and

$$\tilde{\mathcal{M}}_{\oplus} := \tilde{\mathcal{M}}_{\odot}^{X_j \leftarrow \pi_j} = \mathcal{M}_{\odot}^{X_j \leftarrow \pi_j} = (\mathcal{M}^{\{X_i, X_j\} \leftarrow \emptyset})^{X_j \leftarrow \pi_j} = \mathcal{M}^{X_i \leftarrow \emptyset}. \tag{26}$$

The inverse REV move for the example move presented in Fig. 1 is illustrated in Appendix 3. We specify the acceptance probability of REV moves with respect to the Metropolis Hastings algorithm where it is given by the product of the posterior probability ratio and the inverse ratio of the proposal probabilities (the so-called Hastings factor). The acceptance probability $A^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M})$ of a REV move from graph $\mathcal{M}$ to graph $\tilde{\mathcal{M}}$ ($\mathcal{M} \neq \tilde{\mathcal{M}}$) is given by (Hastings 1970):

$$\begin{aligned} A^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M}) &= \min\{1, R^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M})\}, \\ R^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M}) &= \frac{P(\tilde{\mathcal{M}}|\mathcal{D})}{P(\mathcal{M}|\mathcal{D})} \frac{Q^{\triangleright}(\mathcal{M}|\tilde{\mathcal{M}})}{Q^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M})} \end{aligned} \tag{27}$$

where $Q^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M})$ is the proposal probability for a REV move from $\mathcal{M}$ to $\tilde{\mathcal{M}}$ ($\mathcal{M} \neq \tilde{\mathcal{M}}$) , and $Q^{\triangleright}(\mathcal{M}|\tilde{\mathcal{M}})$ is the proposal probability for a REV move from $\tilde{\mathcal{M}}$ to $\mathcal{M}$. The transition probability for a REV move from $\mathcal{M}$ to $\tilde{\mathcal{M}}$ is then given by:

$$\mathcal{K}^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M}) = Q^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M}) \cdot A^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M}) \tag{28}$$

if $\mathcal{M} \neq \tilde{\mathcal{M}}$ and

$$\mathcal{K}^{\triangleright}(\mathcal{M}|\mathcal{M}) = 1 - \sum_{\tilde{\mathcal{M}} \in \mathcal{N}(\mathcal{M})} Q^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M}) \cdot A^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M}) \tag{29}$$

where $\mathcal{N}(\mathcal{M})$ is the set of all DAGs that can be reached from $\mathcal{M}$ by a new edge reversal move. It follows from (27) that:

$$\frac{P(\tilde{\mathcal{M}}|\mathcal{D})}{P(\mathcal{M}|\mathcal{D})} = \frac{\mathcal{K}^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M})}{\mathcal{K}^{\triangleright}(\mathcal{M}|\tilde{\mathcal{M}})}. \tag{30}$$

When computing the overall acceptance ratio for the REV move according to (27), all local scores corresponding to unaffected nodes cancel out in the ratio. The exponential terms, that is, the numerators of the Boltzmann-like proposal distributions also cancel out against the corresponding terms in the posterior ratio. Hence, all that remains to be computed for a REV move from $\mathcal{M}$ to $\tilde{\mathcal{M}}$ ($\mathcal{M} \neq \tilde{\mathcal{M}}$) by reversing the edge $X_i \rightarrow X_j$ are the modified partition functions, and we obtain, from (22):

$$A^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M}) = \min\left\{1, \frac{N^{\dagger}}{\tilde{N}^{\dagger}} \cdot \frac{Z^{\star}(X_i|\mathcal{M}_{\odot}, X_j)}{Z^{\star}(X_j|\tilde{\mathcal{M}}_{\odot}, X_i)} \cdot \frac{Z(X_j|\mathcal{M}_{\oplus})}{Z(X_i|\tilde{\mathcal{M}}_{\oplus})}\right\} \tag{31}$$

where $N^{\dagger}$ is the number of edges in $\mathcal{M}$, $\tilde{N}^{\dagger}$ is the number of edges in $\tilde{\mathcal{M}}$, and the partition functions were defined in (18) and (19).

While the preceding exposition has proven that the novel REV move allows the construction of a reversible Markov chain that, under ergodicity, converges to the correct posterior

distribution, we note that ergodicity is not guaranteed to hold. As an extreme example, consider a network consisting of two nodes A and B. The state space contains two subspaces. The first subspace consists of the unconnected network. The second subspace comprises two networks with a directed edge, one pointing from A to B, the other pointing from B to A. While the REV move enables transitions *within* the second subspace, it does not allow transitions *between* the subspaces, thereby violating ergodicity. Ergodicity can be guaranteed, though, by combining the proposed REV move with classical structure MCMC moves for deleting and creating individual edges. We discuss the combination of these moves in Sect. 4.

3.3 Computational issues

The complexity of the computations required for the proposed edge reversal move can be substantially reduced using the same ideas and approximations as in Friedman and Koller (2003) for order MCMC. For each domain node X_i the scores of its potential parent sets can be precomputed and stored, instead of recomputing them each time when required. To reduce the computational costs and to make this approach viable for applications with large numbers of nodes, we can restrict the potential parents of a node X_i to only those nodes that lie in a candidate set of reduced cardinality m_P. This candidate set, for instance, contains only those nodes that have the highest local scores when considered as sole parents of X_i. Note that this restriction to candidate sets introduces an approximation, though, and it was not applied in the simulations reported in Sect. 5.

When reversing an edge $X_i \to X_j$ in the current graph $\mathcal{M}$ by the new edge reversal move, the most time-consuming step is the computation of the four partition functions $Z^\star(X_i|\mathcal{M}_\odot, X_j)$, $Z^\star(X_j|\tilde{\mathcal{M}}_\odot, X_i)$, $Z(X_i|\mathcal{M}_\oplus)$, and $Z(X_j|\tilde{\mathcal{M}}_\oplus)$. This is because we have to test for each potential parent set whether it satisfies the acyclicity constraint and, hence, leads to a valid DAG. Having stored, for each domain node, a list of all potential parent sets and their local scores, this task can be effected straightforwardly by running through each list and marking all invalid parent sets. Consider, for instance, the computation of $Z(X_j|\mathcal{M}_\oplus)$. First, we determine all descendants of X_j in the graph $\mathcal{M}_\oplus$. Next, we run through the list stored for X_j and mark all those scores that correspond to a parent set containing a descendant of X_j. Finally, we sum over the unmarked scores to obtain the partition function $Z(X_j|\mathcal{M}_\oplus)$, and sample a new parent set $\tilde{\pi}_j$ for X_j according to (21). In the same vein, when computing $Z^\star(X_i|\mathcal{M}_\odot, X_j)$, we run through the list stored for X_i and mark all those scores that either do not include X_j or contain a descendant of X_j in $\mathcal{M}_\odot$. Again, summing over the unmarked scores gives the partition function $Z^\star(X_i|\mathcal{M}_\odot, X_j)$, and we can sample a new parent set $\tilde{\pi}_i$ for X_i according to (20).

4 The new REV-structure MCMC sampler

In this section we describe how to upgrade the classical structure MCMC sampler by integrating the new edge reversal (REV) move. The classical structure MCMC sampler is based on three different edge operations: (i) single-edge-addition, (ii) single-edge-deletion and (iii) single-edge-reversal. More precisely, given a current directed acylic graph (DAG) $\mathcal{M}$, structure MCMC proposes a new DAG $\tilde{\mathcal{M}}$ which can be obtained from $\mathcal{M}$ by the addition, deletion or the reversal of a single edge. See Sect. 2 for further details. We propose not to restrict on these single edge operations but to upgrade structure MCMC by also allowing more extensive new edge reversal (REV) moves presented in Sect. 3. That is, we specify a

probability $p_R \in [0, 1]$, with which a REV move is chosen in each iteration of the MCMC simulation.

If we decide to upgrade the classical structure MCMC move by REV moves, whereby a REV move is performed with probability p_R and a classical single-edge structure MCMC move is performed with the inverse probability $p_S := 1 - p_R$, then we obtain the following new mixture kernel $\mathcal{K}^+(.)$ for a move from $\mathcal{M}$ to $\tilde{\mathcal{M}}$:

$$\mathcal{K}^+(\tilde{\mathcal{M}}|\mathcal{M}) = p_S \cdot \mathcal{K}(\tilde{\mathcal{M}}|\mathcal{M}) + p_R \cdot \mathcal{K}^{\triangleright}(\tilde{\mathcal{M}}|\mathcal{M}) \tag{32}$$

where the transition kernels $\mathcal{K}(.)$ and $\mathcal{K}^{\triangleright}(.)$ have been specified in (8) and (28). As both kernels $\mathcal{K}(.)$ and $\mathcal{K}^{\triangleright}(.)$ have the same stationary distribution and one of them, namely $\mathcal{K}(.)$, is ergodic, the mixture $\mathcal{K}^+(.)$ also converges to this stationary distribution. See Tierney (1994) for more details. For the remainder of this article we will refer to the upgraded structure MCMC sampler as *REV-structure MCMC* sampler.

In principle, p_R can be adapted during the burn-in period so as to optimize the acceptance probability rate. However, in our simulations we experimented with different probabilities p_R and found that the results varied little over quite a large range of p_R around $1/15$. That is, even when integrating the new REV move with a fixed probability of p_R, the classical structure MCMC sampler is already substantially upgraded and the learning performances become approximately as good as those of order MCMC (but without having any systematic bias). See Appendix IV for some experimental results that we obtained with different parameters p_R for the Boston Housing data (see Sect. 5.4). We therefore decided to avoid unnecessary complexity of our algorithm and selected the fixed value of $p_R = 1/15$ for all our REV-structure MCMC simulations in Sect. 5. Another advantage according to our experimental evaluation (see Sect. 5) is the following one: Choosing a fixed value for p_R renders possible an ex ante estimation of the average computational costs of a REV-structure MCMC move, and we can assign run lengths to all three MCMC samplers which lead to approximately the same amount of computational time.

5 Experimental results

5.1 Inference and implementation details

In this section we will demonstrate that the (standard) structure MCMC approach for sampling Bayesian networks (see Sect. 2) can be substantially improved by additionally allowing the new REV move presented in Sect. 3. We show that the resulting REV-structure MCMC sampler (see Sect. 4) performs equally well in terms of AUROC-values as the node order based order MCMC approach by Friedman and Koller (2003), but without having any systematic bias. Due to computational constraints it is practically impossible to compute exact *edge (relation) feature* posterior probabilities (see Sect. 5.2) for domains with too many nodes, say more than $N = 5$ or $N = 6$ nodes. Therefore we compare the posterior probabilities obtained by the different MCMC sampling schemes with the true posterior probabilities only for small graphs with $N = 5$ nodes. For graphs with more than $N = 5$ nodes, we restrict on diagnostics, such as AUROC values (see Sect. 5.2). The exact diagnostics for small domains reveal that the bias of order MCMC can be avoided by using our REV-structure MCMC sampler, and for bigger domains we show that the REV-structure MCMC sampler (i) performs often much better than classical structure MCMC and (ii) does not perform worse than order MCMC in terms of AUROC values.

We implemented the standard structure MCMC approach and the order MCMC approach according to the presentations given in Madigan and York (1995) and Friedman and Koller (2003), respectively. As the computational costs c_O of order MCMC iterations are approximately 10-times as expensive as the computational costs c_S of structure MCMC iterations (Friedman and Koller 2003), we decided to use inversely proportional MCMC run-lengths n_S (structure MCMC) and n_O (order MCMC), that is: $n_S = 10 \cdot n_O$. For REV-structure MCMC we have to distinguish between the computational costs c_R that occur when a new edge reversal (REV) move is performed and the computational costs c_S that occur when a standard structure MCMC move is performed. More precisely, we have to specify the probability p_R with which a REV move is performed, and the computational costs c_N of a new REV-structure MCMC iteration are then (on average) given by:

$$c_N = p_R \cdot c_R + (1 - p_R) \cdot c_S. \tag{33}$$

In our implementation the computational costs c_R of a REV move are much lower than the computational costs c_O of an order MCMC move. Nevertheless, we assumed that both are equally expensive; symbolically we then have: $c_N = c_O = 10 \cdot c_S$, and we obtain from (33):

$$c_N = p_R \cdot (10 \cdot c_S) + (1 - p_R) \cdot c_S = c_S \cdot (9 \cdot p_R + 1). \tag{34}$$

A fair run length n_N of our new REV-structure MCMC sampler can then be obtained from:

$$n_S \cdot c_S = n_N \cdot c_N = n_N \cdot c_S \cdot (9 \cdot p_R + 1) \Leftrightarrow n_N = \frac{n_S}{9 \cdot p_R + 1}. \tag{35}$$

In our applications we always used the following settings: For structure MCMC we set the burn-in length to 500,000 and then collected 1000 DAGs by sampling every 1000 iterations. Correspondingly, for order MCMC we set the burn-in length to 50,000 and then collected 1000 DAGs by sampling every 100 iterations. For REV-structure MCMC the burn-in length and the distance between sampling steps depends on the probability p_R with which a REV move is performed. From (35) it follows that it is fair to set the new REV-structure MCMC burn-in length to $\frac{500{,}000}{9 \cdot p_R + 1}$ and then to collect 1000 DAGs by sampling every $\frac{1000}{9 \cdot p_R + 1}$ iterations. For our choice: $p_R = 1/15$ this yields a burn-in length of 312,500 and then sampling every 625 iterations until a DAG sample of size 1000 is collected.

Finally we note that we always performed at least two independent runs with each MCMC sampler on every test data set. Following (Friedman and Koller 2003), we started the simulations from the following initializations: As uninformed initialization the first run was always seeded by an empty DAG without any edges (structure and REV-structure MCMC) or a random node order (order MCMC). To obtain an informed initialization we always performed a greedy search in the space of node orders in advance. We seeded the second run of order MCMC with this order and the other two samplers with the most likely DAG consistent with that order.

Except for the analysis of cytometric data in the last subsection, we will focus our comparison on data sets already used by Friedman and Koller (2003) to compare the performances of structure MCMC and order MCMC. We decided to do so as the analysis of these data sets revealed that the fundamental drawback of the structure MCMC sampler is the insufficient (too slow) convergence. We will show that integrating the REV move into conventional structure MCMC improves the convergence substantially and leads to a performance that is equivalent to order MCMC.

5.2 Evaluation criteria

All three MCMC sampling schemes (structure, order, and the new REV-structure MCMC) generate a directed acyclic graph (DAG) sample $\mathcal{M}_1, \ldots, \mathcal{M}_T$. Usually the next step is to search for features that are common to most of the graphs in the sample. Edge (relation) features indicate the presence of a particular directed or undirected edge in a DAG or its completed partially directed acyclic graph (CPDAG) representation (see Sect. 2). More formally, an edge (relation) feature F is a binary indicator variable over a space of graphs (DAGs or CPDAGs), which is 1 if the edge feature is present in the graph, and 0 otherwise. In this article we will focus on *directed edge (relation) features* F^D. There is a *directed edge (relation) feature* between X_i and X_j in the graph $\mathcal{G}$, symbolically $F_{ij}^D(\mathcal{G}) = 1$, if there is a directed edge pointing from X_i to X_j in $\mathcal{G}$.

From Sect. 2 we know that there are equivalence classes of Bayesian networks scoring the same marginal likelihood. Equivalence classes can be represented by CPDAGs, which comprise both directed and undirected edges. For computing the directed edge features of a CPDAG we interpret each undirected edge as a superposition of two directed edges, pointing in opposite directions. When no explicit prior knowledge is given, we convert a trajectory of DAGs into the trajectory of corresponding CPDAGs to allow for the intrinsic symmetry of the marginal likelihood, and we compute the edge features from these CPDAGs. Note that the symmetry of the posterior distribution is broken when explicit prior knowledge is included; in this case, the edge features are computed directly from the trajectory of DAGs. An estimator for the posterior probabilities of an edge feature F given the data $\mathcal{D}$ is given by the fraction of graphs in the sample that contain the edge feature of interest. For each edge feature F the corresponding estimator is given by:

$$\widehat{P(F|\mathcal{D})} = \frac{1}{T}\sum_{t=1}^{T} F(\mathcal{G}_t). \tag{36}$$

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. We assume that $e_{ij} = 1$ indicates that there is a edge feature between X_i and X_j in the true graph, while $e_{ij} = 0$ indicates that this edge feature is not given in the true graph. The Bayesian network approach outputs a posterior probability estimate $\widehat{P(F_{ij}|\mathcal{D})}$ for each edge feature e_{ij}.

Let $\epsilon(\theta) = \{e_{ij} | \widehat{P(F_{ij}|\mathcal{D})} > \theta\}$ denote the set of all edge features whose posterior probability estimates exceed a given threshold θ. Given θ the number of true positive (TP), false positive (FP), and false negative (FN) edge feature findings can be counted, and the *sensitivity* $S = TP/(TP + FN)$ and the *inverse specificity* $I = FP/(TN + FP)$ can be computed. But rather than selecting an arbitrary value for the threshold θ, this procedure can be repeated for several values of θ and the ensuing sensitivities can be plotted against the corresponding inverse specificities. This gives the *receiver operator characteristic* (ROC) curve. A quantitative measure for the learning performance can be obtained by integrating the ROC curve so as to obtain the area under the ROC curve, which is usually referred to as AUROC_1 value. We note that larger AUROC_1 values indicate a better learning performance, whereby 1 is an upper limit and corresponds to a perfect estimator, while 0.5 corresponds to a random estimator. The right range of the inverse specificity is usually of no practical interest, as the number of false positive (FP) counts, in absolute terms, would be unreasonably high. For this reason it is often preferred to compute the area under the ROC curve up to a small, pre-specified upper limit on the inverse specificity: $I < \epsilon$. This yields the AUROC_ϵ score. See Husmeier (2003) for further details.

5.3 Little domain diagnostics

In this first subsection we will compare the directed edge feature posterior probability estimates (see Sect. 5.2) obtained from the three MCMC samplers with the true posterior probabilities (computed by exhaustive enumeration of all possible graphs using the graph prior given in (3)) for some little domains with $N = 5$ variables only to demonstrate that the order MCMC estimates are (slightly) biased while the structure MCMC and the new REV-structure MCMC estimates are not. To this end, we randomly selected $N = 5$ variables from various data sets available from the UCI repository (Newman et al. 1998), and then sampled data sets with different numbers of observations (cases) m from the available data. Figure 2 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 available from UCI consists of $N = 16$ discrete variables and $m = 435$ observations, whereby plenty of values are missing. We excluded observations where the value of one of the 5 selected variables was missing. The complete FLARE data set available from UCI consists of $N = 13$ discrete variables and $m = 1389$ observations. From the latter selection we always excluded variables having more than 3 different discrete values.

Each panel in Fig. 2 refers to a data set, and the true posterior probabilities of the directed edge features are plotted against the x-axis. Parallel to the x-axis there is a reference line for each of the three MCMC samplers, and the deviations of the estimates are plotted around these lines, whereby the lines themselves correspond to zero deviations. Note that we performed two independent MCMC runs with each MCMC sampler and that we plotted the deviations of both runs in the same panel.

It can clearly be seen from Fig. 2 that the order MCMC estimates are systematically biased while the estimates obtained by structure and REV-structure MCMC are not. It appears that order MCMC tends to underestimate the true marginal posterior probabilities for domains with few observations.

5.4 The Boston HOUSING data

In this subsection we deal with the question whether the convergence of REV-structure MCMC is better than the convergence of structure MCMC. To this end, we ran the three MCMC samplers on the Boston HOUSING data from the UCI repository consisting of $N = 14$ continuous variables and $m = 506$ observations. The Boston HOUSING data from the UCI repository was already used by Friedman and Koller (2003) as an example where the order MCMC estimates from different simulations are well-correlated while the structure MCMC estimates are not. To confirm this finding and to show that the new REV-structure MCMC estimates are well-correlated too, we started for each of the three MCMC samplers four independent MCMC runs: two starting from the uniformed initialization (random node order or empty DAG) and two starting from the informed initialization (node order found by greedy search or the highest scoring DAG consistent with that order). Trace plots of the logarithmic scores can be found in Appendix 5. Both runs of the new REV-structure MCMC and order MCMC scheme reach the same plateau, whereas the empty seeded structure MCMC run gets stuck in a lower scoring region of the posterior landscape. We then compared the estimates of the posterior probabilities of the directed edge features obtained from those different runs. See Fig. 3 for some scatter plots comparing the posterior probability estimates obtained by independent and differently seeded runs.

From the scatter plots in Fig. 3 it can be seen that we obtain structure and order MCMC results that are comparable to the results of Friedman and Koller (2003). The order MCMC

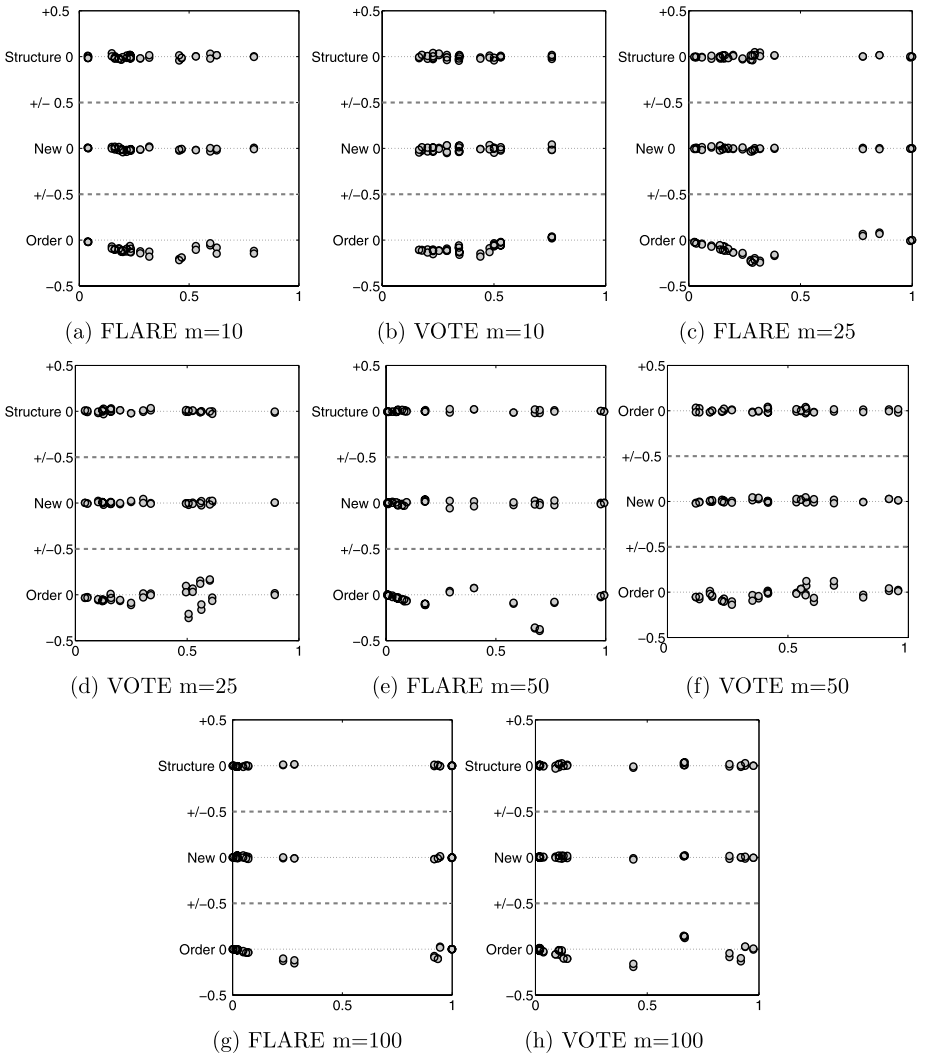


Fig. 2 Deviations between true and estimated directed edge feature posterior probabilities for different subsets of the Vote and Flare data. In each panel the true posteriors are plotted along the x-axis, and parallel to the x-axis there is a *thin reference line* for each of the three samplers. The *upper line* corresponds to structure, the *centre line* corresponds to the new REV-structure, and the *lower line* corresponds to order MCMC, whereby the lines correspond to a zero deviation from the true score. The *circles* 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 using the graph prior given in (3)

estimates are well-correlated for both seeds (panels (g) and (h)) and it can be seen from panel (i) that the estimates do not depend on the initialization. The empty DAG seeded runs of structure MCMC are not strongly correlated (panel (a)) but the greedy-search DAG seeded runs are (panel (b)). Furthermore, it can be seen from panel (c) that the initialization influences the results. There are several edge features that obtain a high posterior proba-

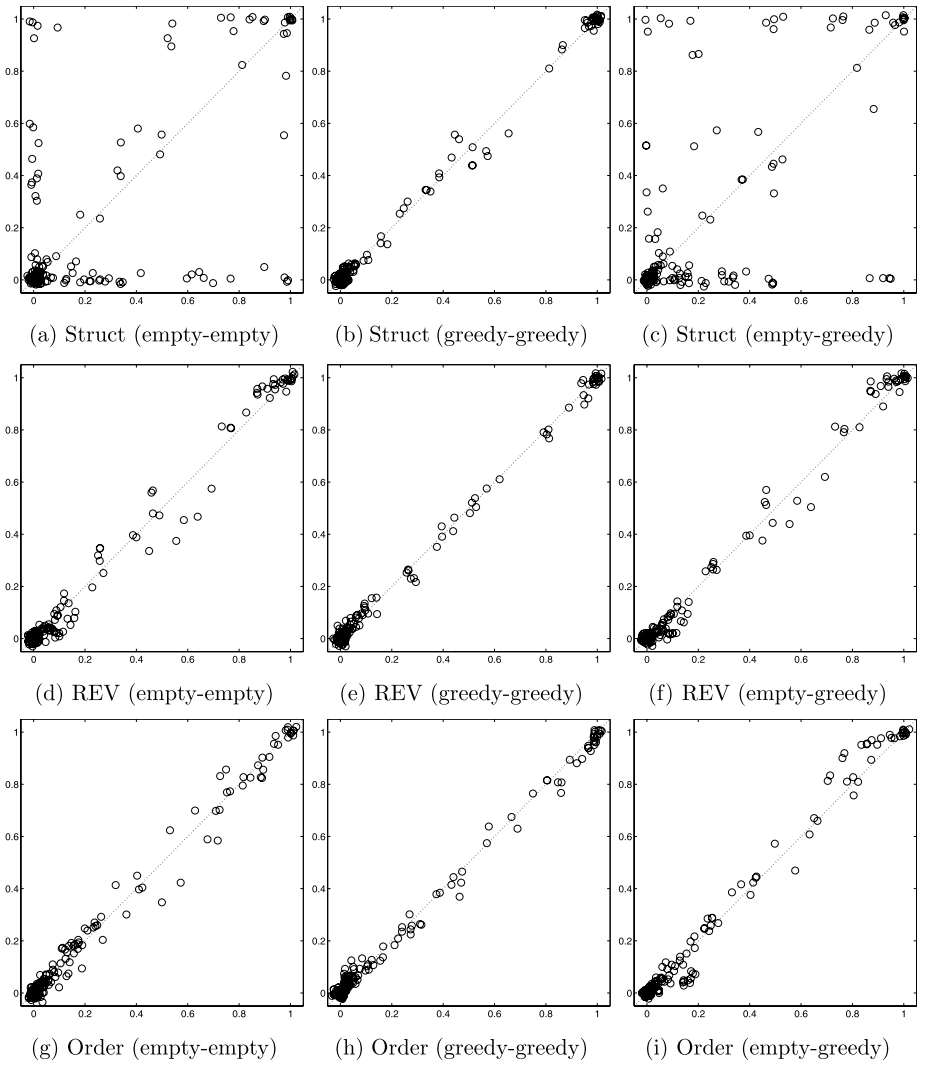
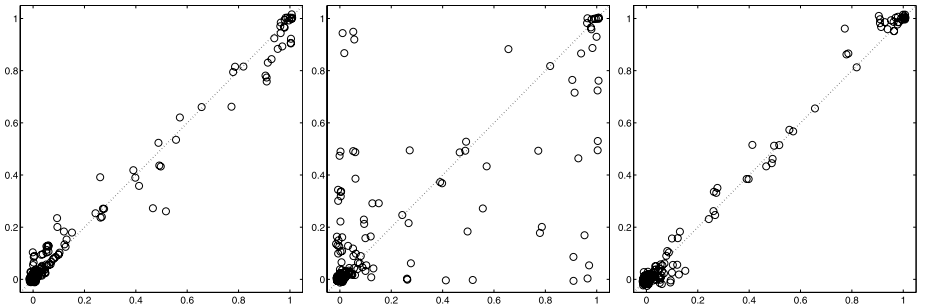


Fig. 3 Convergence control for the HOUSING data. Each *panel* corresponds to a MCMC sampler and gives a scatter plot of the posterior probability estimates for the directed edge features (see Sect. 5.2) either obtained by two independent but identically seeded runs (*left* and *centre column*) or obtained by differently seeded runs (*right column*). When comparing differently seeded runs the DAG samples of both identically seeded runs were combined, that is, the combined sample was used for estimation. 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

bility for both empty DAG seeded runs and a low posterior probability for the two greedy search seeded runs and vice-versa. The correlation of the REV-structure MCMC estimates is approximately as strong as the correlation of the order MCMC estimates. That is, REV-structure MCMC seems to converge for both initializations (panels (d) and (e)) and it can be seen from panel (f) that the estimates do not differ between the two different initializations.



(a) REV vs. Order (b) REV vs. Struct-empty (c) REV vs. Struct-greedy

Fig. 4 Convergence comparison for the HOUSING data using directed edge features. Panel (**a**) compares the estimates obtained by REV-structure MCMC and the estimates obtained by order MCMC. Panels (**b**) and (**c**) compare the REV-structure MCMC estimates with the estimates obtained by the empty DAG seeded (**b**) and greedy DAG seeded (**c**) structure MCMC runs. For order MCMC and REV-structure MCMC all four DAG samples were combined. For structure MCMC we had to distinguish between the two different seeds. Hence, only two independent DAG samples were combined in both cases. 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

Next, we wanted to cross-compare the results of the different samplers. To this end, we combined the DAG sample of all four REV-structure MCMC runs and all four order MCMC runs to obtain unique REV-structure MCMC and order MCMC directed edge feature posterior probability estimates and plotted them against each other. From the scatter plot in panel (a) in Fig. 4 it can be seen that the estimates of order and REV-structure MCMC differ only slightly. For structure MCMC only the DAG samples obtained by identically seeded runs can be combined, as the initialization influences convergence as well as estimates. The results obtained by greedy search seeded structure MCMC runs are very similar to the results obtained by REV-structure MCMC (see panel (c) in Fig. 4)), but the empty-seeded structure MCMC estimates differ drastically from the REV-structure MCMC results (see panel (b) in Fig. 4)). That is not surprising, as the empty seeded structure MCMC runs had not converged.

We can summarize that the new REV-structure MCMC and the order MCMC estimates independently of the initialization are both strongly correlated, and that the estimates obtained by order MCMC and REV-structure MCMC are very similar to each other. On the other hand, the results obtained by structure MCMC depend on the initialization. It appears that structure MCMC runs tend to get trapped in local maxima. Only when structure MCMC is seeded by a greedy-search DAG then we obtain results that are consistent with the order and REV-structure MCMC results. This demonstrates that REV-structure MCMC (as an improved structure MCMC sampler) is superior to the classical structure MCMC in terms of convergence. In Appendix 4 we present some results we obtained by running REV-structure MCMC simulations on the Boston Housing data set with different parameters p_R.

5.5 The ALARM network

In this subsection we continue with the convergence evaluation but also compare the learning performances of the three MCMC samplers on data sets generated from the well-known Alarm network (Beinlich et al. 1989). The true Alarm network is a directed acyclic graph

(DAG) that consists of $N = 37$ nodes (variables) and 46 directed edges. We note that the algorithms of Ellis and Wong (2006) and Eaton and Murphy (2007) are not applicable to this domain, as it possesses more than 30 nodes. We generated seven data sets from the Alarm network with different numbers of observations: $m = 25, 50, 100, 250, 500, 750$, and 1000. For each of the three MCMC samplers we then performed two independent MCMC runs (differing by the initialization) on each of these seven data sets. The convergence of the three MCMC samplers can be evaluated by scatter plots of the individual edge feature posterior probability estimates for the two independent MCMC runs. Figures 5 and 6 show for each combination of MCMC sampler and Alarm data set size m a scatter plot of the 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 there is a high degree of convergence between the differently seeded runs. But except for the smallest Alarm test data set ($m = 25$) the structure MCMC runs do not converge to a sufficient degree. It seems that the number of completely differing posterior probability estimates is increasing in the data set size m. That is probably due to the fact that the posterior probability landscape becomes more rugged with increasing m, which hampers convergence. Furthermore it can be observed in the scatter plots that the inference uncertainty is reduced for large data set sizes; the posterior probability becomes more bimodal, with values clustering around 0 and 1.

The learning performance of the three MCMC samplers can be compared in terms of AUROC values as the true network of the ALARM domain is known. For each of the three samplers there are directed edge feature posterior probabilities from two independent MCMC runs available. From these estimates we can compute AUROC_ϵ values by ranking and comparing them with the true Alarm network as explained in Sect. 5.2. So we obtain two AUROC_ϵ values for each combination of MCMC sampler and data set size. Figure 7 shows the AUROC_ϵ values for $\epsilon = 1, 0.1, 0.05$, and 0.01. The first trend that can be seen from these AUROC plots is that the AUROC_ϵ values increase in the number of observations m. It is not surprising that the AUROC values of REV-structure and order MCMC differ insignificantly for the two initializations, as the corresponding posterior probability estimates are almost the same (see Figs. 5 and 6). It is worth mentioning that the learning performance of REV-structure and order MCMC is approximately the same. That is for each data set size m REV-structure and order MCMC yield (independently of the initialization) approximately the same AUROC_ϵ values. On the other hand, the structure MCMC AUROCs depend on the initialization and are lower than those of order MCMC and REV-structure MCMC for bigger sample sizes m. It can be summarized that REV-structure and order MCMC yield the same convergence level as well as the same learning performance (in terms of AUROC values) on the ALARM data sets. The Structure MCMC estimates are not well-correlated and the estimates yield lower AUROC values and so a worse learning performance overall. This again demonstrates that the new reversal move upgrades the classical structure MCMC substantially. Trace plots of the logarithmic scores show consistently the following trend: The new REV-structure MCMC sampler and the order MCMC sampler show very similar trace plots, whereas the empty seeded structure MCMC runs are usually inferior. The logarithmic score trace plots for the Alarm data sets with $m = 750$ and $m = 1000$ can be found in Appendix 5.

5.6 The Raf-Mek-Erk signaling network

We finally discuss an application of our method to a problem related to systems biology. The objective is to reconstruct the Raf-Mek-Erk pathway, a cellular signaling network de-

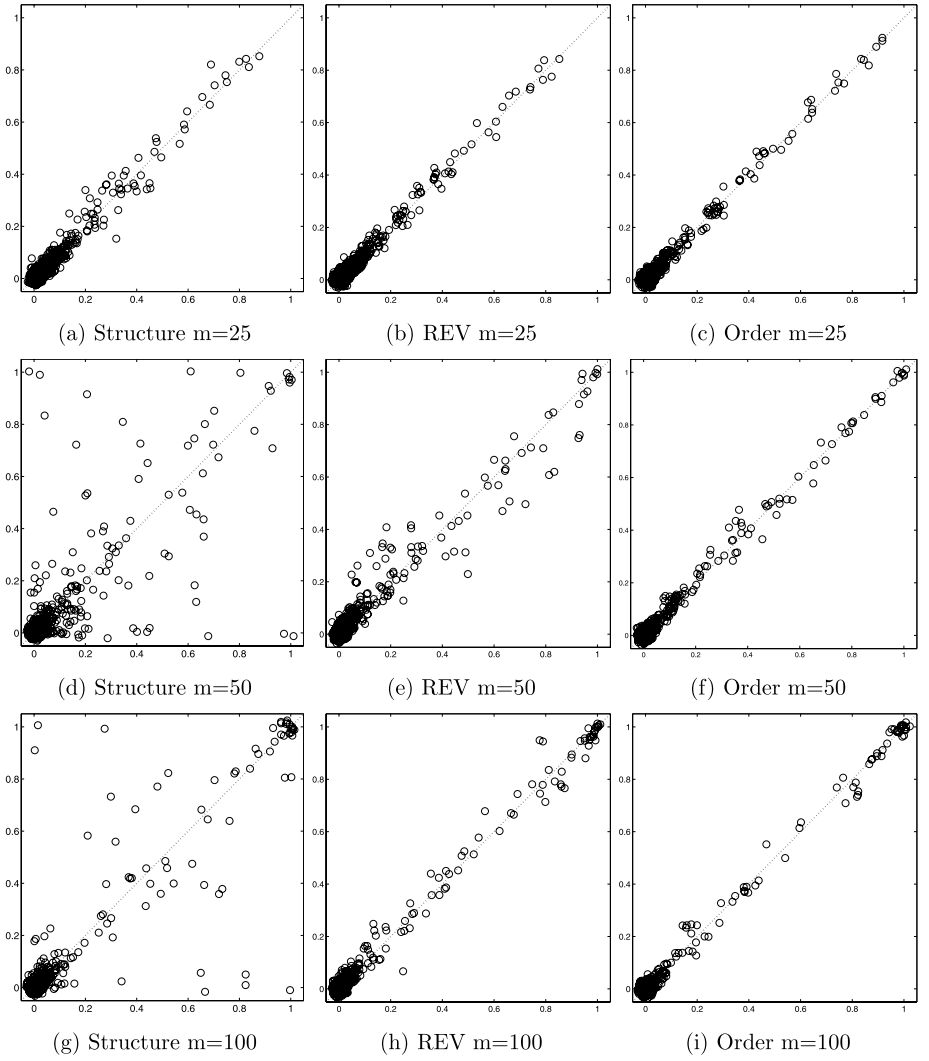
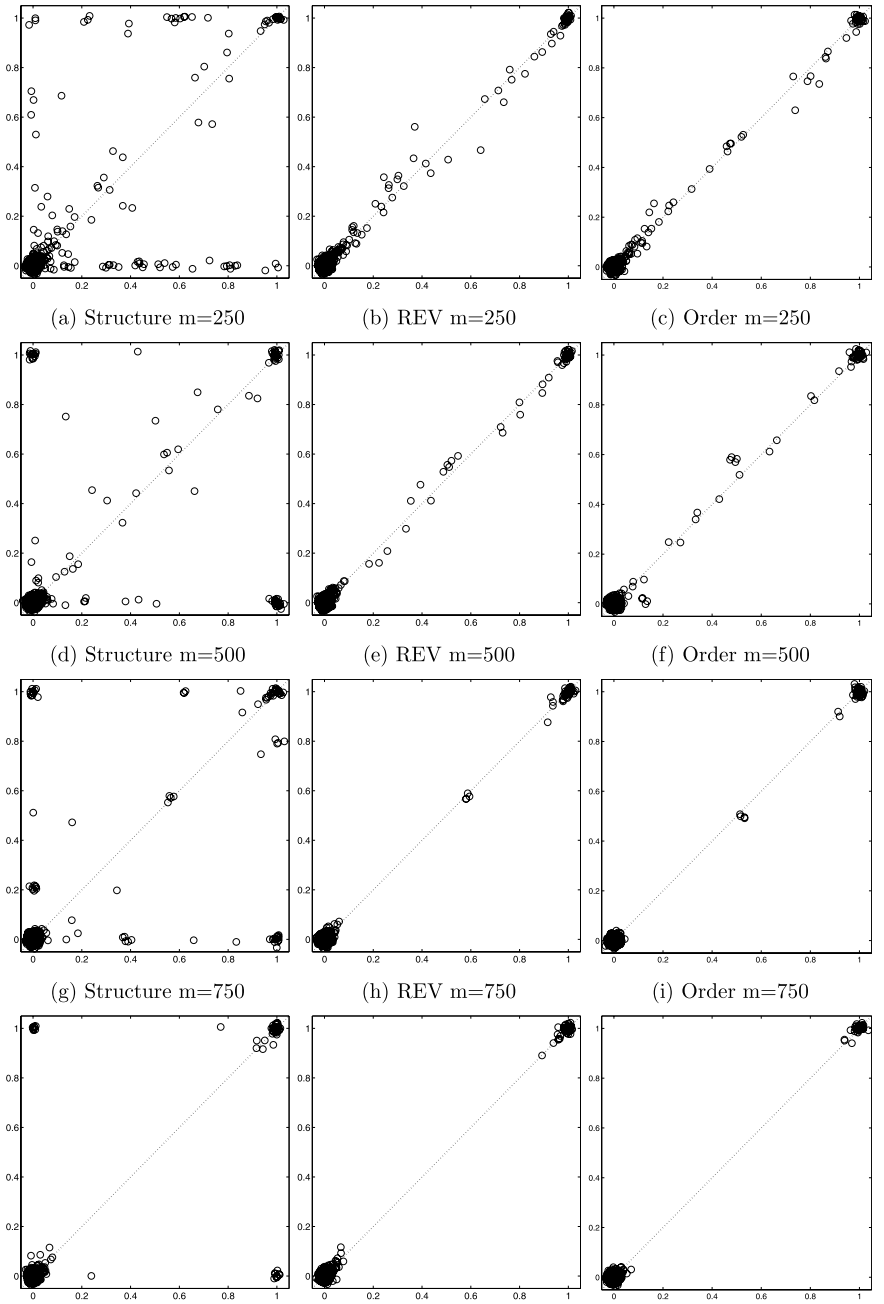


Fig. 5 Convergence control via scatter plots. Scatter plots that compare the posterior probability estimates of the directed edge features (see Sect. 5.2) on Alarm network data sets of different size m. *Left column*: structure MCMC, *centre column*: REV-structure MCMC, and *right column*: order MCMC. In each plot every *circle* corresponds to a single directed edge feature and its coordinates were estimated from two DAG samples of size 1000 obtained from two independent MCMC runs. The x coordinates were estimated from a sample obtained from an empty seeded (structure and REV-structure MCMC) or randomly seeded (order MCMC) run, and the y coordinates were estimated from a sample obtained from 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

scribing the interaction of eleven phosphorylated proteins and phospholipids. Raf is a critical signaling protein involved in regulating cellular proliferation in human immune system cells. The deregulation of the Raf-Mek-Erk pathway can lead to carcinogenesis, and this pathway has therefore been extensively studied in the literature; see e.g. Sachs et al.



(a) Structure m=250 (b) REV m=250 (c) Order m=250

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

(g) Structure m=750 (h) REV m=750 (i) Order m=750

(j) Structure m=1000 (k) REV m=1000 (l) Order m=1000

Fig. 6 Convergence control via scatter plots continued. See caption of Fig. 5 for explanations

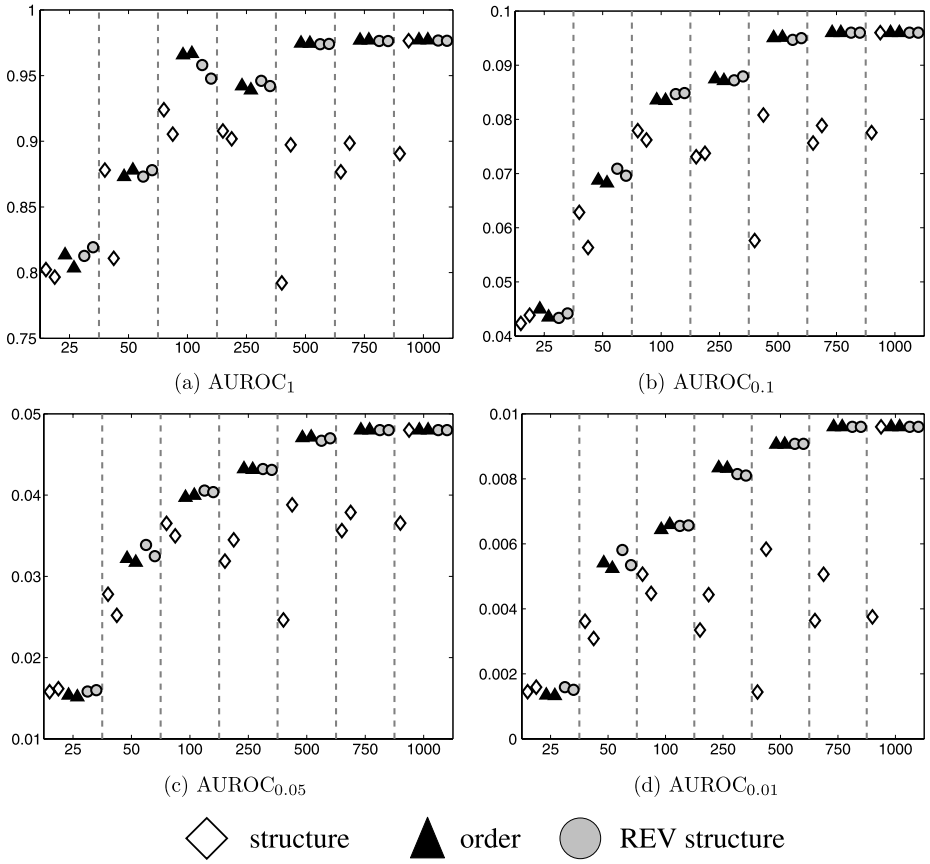


Fig. 7 Comparison of the learning performance for the ALARM network data in terms of AUROC$_\epsilon$ values using directed edge features. Each panel corresponds to a particular ϵ value and can be interpreted as follows: Along the x-axis there is a 'column' for each of the seven different sample sizes m and the corresponding AUROC$_\epsilon$ values have been plotted on the y-axis. In each *column* the structure MCMC AUROCs have been plotted as slightly leftwards shifted *white diamonds*, the order MCMC AUROCs have been plotted as centred *black triangles*, and the REV-structure MCMC AUROCs have been plotted as slightly rightwards shifted *grey circles*. For each sampling scheme the AUROCs corresponding to greedy-search seeded runs stand to the left

(2005). Our work is based on the work of Sachs et al. (2005), who have applied intracellular multicolour flow cytometry experiments to measure protein concentrations related to the Raf-Mek-Erk pathway. To investigate the effect of including explicit prior knowledge, we repeated simulations in the vein of those reported in Werhli and Husmeier (2007). We used the same five data sets as the authors, which had been subsampled from the observational flow cytometry data of Sachs et al. (2005). In addition, we extracted the first ten observations of each of these data sets of size $m = 100$ to obtain five data sets with a smaller sample size of $m = 10$. As prior knowledge we included information extracted from the Kyoto Encyclopedia of Genes and Genomes (KEGG) database (Kanehisa 1997; Kanehisa and Goto 2000, and Kanehisa et al. 2006) as in Werhli and Husmeier (2007), giving us the same prior knowledge matrix $\mathcal{B}$. In addition, as an alternative form of prior knowledge we used the Raf-Mek-Erk gold standard network from Sachs et al. (2005) itself in the following way: we set an element $\mathcal{B}_{ij}$ of the prior knowledge matrix $\mathcal{B}$ to $1 - \varepsilon$ if

the edge $X_i \to X_j$ was contained in the gold standard network; otherwise it was set to ε. Following this scheme, we computed three further prior knowledge matrices with different confidence levels: $\varepsilon = 0.1, 0.4, 0.45$. From each of these four prior knowledge matrices, a prior probability distribution over DAGs $\mathcal{M}$ can be obtained, as described in Werhli and Husmeier (2007):

$$P(\mathcal{M}|\beta, \mathcal{B}) = \frac{e^{-\beta \cdot E(\mathcal{M}|\mathcal{B})}}{\sum_{\tilde{\mathcal{M}} \in \Omega} e^{-\beta \cdot E(\tilde{\mathcal{M}}|\mathcal{B})}} \tag{37}$$

where $E(\mathcal{M}|\mathcal{B}) = \sum_{i=1}^{N} \sum_{j=1}^{N} |\mathcal{M}_{ij} - \mathcal{B}_{ij}|$, $\mathcal{M}_{ij}$ is 1 if the edge $X_i \to X_j$ is contained in $\mathcal{M}$ and 0 otherwise, and Ω represents the set of all valid DAGs over the domain.

The hyperparameter β of this prior distribution was kept fixed at the value inferred in Werhli and Husmeier (2007) ($\beta = 4$). We note that this graph prior is a modular prior, that is, the prior can be factorized into a product where each factor corresponds to a node:

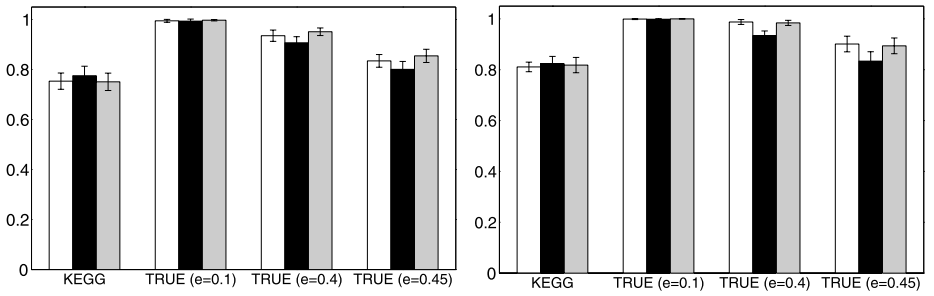
$$P(\mathcal{M}|\beta, \mathcal{B}) \propto \prod_{n=1}^{N} e^{-\beta \cdot (\sum_{k \in \pi_n} \mathcal{B}_{kn} + \sum_{k \notin \pi_n} (1 - \mathcal{B}_{kn}))} \tag{38}$$

where N is the number of network nodes, and π_n is the parent set of node X_n in DAG $\mathcal{M}$. As before, on each of the 10 cyctometric data sets we ran two independent and differently seeded MCMC runs with each of the three MCMC samplers. We quantified the prediction accuracy in terms of AUROC scores by extracting directed edge features defined over the space of DAGs, and using the network published in Sachs et al. (2005) as a gold-standard for evaluation.

The results of this application are shown in Fig. 8. For the prior knowledge extracted from the KEGG database, which is only partially correct, there is no significant difference in performance between the three schemes. For the 'true' prior knowledge constructed from Sachs et al. (2005) with a high confidence level ($\varepsilon = 0.1$), there is also no significant difference between the schemes. This finding is to be expected, since for strong prior knowledge the distortional effect of order-MCMC will not affect the ordering of the marginal posterior probabilities of the edges (which defines the AUROC score). However, when the prior knowledge is weak ($\varepsilon = 0.4, 0.45$), the distortional effect of order MCMC was found to have a slight, yet significant detrimental effect, where the difference between order and REV-structure MCMC was significant at the $p = 0.01$ level, as computed from a paired t-test, for both sample sizes $m = 10$ and $m = 100$. This demonstrates that the systematic bias of order MCMC can negatively influence the learning performance when prior knowledge is explicitly included. Owing to the smaller number of nodes, convergence and mixing problems were less pronounced than in the two previous applications; hence structure MCMC is not outperformed by our new scheme.

6 Conclusion

Our paper contributes to recent research on sampling Bayesian network structures from the posterior distribution with MCMC. Two principled paradigms have been applied in the past. Structure MCMC, first proposed by Madigan and York (1995), defines a Markov chain in the space of graph structures by applying basic operations to individual edges of the graph, like the creation, deletion or reversal of an edge. Alternatively, order MCMC, proposed by Friedman and Koller (2003), defines a Markov chain in the space of node orders. While the



(a) Raf-Mek-Erk data + Prior, m=10 (b) Raf-Mek-Erk data + Prior, m=100

Fig. 8 Effect of including prior knowledge for the reconstruction of the Raf-Mek-Erk pathway from flow cytometry data. The histograms show mean AUROC scores based on the gold-standard Raf-Mek-Erk pathway of Sachs et al. (2005). Five independent subsamples of the observational flow cytometry protein concentration from Sachs et al. (2005) were used. *Left panel*: sample size $m = 10$. *Right panel*: sample size $m = 100$. *Error bars* show standard deviations computed from the five independent replications. The different *shadings* represent different MCMC schemes. *White*: structure MCMC. *Black*: order MCMC. *Grey*: REV-structure MCMC. The different *groups of histograms* are related to different forms of prior knowledge. *From left to right*: prior knowledge from KEGG, as used in Werhli and Husmeier (2007), gold-standard network from Sachs et al. (2005) with different confidence scores: $\varepsilon = 0.1, 0.4, 0.45$

second approach has been found to substantially improve the mixing and convergence of the Markov chain, it does not allow an explicit specification of the prior distribution over graph structures or, to phrase this differently, it incurs a distortion of the specified prior distribution as a consequence of the marginalization over node orders. This distortion can lead to problems for applications in systems biology, where owing to the limited number of experimental conditions the integration of biological prior knowledge into the inference scheme becomes desirable. Different approaches and modifications have been developed in the literature to address this shortcoming (e.g. see Ellis and Wong 2006 and Eaton and Murphy 2007). Unfortunately, these methods incur extra computational costs and are not practically viable for inferring large networks with more than 20 to 30 nodes. There have been suggestions of how to improve the classical structure MCMC approach (e.g. see Castelo and Kočka 2003 and Mansinghka et al. 2006), but these methods only partially address the convergence and mixing problems, as discussed in the Introduction section.

In the present paper we have proposed a novel structure MCMC scheme, which augments the classical structure MCMC method of Madigan and York (1995) with a new edge reversal move. The idea of the new move is to resample the parent sets of the two nodes involved in such a way that the selected edge is reversed subject to the acyclicity constraint. The proposal of the new parent sets is done effectively by adopting ideas from importance sampling; in this way faster convergence is effected. For methodological consistency, and in contrast to Castelo and Kočka (2003), we have properly derived the Hastings factor, which is a function of various partition functions that are straightforward to compute. The resulting Markov chain is reversible, satisfies the condition of detailed balance, and is hence theoretically guaranteed to converge to the desired posterior distribution.

For our experimental evaluation we focused on various data sets from the UCI repository (Newman et al. 1998), such as Vote, Flare, Boston Housing, and Alarm, which were already used by Friedman and Koller (2003) to demonstrate that order MCMC outperforms structure MCMC. Our experimental results show that integrating the novel edge reversal move yields a substantial improvement of the resulting MCMC sampler over classical structure MCMC,

with convergence and mixing properties that are similar to those of the order MCMC sampler of Friedman and Koller (2003).

To demonstrate the distortional effect of order MCMC, we extended our empirical evaluation by analysing flow cytometry protein concentrations from the Raf-Mek-Erk signaling pathway in the vein of Werhli and Husmeier (2007). The experimental results showed that our new REV-structure sampling scheme can lead to a slight yet significant performance improvement over order MCMC when explicit prior knowledge is integrated into the learning scheme. This suggests that the avoidance of any systematic distortion of the prior probability on network structures renders the proposed REV-structure MCMC sampler preferable to order MCMC, especially for those contemporary systems biology applications where the number of experimental conditions relative to the complexity of the investigated system, and hence the weight of the likelihood, is relatively low, and explicit prior knowledge about network structures from publicly accessible data bases is included (Imoto et al. 2003; Nariai et al. 2005; Imoto et al. 2006, and Werhli and Husmeier 2007).

Acknowledgements The Centre for Systems Biology at Edinburgh (CSBE) is a Centre for Integrative Systems Biology (CISB) funded by the BBSRC and EPSRC. Parts of the work reported in this paper were carried out while Marco Grzegorczyk was supported by the "Bioforschungsband" of the University of Dortmund. Dirk Husmeier is supported by the Scottish Government Rural and Environment Research and Analysis Directorate (RERAD). Finally, we would like to thank three anonymous referees for their constructive comments on an earlier version of this paper.

Appendix 1: The REV move algorithm

In this first appendix a brief and concise summary of the new edge reversal (REV) move is given. We assume that for each of the N domain variables $X_1, \ldots, X_N$ a list of the scores of all valid parent sets is available, i.e. has been precomputed and stored. If necessary, e.g. for domains with a large number of variables, the potential parent sets of the nodes can be restricted as described in Friedman and Koller (2003) (see Sect. 3.3).

The new edge reversal move procedure:

- Given the current DAG $\mathcal{M}$, perform a new edge reversal (REV) move with probability p_R. If a new REV move is performed, then proceed as follows:
- Determine and store the number of edges $N^{\dagger}$ in $\mathcal{M}$ and randomly select one of these edges. This edge points from node X_i to node X_j, symbolically $X_i \rightarrow X_j$.
- Store the current parent set π_j of node X_j in $\mathcal{M}$, and then remove from $\mathcal{M}$ all edges pointing either to X_i or to X_j to obtain the DAG $\mathcal{M}_{\odot}$, and store this DAG $\mathcal{M}_{\odot}$.
- Determine and store the sets $\mathcal{D}(X_i|\mathcal{M}_{\odot})$ and $\mathcal{D}(X_j|\mathcal{M}_{\odot})$ of all descendant nodes of X_i and X_j in $\mathcal{M}_{\odot}$.
- Run through the list of precomputed scores of node X_i and mark all parent sets in the list that do not include node X_j. Extract the unmarked parent sets, and in a second step remove all those parent sets from the extracted ones which contain a node out of the set $\mathcal{D}(X_i|\mathcal{M}_{\odot})$. From the remaining parent sets sample a new parent set $\tilde{\pi}_i$ for node X_i according to (20) and store the sum of those remaining scores as the partition function $Z^{\star}(X_i|\mathcal{M}_{\odot}, X_j)$.
- Add the new parent set $\tilde{\pi}_i$ to node X_i to obtain the DAG $\mathcal{M}_{\oplus}$. Afterwards determine the set $\mathcal{D}(X_j|\mathcal{M}_{\oplus})$ of all descendants of node X_j in $\mathcal{M}_{\oplus}$.

- Run through the list of precomputed scores of node X_j and extract all those parent sets from the list that contain a node out of the set $\mathcal{D}(X_j|\mathcal{M})$. From the remaining parent sets sample a new parent set $\tilde{\pi}_j$ for node X_j according to (21) and store the sum of those remaining scores as the partition function $Z(X_j|\mathcal{M}_\oplus)$.
- Add the new parent set $\tilde{\pi}_j$ to node X_j in $\mathcal{M}_\oplus$ to obtain the DAG $\tilde{\mathcal{M}}$ which is finally proposed by the REV move. Determine and store the number of edges $\tilde{N}^\dagger$ in $\tilde{\mathcal{M}}$.
- For computing the acceptance probability continue as follows:
- Run through the list of precomputed scores of node X_j and mark all parent sets in the list that do no contain node X_i. Extract the unmarked parent sets and in a second step remove all those parent sets from the extracted ones which contain a node out of the set $\mathcal{D}(X_j|\mathcal{M}_\odot)$. Afterwards store the sum of those remaining scores as the partition function $Z^\star(X_j|\mathcal{M}_\odot, X_i)$, and then add π_j as new parent set for node X_j to obtain the DAG $\tilde{\mathcal{M}}_\oplus$. Determine the set $\mathcal{D}(X_i|\tilde{\mathcal{M}}_\oplus)$ of all descendants of node X_i in $\tilde{\mathcal{M}}_\oplus$.
- Run through the list of precomputed scores of node X_i and extract all those parent sets from the list that contain a node out of the set $\mathcal{D}(X_i|\tilde{\mathcal{M}}_\oplus)$. Compute the sum of the remained scores as the partition function $Z(X_i|\tilde{\mathcal{M}}_\oplus)$.
- Use $N^\dagger$, $\tilde{N}^\dagger$, $Z^\star(X_i|\mathcal{M}_\odot, X_j)$, $Z^\star(X_j|\mathcal{M}_\odot, X_i)$, $Z(X_j|\mathcal{M}_\oplus)$, and $Z(X_i|\tilde{\mathcal{M}}_\oplus)$ to compute the acceptance probability of the move from $\mathcal{M}$ to $\tilde{\mathcal{M}}$ according to (31).
- If the move is accepted replace the current DAG $\mathcal{M}$ by $\tilde{\mathcal{M}}$. Otherwise, leave the current state unchanged.

We note that in our current implementation of the new edge reversal (REV) move the algorithms of Giudici and Castelo (2003) are employed for determining the descendant nodes of X_i and X_j as well as for updating the ancestor matrix whenever necessary. More precisely, the ancestor nodes of the i-th (j-th) domain node are given by the non-zero entries of the i-th (j-th) column of the *ancestor matrix*. Adding a parent set to an orphaned node can be seen as a series of single edge additions, and correspondingly, removing a parent set can be seen as a series of single edge deletions. Algorithms for efficiently updating the ancestor matrix after single edge additions and deletions are given in Giudici and Castelo (2003).

Appendix 2: Proof of theorem

Firstly, let π_i and π_j denote the parent sets of X_i and X_j in $\mathcal{M}$ in the first step of the new edge reversal move that reverses the edge $X_i \to X_j$ given in $\mathcal{M}$. Then per definition we have: $X_i \in \pi_j$ and it immediately follows from the acyclicity of $\mathcal{M}$ that $X_j \notin \pi_i$. Hence, in summary we have:

$$X_i \in \pi_j \ \wedge \ X_j \notin \pi_i. \tag{39}$$

Secondly, we note that $X_j \in \tilde{\pi}_i$ in the second step of the new edge reversal move together with $\delta(\mathcal{M}_\oplus^{X_j \leftarrow \tilde{\pi}_j}) = 1$ in the third step implies $X_i \notin \tilde{\pi}_j$, because if X_i was a parent node of X_j in $\tilde{\mathcal{M}}$, then there would be the directed cycle $X_j \to X_i \to X_j$ in $\tilde{\mathcal{M}}$. Therefore, we have:

$$X_j \in \tilde{\pi}_i \ \wedge \ X_i \notin \tilde{\pi}_j. \tag{40}$$

Theorem *For each new reversal (REV) move leading from a DAG $\mathcal{M}$ to a DAG $\tilde{\mathcal{M}}$ by reversing the edge $X_i \to X_j$, there is exactly one inverse new reversal move leading back from $\tilde{\mathcal{M}}$ to $\mathcal{M}$. The inverse REV move selects the edge $X_j \to X_i$ in $\tilde{\mathcal{M}}$ for edge reversal, and orphaning both nodes X_i and X_j in the first step yields the DAG $\tilde{\mathcal{M}}_\odot$. In the second*

step the parent set π_j of X_j in $\mathcal{M}$ is sampled and assigned as the new parent set for X_j in $\tilde{\mathcal{M}}_\odot$, which gives the DAG $\tilde{\mathcal{M}}_\oplus$. Finally, in the third step the parent set π_i of X_i in $\mathcal{M}$ is sampled and assigned as the new parent set for X_i in $\tilde{\mathcal{M}}_\oplus$, which gives the DAG $\mathcal{M}$.

Proof of Theorem We know that an inverse REV move that leads back from $\tilde{\mathcal{M}}$ to $\mathcal{M}$ has to change the parent sets of both nodes X_i and X_j. This can be accomplished only if there is an edge between X_i and X_j in $\tilde{\mathcal{M}}$, so that this edge can be selected for edge reversal. It follows from (40) that the edge $X_j \to X_i$ is present in $\tilde{\mathcal{M}}$, and there can be no other edge between X_i and X_j in $\tilde{\mathcal{M}}$. Hence, the edge $X_j \to X_i$ must be selected for reversal from a uniform distribution over all edges in $\tilde{\mathcal{M}}$, and as this is the sole edge connecting X_i and X_j in $\tilde{\mathcal{M}}$, it follows that there is no other edge in $\tilde{\mathcal{M}}$ whose selection could give an inverse REV move.

Having selected the edge $X_j \to X_i$ for reversal in $\tilde{\mathcal{M}}$, in the first step both nodes X_i and X_j are orphaned. This yields the DAG: $\tilde{\mathcal{M}}_\odot := \tilde{\mathcal{M}}^{\{X_i, X_j\} \leftarrow \emptyset}$. From (23) it follows:

$$\tilde{\mathcal{M}}_\odot := \tilde{\mathcal{M}}^{\{X_i, X_j\} \leftarrow \emptyset} = (\mathcal{M}^{X_i \leftarrow \tilde{\pi}_i, X_j \leftarrow \tilde{\pi}_j})^{\{X_i, X_j\} \leftarrow \emptyset} = \mathcal{M}^{\{X_i, X_j\} \leftarrow \emptyset} \tag{41}$$

and it can be seen that the DAG $\tilde{\mathcal{M}}_\odot$ differs from $\mathcal{M}$ by the parent sets of X_i and X_j only.

In the next two steps, new parent sets are sampled from (20) and (21), and assigned to X_j and X_i, respectively.

The REV move can lead back to $\mathcal{M}$ only if the parent sets π_i and π_j of X_i and X_j in $\mathcal{M}$ are sampled and assigned as new parent sets in $\tilde{\mathcal{M}}_\odot$. So, there is only one possible inverse REV move which has to assign these parent sets to the nodes X_i and X_j. All that remains to be shown is that the parent sets π_i and π_j of X_i and X_j can actually be sampled and assigned by a REV move which reverses the edge $X_j \to X_i$ in $\tilde{\mathcal{M}}$.

In the second step the DAG $\tilde{\mathcal{M}}_\odot$ is considered, and the inverse REV move has to sample a new valid parent set containing X_i for node X_j. Due to (39) π_j contains X_i, and it follows from (23):

$$\tilde{\mathcal{M}}_\odot^{X_j \leftarrow \pi_j} = (\mathcal{M}^{\{X_i, X_j\} \leftarrow \emptyset})^{X_j \leftarrow \pi_j} = \mathcal{M}^{X_i \leftarrow \emptyset}, \tag{42}$$

so that $\tilde{\mathcal{M}}_\oplus := \tilde{\mathcal{M}}_\odot^{X_j \leftarrow \pi_j}$ is actually acyclic, as it is a subgraph of the valid DAG $\mathcal{M}$. Therefore π_j can be sampled as the new parent set for X_j in the second step of the inverse move. Finally, in the third step the DAG $\tilde{\mathcal{M}}_\oplus$ is considered, and the inverse REV move has to sample a new valid parent set for node X_i from (21). The parent set π_i can be sampled for X_i as

$$\tilde{\mathcal{M}}_\oplus^{X_i \leftarrow \pi_i} = (\tilde{\mathcal{M}}_\odot^{X_j \leftarrow \pi_j})^{X_i \leftarrow \pi_i} = (\mathcal{M}^{X_i \leftarrow \emptyset})^{X_i \leftarrow \pi_i} = \mathcal{M} \tag{43}$$

so that assigning the parent set π_i to X_i in $\tilde{\mathcal{M}}_\oplus$ yields the DAG $\mathcal{M}$, which is valid (acyclic) by assumption. □

Appendix 3: The conjugate edge reversal move

We illustrate the reversibility of the proposed REV move in Fig. 9. This figure shows the complementary backward move for the move illustrated in Fig. 1.

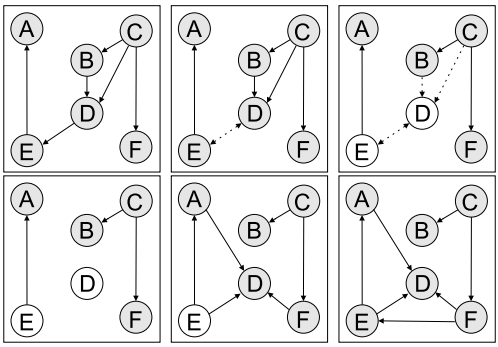
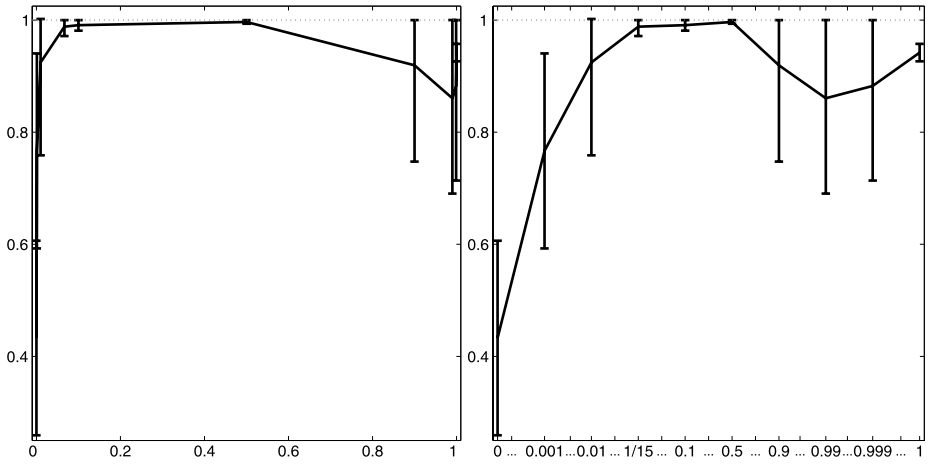


Fig. 9 The new edge reversal (REV) move conjugate to the REV move illustrated in Fig. 1. Step 1 (*top left*): We start with the directed acyclic graph $\tilde{\mathcal{M}}$ in the bottom right of Fig. 1. Our goal is to construct a move leadings back to $\mathcal{M}$ shown in the top left of Fig. 1. Step 2 (*top centre*): The edge $E \leftarrow D$ has to be selected for edge reversal, as the parent sets of the nodes E and D must be changed when the goal is to reobtain $\mathcal{M}$. Step 3 (*top right*): Deleting all (*dotted*) edges pointing to node D or node E gives the DAG $\tilde{\mathcal{M}}_{\odot}$. Step 4 (*bottom left*): This is the DAG $\tilde{\mathcal{M}}_{\odot} = \tilde{\mathcal{M}}^{\{D,E\}\leftarrow\emptyset}$. The nodes D and E have been orphaned, and as pointed out in (24) this DAG is identical to the DAG $\mathcal{M}_{\odot}$ in the bottom left of Fig. 1. Step 5 (*bottom centre*): From $Q(\pi_D|\tilde{\mathcal{M}}_{\odot}, E)$ (see (20)) we can sample the new parent set $\pi_D = \{A, E, F\}$ for node D, as this set contains node E, and assigning this new parent set to node D gives the valid DAG: $\tilde{\mathcal{M}}_{\oplus} = \tilde{\mathcal{M}}_{\odot}^{D\leftarrow\{A,E,F\}}$ shown here. This DAG $\tilde{\mathcal{M}}_{\oplus}$ differs from the DAG $\mathcal{M}$ only by the parent set of node E. More precisely, as pointed out in (26) orphaning node E in $\mathcal{M}$ would give the same DAG, symbolically: $\tilde{\mathcal{M}}_{\oplus} = \mathcal{M}^{E\leftarrow\emptyset}$. Step 6 (*bottom right*): Subsequently, the parent set $\pi_E = \{F\}$ can be sampled from $Q(\pi_E|\tilde{\mathcal{M}}_{\oplus})$ (see (21)) for node E, as assigning this parent set π_E to node E gives the valid DAG $\mathcal{M}_{\oplus}^{E\leftarrow\{F\}}$, which is identical to the DAG $\mathcal{M}$ in the top left of Fig. 1

Appendix 4: Varying the probability for a REV move

We obtained all experimental REV-structure MCMC results in Sect. 5 by using a fixed probability $p_R = 1/15$ for a new edge reversal (REV) move. In this fourth appendix we substantiate that the performance of the REV-structure MCMC sampler varies little over a quite range of parameters p_R. The results presented here were obtained by running several REV-structure MCMC simulations with ten different parameters p_R on the Boston HOUSING data (see Sect. 5.4). More precisely, for each of ten different parameters p_R we started five pairs of mutually independent REV-structure MCMC runs on the HOUSING data. For each of the five replications we seeded the first MCMC run with a 'full DAG', that is a randomly selected DAG having maximal number of edges (given the fan-in restriction of 3) and the second run with the greedy-search DAG (informed graph prior). Note that we initialized the first runs with randomly selected full DAGs (instead of empty DAGs), as the REV-structure MCMC sampler with $p_R = 1$ (performing exclusively new edge reversal moves) cannot start from a DAG without any edges: It would get immediately trapped in the initial (empty) DAG. For each p_R parameter and each of the 5 replications we then computed correlation coefficients from the directed edge (relation) feature posterior probabilities of the two independent MCMC runs. Figure 10 summarizes the results in a trace plot with errorbars. It can be seen from the trace plot that a high degree of convergence is reached as long as the parameter p_R is neither close to zero nor close to 1. For p_R values near zero too few REV moves are performed, which results in an insufficient degree of convergence. That is, the performance of the classical structure MCMC sampler ($p_R = 0$) is improved inessentially only. On the other hand, for p_R values near 1 the convergence level also decreases. This



(a) original x-axis (b) distorted x-axis

Fig. 10 Convergence control for different parameters p_R using the HOUSING data. For each of 10 different parameters p_R we performed 5-times two independent REV-structure MCMC runs on the HOUSING data. The first MCMC runs were seeded by randomly selected 'full DAGs', the second runs were seeded by the greedy-search DAG. For each p_R and each of the 5 replications we then computed correlation coefficients from the directed edge (relation) feature posterior probabilities of the two independent runs. In the plots the ensuing mean correlation coefficients are plotted against the parameters p_R. The errorbars are bounded by 1 and correspond to one standard deviation. As we selected non-equidistant p_R values, the original plot in panel (**a**) is not well arranged. Therefore we present the same results with a distorted x-axis in panel (**b**)

demonstrates that restricting mainly (or even exclusively) on new edge reversal moves does not yield an optimal performing MCMC sampler either. As already discussed at the end of Sect. 3.2, the Markov chain is not guaranteed to be ergodic for $p_R = 1$. For values very close to $p_R = 1$, mixing of the Markov chain might still be hampered by the fact that the deletion and creation of edges is effectively only achieved indirectly, via resampling the parent sets of nodes involved in an edge direction change. This suggests that p_R should not be too close to the boundaries $p_R = 0$ and $p_R = 1$. However, we note that according to our findings, these mixing and convergence difficulties only occur for extreme values of p_R close to the boundaries, and that the proposed method is quite robust with respect to a variation of p_R over a large range of its permissible domain. We also note that p_R could, in principle, be adjusted automatically in the burn-in phase of the Markov chain, which owing to the robustness we found was not implemented in our simulations.

Appendix 5: Trace plot diagnostics

In this fifth appendix we present various results of trace plot diagnostics, which we also considered to assess convergence of the different MCMC sampling schemes during the experimental evaluation in Sect. 5. For clarity, we decided to plot the two trace plots obtained from two independent and differently seeded runs for each of the three different MCMC samplers in separate but identically scaled panels. Furthermore, we left out the burn-in phase and started the trace plots with the logarithmic scores of the sampled DAGs. As the appearance of the different trace plots looks very similar, we present only the trace plots for the Boston Housing data set (see Fig. 11), and the Alarm data sets with $m = 750$ (see Fig. 12) and

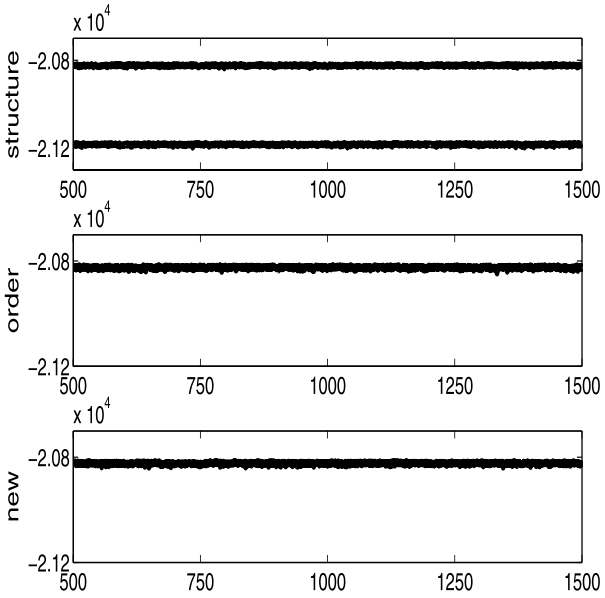


Fig. 11 Trace plots of the logarithmic scores of the DAGs sampled after the burn-in phase for the Boston Housing data set. The *upper panel* shows the trace plots for two structure MCMC runs, the *middle panel* shows the trace plots for two order MCMC runs, and the *lower panel* shows trace plots for two REV-structure MCMC runs

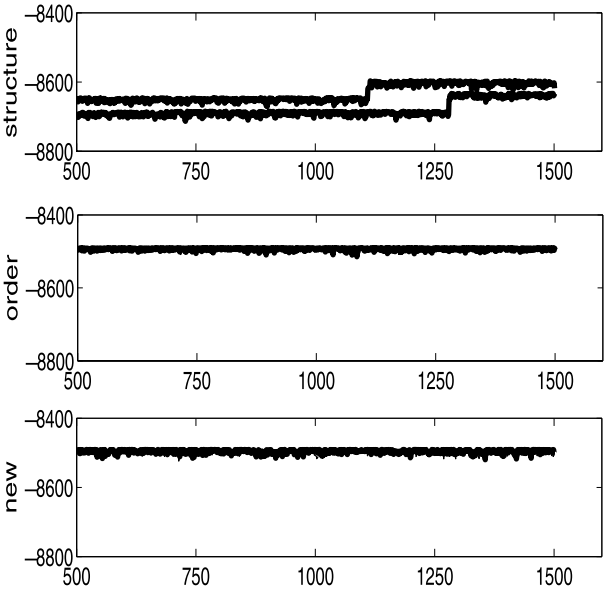


Fig. 12 Trace plots of the logarithmic scores of the DAGs sampled after the burn-in phase for Alarm data with $m = 750$. The *upper panel* shows the trace plots for two structure MCMC runs, the *middle panel* shows the trace plots for two order MCMC runs, and the *lower panel* shows the trace plots for two REV-structure MCMC runs

$m = 1000$ (see Fig. 13). The trace plot diagnostics confirm our findings derived from scatter plots of the directed edge feature posterior probabilities in Sect. 5. While order MCMC (middle panel) and REV-structure MCMC (lower panel) always reach the same plateau, structure MCMC reaches different lower plateaus, that is, it probably gets stuck in local optima. These results reveal that upgrading the structure MCMC sampler by the new edge revereral (REV) move substantially improves convergence.

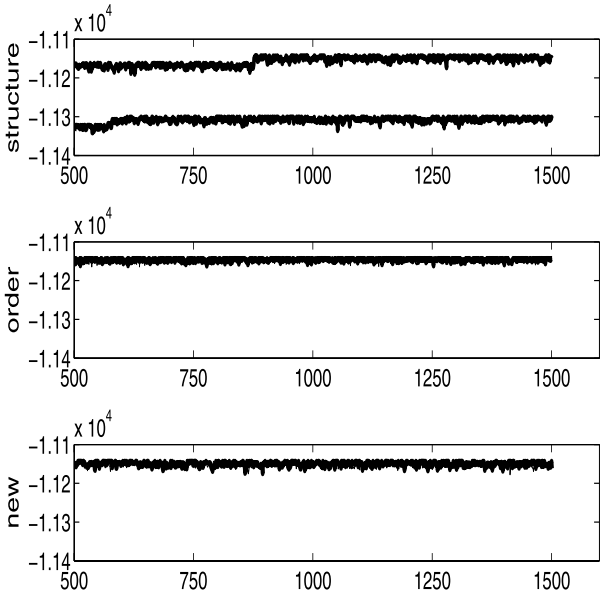


Fig. 13 Trace plots of the logarithmic scores of the DAGs sampled after the burn-in phase for Alarm data with $m = 1000$. The *upper panel* shows the trace plots for two structure MCMC runs, the *middle panel* shows the trace plots for two order MCMC runs, and the *lower panel* shows the trace plots for two REV-structure MCMC runs

Appendix 6: Comparison with inclusion-driven MCMC

This sixth appendix provides a comparison between our new REV-structure MCMC sampler and the method of inclusion-driven MCMC, as proposed by Castelo and Kočka (2003).

The methodology of inclusion-driven MCMC

We note that our sampling scheme seems to have some similarity with the method of inclusion-driven MCMC, proposed by Castelo and Kočka (2003). The approach of Castelo and Kočka (2003) is based on the concept of the inclusion boundary. The inclusion boundary of a DAG is the set of all the DAGs that can be reached from any DAG in the equivalence class of the current DAG by adding or removing an edge (ENR: Equivalence class No Reversals). Alternatively, the authors also consider an extended neighborhood that, in addition to the DAGs in ENR, contains the DAGs reached from any DAG in the equivalence class of the current DAG by reverting a non-covered edge (ENCR: Equivalence class Non-Covered Reversals), where a non-covered edge is an edge that, on reversal, leads to a DAG in a different equivalence class (see below for another definition). Inclusion-driven MCMC, in its strict sense, is a modified structure MCMC simulation where at each step a new DAG from the ENR or ENCR neighborhood of the current DAG is proposed. In practice, the inclusion boundary is not computationally efficient to handle. Castelo and Kočka (2003) therefore propose an alternative approach that is based on the reversal of covered edges. A directed edge from node A to node B is said to be covered if on removing this edge the parents of nodes B and A are identical. It can be shown (references in Castelo and Kočka 2003) that a reversal of a covered edge leads to a new DAG in the same equivalence class as the old DAG. Consequently, the reversal of covered edges provides an efficient way to traverse an equivalence class. The algorithm of Castelo and Kočka (2003) precedes a standard edge addition or removal operation by a randomly chosen number of covered edge reversals. The authors refer to this method as the RCAR (Repeated Covered Arc Reversal) algorithm. Since for a sufficiently large number of covered edge removals any DAG in the current equivalence class can

be reached with non-zero probability, the algorithm proposed by Castelo and Kočka (2003) effectively proposes a new DAG from the inclusion boundary without ever having to explicitly compute the inclusion boundary itself. A proposal move in inclusion-driven MCMC thus consists in applying the RCAR algorithm and then adding or removing an edge (RCARNR, in approximation of ENR). This can be extended to additionally reverting a non-covered edge (RCARR, in approximation of ENCR).

A theoretical comparison between inclusion-driven MCMC and REV-structure MCMC

As described above, Castelo and Kočka (2003) approximate the proposal from the inclusion boundaries ENR and ENCR with a series of covered edge reversals, using the RCAR algorithm. However, the very fact that the inclusion boundary is not computed explicitly causes problems when extending the scheme from an optimization task, for which it was originally devised, to MCMC. This is because the Hastings ratio required for computing the acceptance probability in the Metropolis-Hastings acceptance step is given by the ratio of the neighborhood sizes of the current and the proposed DAG, and this requires the computation of the inclusion boundary. Castelo and Kočka (2003) propose an approximate scheme in which they set the Hastings ratio to 1, ignoring the difference in the neighborhood sizes. Unfortunately, this introduces a small, but systematic bias, which our method avoids altogether by computing the Hastings factor properly.

The second difference concerns the improvement one expects to achieve over conventional structure MCMC. In conventional structure MCMC a move within the equivalence class has the same computational costs as any other move: it requires an acyclicity check to be carried out and the score of the proposed DAG to be computed. Inclusion driven MCMC reduces the computational costs of these types of move drastically by identifying and reverting covered edges. This renders both the computation of the network score and the acyclicity check obsolete, as we know from the theory presented in Castelo and Kočka (2003) that the reversal of a covered edge leads to a DAG in the same equivalence class. The essence of inclusion driven MCMC, thus, is an accelerated traversal of the equivalence classes. While this certainly speeds up the MCMC simulation to some extent, we would not expect it to solve convergence and mixing problems of structure MCMC in principle. This is because mixing and convergence problems are caused by high-probability regions in configuration space being separated by low-probability regions that are difficult to traverse. DAGs of the same equivalence class lie along a ridge with the same probability score. While inclusion driven MCMC provides a mechanism for a fast movement along these ridges, it does not provide a mechanism for bridging low-probability valleys. Hence, if the latter cause a structure MCMC simulation to stall, we would not expect inclusion driven MCMC to be able to fix it.

To shed further light onto this issue, we consider the configuration space of all 3-node DAGs, as shown in Fig. 14. This configuration space is sufficiently small to allow us to compute the Markov transition matrix explicitly. Recall that a Metropolis-Hastings step consists of two parts. First, a new DAG is proposed from some proposal distribution Q. Second, the proposed DAG is accepted with some acceptance probability A. For the 3-node configuration space, we can compute the matrices of proposal and acceptance probabilities $\mathbf{Q}$ and $\mathbf{A}$ explicitly. The latter matrix is computed from the data via the likelihood. The former matrix is directly computed from the definition of the MCMC scheme. Consider for example the move out of DAG 8 in Fig. 14. With standard structure MCMC, we can move into all valid DAGs that can be reached via the addition, deletion or reversal of an edge. These are the DAGs with the following ID numbers: 2, 4, 9, 11, 12, 13. Hence, the 8th column of the proposal probability matrix $\mathbf{Q}$ contains the entry $1/6$ in rows 2, 4, 9, 11, 12, and 13, and a zero

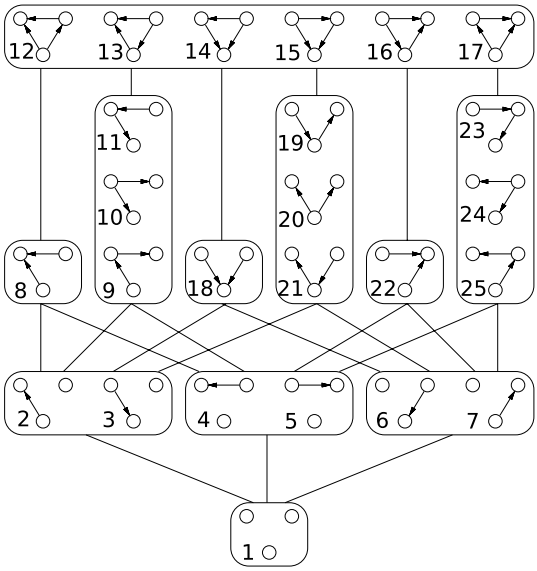
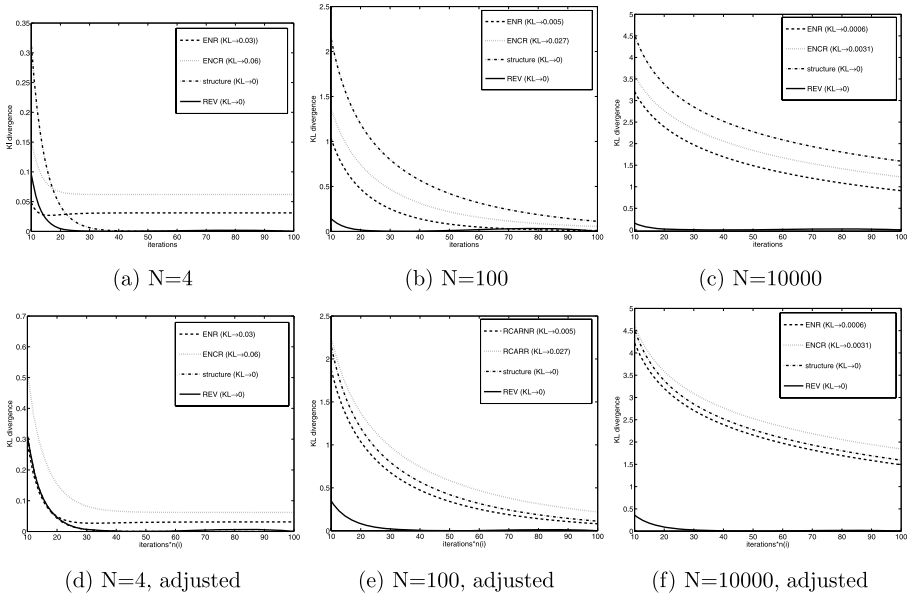


Fig. 14 Configuration space of 3-node DAGs. The figure shows the configuration space of 3-node DAGs, grouping DAGs in the same equivalence class together. Two equivalence classes are connected by an edge if one can transit between them by the application of an ENR move. Note that a direct transition between the three v-structured DAGs is not possible with such a move, in contrast to the REV MCMC moves proposed in Sect. 3. Adapted from Fig. 2 in Castelo and Kočka (2003).

everywhere else. The inclusion driven MCMC method ENR can propose any DAG with one edge added to or deleted from the DAGs in the current equivalence class. Since the current equivalence class does not contain any other DAGs besides DAG 8, we get a proposal probability matrix $\mathbf{Q}$ whose 8th column contains the entry 1/4 in rows 2, 4, 12, and 13, and a zero everywhere else. Note that neither of these two proposal probability matrices allows a direct transition into DAG 18. This can be effected with our new REV-structure MCMC scheme, though, for which the proposal probability matrix element $Q_{18,8}$ will be non-zero. We computed the proposal probability matrix $\mathbf{Q}$ for structure MCMC and the two inclusion-driven MCMC methods ENR and ENCR analytically along the line described above. For structure MCMC, the acceptance probability matrix $\mathbf{A}$ is computed from the ratio of the marginal likelihoods and the Hastings factor according to the standard Metropolis-Hastings acceptance criterion (see (5) and (6) in Sect. 2). For ENR and ENCR, the Hastings factor is set to 1. The product of the two matrices gives the Markov transition probability matrix $\mathbf{M} = \mathbf{AQ}$. The computation of $\mathbf{M}$ for REV-structure MCMC is in principle identical, but complicated by the fact that the proposal probabilities and the Hastings factor are now functions of the data. We therefore proceeded in an alternative way: given a data set and a DAG, we proposed 10^6 DAGs according to the scheme described in Sect. 3.2 and computed their acceptance probabilities from (31). Repeating this procedure for all 25 possible DAGs shown in Fig. 14, we numerically computed the Markov transition matrix $\mathbf{M}$.

Given the Markov transition matrix $\mathbf{M}$, we can analytically investigate the convergence properties of the different MCMC algorithms. Define $\mathbf{p}^t$ to denote a vector containing the probabilities of all possible DAGs in the tth step of the Markov chain, which is defined by $\mathbf{p}^{t+1} = \mathbf{M}\mathbf{p}^t$ starting from some initialization $\mathbf{p}^0$. For a given data set we compute the true posterior distribution $\mathbf{p}^*$ by exhaustive enumeration of all possible DAGs, which is trivial for 3 nodes. For every step t of the Markov chain we compute the symmetrized Kullback-Leibler (KL) divergence

$$KL(\mathbf{p}^t, \mathbf{p}^*) = \frac{1}{2} \sum_i \left(p_i^t \log \frac{2p_i^t}{p_i^t + p_i^*} + p_i^* \log \frac{2p_i^*}{p_i^t + p_i^*} \right). \tag{44}$$



(a) N=4 (b) N=100 (c) N=10000

(d) N=4, adjusted (e) N=100, adjusted (f) N=10000, adjusted

Fig. 15 Convergence characteristics of the Markov chain for the XOR problem, analytically computed. The figure shows the analytically computed convergence of a probability distribution on the space of 3-node DAGs of Fig. 14 towards the stable eigenvector of the Markov transition matrix. The *vertical axis* shows the symmetrized Kullback-Leibler divergence between the current and the true probability distribution, computed from (44), where the true distribution was obtained via exhaustive enumeration of all possible DAGs. Data were generated from an XOR with 4, 100, and 10000 data points. Four methods were compared: (1) standard structure MCMC (*dash-dotted line*), the two inclusion-driven MCMC methods proposed in Castelo and Kočka (2003) using the neighborhoods ENR (equivalence class no reversals, *dashed line*) and ENCR (equivalence class non-covered reversals, *dotted line*), and the REV-structure MCMC sampler (*solid line*). Each step in the iteration corresponds to a left-multiplication of the probability vector with the Markov transition matrix. To allow for the different computational costs of the corresponding Metropolis-Hastings steps, we have also considered an adjustment factor. Using the experimental findings of Castelo and Kočka (2003) conservatively and the estimates from Sect. 5.1 (see (35)), we obtain the following adjustment factors $n(\text{STRUCTURE}) = 1$, $n(\text{RCARR}) = n(\text{RCARNR}) = 2$, and $n(\text{REV}) = 1.6$. These results are shown in the *bottom row*

We repeated the procedure for three XOR data sets of different sample size: $m = 4, 100$ and 10000. The XOR data for three binary nodes X_1, X_2, and X_3 each having two possible realizations 0 and 1 were generated such that for the triple (X_1, X_2, X_3) the four realizations $(1, 0, 1)$, $(0, 1, 1)$, $(1, 1, 0)$, and $(0, 0, 0)$ had empirical probability 0.25 while the four other realizations had probability 0. The initialization vector $\mathbf{p}^0$ was chosen to be concentrated with probability 1 on the v-structure network with ID number 8 (see Fig. 14).

The results are shown in Fig. 15. For $m = 4$, there is no noticeable difference in the rate of convergence. This is because for such a small data set, the likelihood surface is relatively flat, and the movement in configuration space is effectively unrestricted. However, the inclusion-driven MCMC methods ENR and ENCR have a systematic bias and do not converge to the correct distribution; this is a consequence of the fact that the Hastings factor is not properly computed. As the data set size increases, the likelihood surface becomes more rugged, with three increasingly sharp peaks appearing at the DAGs with ID numbers 8, 18 and 22. Structure MCMC has mixing problems as a consequence of the need to cross a low-probability region to move between the peaks. According to the discussion above, inclusion-driven MCMC does not address this problem either: moving faster along

the likelihood ridges associated with the equivalence classes does not help us to bridge the low-probability regions. However, the proposed REV-structure MCMC scheme introduces direct proposal moves between the three peaks of the likelihood landscape, thereby achieving the very bridging effect that is required to improve mixing. We would therefore expect the convergence of $\mathbf{p}^t$ to show only a minor improvement when using ENR or ENCR instead of structure MCMC, but a substantial improvement with our new REV-structure MCMC scheme. This behavior is, in fact, observed in Fig. 15, which corroborates our conjecture.

We finally note that the plots of the KL-divergence against the iteration number of the Markov chain do not allow for the fact that in an MCMC simulation, the proposal steps come at different computational costs. We therefore adjusted the step number by a correction factor. Taking the experimental findings from Castelo and Kočka (2003) for ENR and ENCR, and our own estimates from Sect. 5.1 for REV-structure MCMC, we obtained that the computational costs of 10 structure MCMC steps correspond approximately to 6.25 REV-structure MCMC steps and 5 inclusion driven MCMC steps. This gives the following correction factors: $n(\text{STRUCTURE}) = 1$, $n(\text{RCARR}) = n(\text{RCARNR}) = 2$, and $n(\text{REV}) = 1.6$. The results are shown in the bottom panel of Fig. 15. It is seen that with this correction factor, the difference between structure and inclusion-driven MCMC becomes less pronounced. This corroborates our conjecture that inclusion-driven MCMC is effectively just an accelerated version of structure MCMC, whereas the proposed REV-structure MCMC scheme introduces new moves of a very different nature that potentially improve mixing and convergence.

An empirical comparison between inclusion-driven MCMC and REV-structure MCMC

For an empirical comparison, we have applied both inclusion-driven MCMC algorithms from Castelo and Kočka (2003) RCARNR10 (in approximation of ENR) and RCARR10 (in approximation of ENCR) to the seven Alarm data sets of different size, on which we had previously compared structure MCMC, order MCMC and REV-structure MCMC. According to the experimental findings in Castelo and Kočka (2003), RCARR10 and RCARNR10 steps are between 2 and 3 times as expensive as classical structure MCMC steps. Therefore, it seems to be fair with regard to the computational costs when the burn-in and sampling phase lengths of RCARNR10 and RCARR10 are obtained by dividing the corresponding lengths of structure MCMC by 2. This gives the following sampling scheme for RCARR10 and RCARNR10: Burn-in length 250,000 and a sampling phase of length 500,000 whereby each 500th DAG is sampled. Performing two independent MCMC runs on each of these seven Alarm data sets, whereby the first run was empty DAG seeded and the second run was greedy search seeded (see Sect. 5.1 for further details), we obtain for each data set two sets of directed edge (relation) feature estimates which can be plotted against each other. These scatter plots along with the corresponding scatter plots obtained with our REV-structure MCMC sampler (see the centre panels in Figs. 5 and 6) can be found in Figs. 16 and 17.

The results of Sect. 5.5 can be briefly summarized as follows. For small Alarm data sets, when the likelihood landscape is smooth, all three MCMC methods show similar convergence characteristics. However, as the data sets get larger and the likelihood surface becomes more rugged, structure MCMC increasingly fails the convergence tests; this is consistent with the results of Friedman and Koller (2003). Interestingly it can be seen from the right and left panels of Figs. 16 and 17 that inclusion-driven MCMC—as opposed to REV-structure MCMC—does not overcome these mixing and convergence problems: the persistent presence of off-diagonal entries in almost all scatter plots indicates poor convergence of the two inclusion-driven MCMC algorithms. A comparison of the scatter plots obtained with

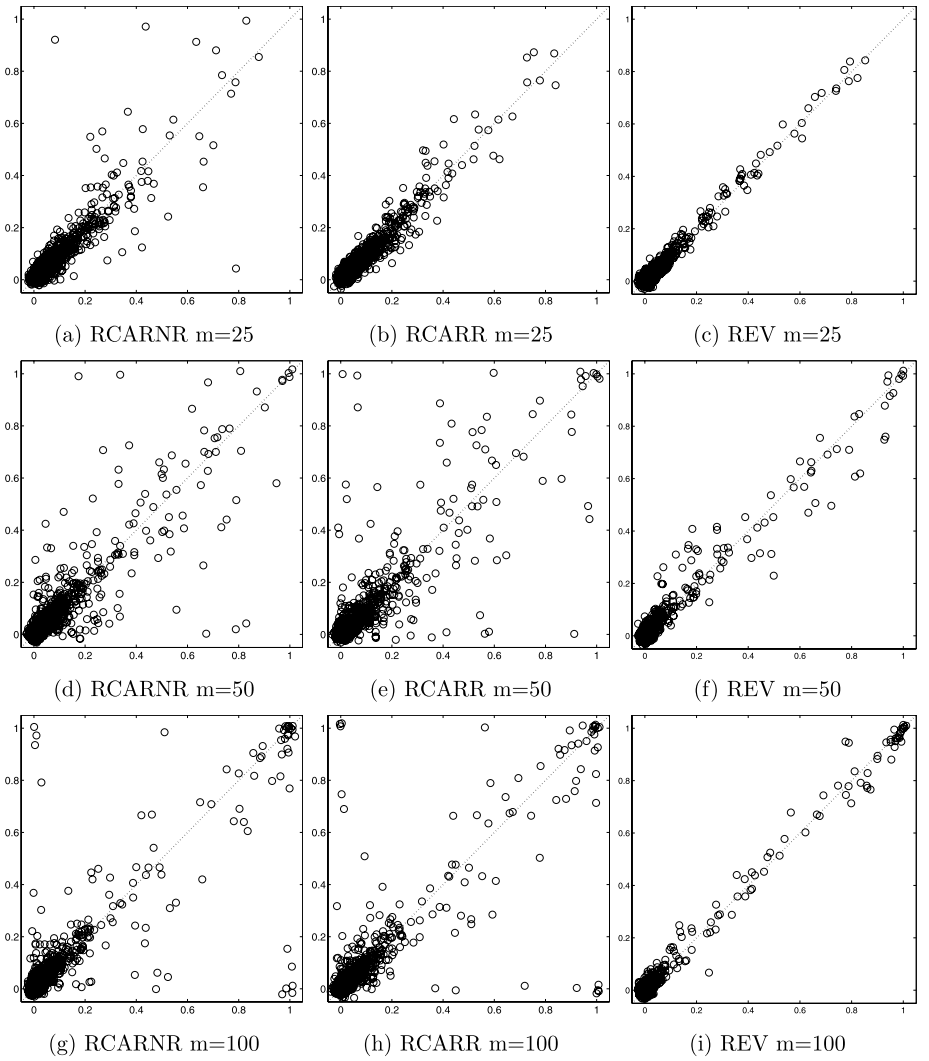
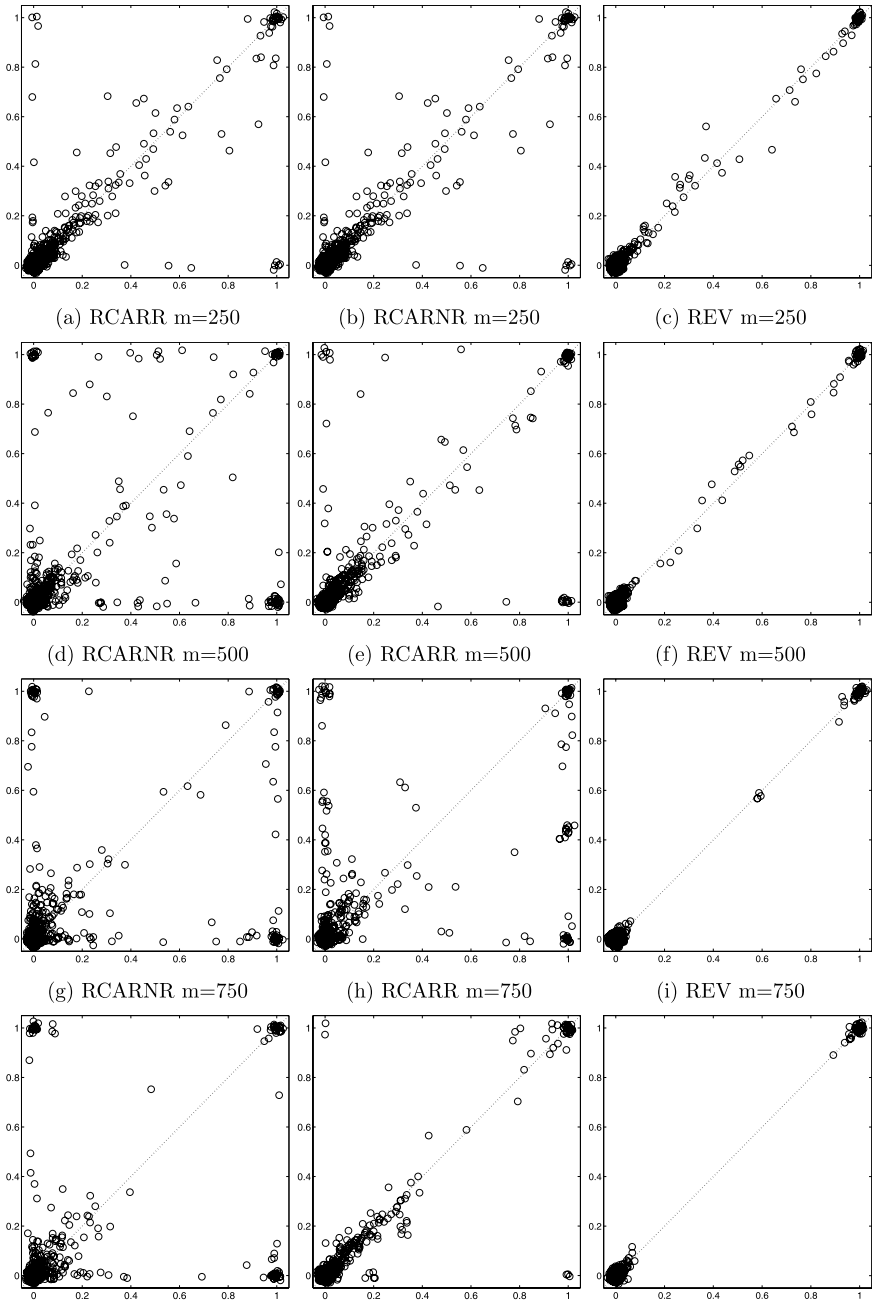


Fig. 16 Convergence test via scatter plots. The scatter plots compare the posterior probability estimates of the directed edge (relation) features on Alarm network data of different size m. *Left column*: RCARNR10; *centre column*: RCARR10; *right column*: REV-structure MCMC. Each *circle* corresponds to a directed edge feature. Its coordinates were estimated from two DAG samples of size 1000 obtained from two independent MCMC runs. The *abscissa* represents the results of an MCMC simulations with an empty-DAG initialization, while the *ordinate* was obtained from an MCMC simulation with a greedy initialization. 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

inclusion-driven MCMC and structure MCMC (see the right panels in Figs. 5 and 6) does *not* support the conjecture that inclusion-driven MCMC performs any better than structure MCMC. We assume that this is due to the fact that in our comparison the computational costs were taken into account such that the inclusion-driven MCMC estimates were obtained from MCMC runs of half the lengths of their structure MCMC counterparts. Castelo and Kočka



(a) RCARR m=250 (b) RCARNR m=250 (c) REV m=250

(d) RCARNR m=500 (e) RCARR m=500 (f) REV m=500

(g) RCARNR m=750 (h) RCARR m=750 (i) REV m=750

(j) RCARNR m=1000 (k) RCARR m=1000 (l) REV m=1000

Fig. 17 Convergence control via scatter plots continued. See caption of Fig. 16 for explanations

(2003) found that inclusion driven MCMC steps are between two and three times slower than classical structure MCMC steps, but did not adjust the run lengths in their experiments correspondingly.

The results of this additional study thus suggest that the proposed REV-structure MCMC scheme has the potential to overcome mixing and convergence problems of conventional structure MCMC much more effectively than inclusion-driven MCMC. We also note that our results are consistent with the authors' own findings, where convergence and mixing problems of inclusion-driven MCMC on the ALARM data become apparent in Fig. 14b of Castelo and Kočka (2003).

References

Beinlich, I., Suermondt, R., Chavez, R., & Cooper, G. (1989). The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In J. Hunter (Ed.), *Proceedings of the second European conference on artificial intelligence and medicine*. Berlin: Springer.

Castelo, R., & Kočka, T. (2003). On inclusion-driven learning of Bayesian networks. *Journal of Machine Learning Research*, *4*, 527–574.

Chickering, D. M. (1995). A transformational characterization of equivalent Bayesian network structures. In *International conference on uncertainty in artificial intelligence (UAI)* (Vol. 11, pp. 87–98).

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

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

Eaton, D., & Murphy, K. (2007). Bayesian structure learning using dynamic programming and MCMC. In *Proceedings of the twenty-third conference on uncertainty in artificial intelligence (UAI 2007)*.

Ellis, B., & Wong, W. (2006). Sampling Bayesian networks quickly. In *Interface*, Pasadena, CA.

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

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

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

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

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*, 97–109.

Husmeier, D. (2003). Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, *19*, 2271–2282.

Imoto, S., Higuchi, T., Goto, T., Kuhara, S., & Miyano, S. (2003). Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. In *Proceedings IEEE computer society bioinformatics conference (CSB'03)* (pp. 104–113).

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

Jensen, F. V. (1996). *An introduction to Bayesian networks*. London: UCL Press.

Kanehisa, M. (1997). A database for post-genome analysis. *Trends in Genetics*, *13*, 375–376.

Kanehisa, M., & 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., & Hirakawa, M. (2006). From genomics to chemical genomics new developments in kegg. *Nucleic Acids Research*, *34*, 354–357.

Kovisto, M. (2006). Advances in exact Bayesian structure discovery in Bayesian networks. In *Proceedings of the twenty-second conference on uncertainty in artificial intelligence (UAI 2006)*.

Kovisto, M., & Sood, K. (2004). Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, *5*, 549–573.

Larget, B., & Simon, D. L. (1999). Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees. *Molecular Biology and Evolution*, *16*(6), 750–759.

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

Mansinghka, V. K., Kemp, C., Tenenbaum, J. B., & Griffiths, T. L. (2006). Structured priors for structure learning. In *Proceedings of the twenty-second conference on uncertainty in artificial intelligence (UAI 2006)*.

Moore, A., & Wong, W. K. (2003). Optimal Reinsertion: a new search operator for accelerated and more accurate Bayesian network structure learning. In T. Fawcett & N. Mishra (Eds.), *Proceedings of the 20th international conference on machine learning (ICML '03)* (pp. 552–559). Menlo Park: AAAI Press.

Nariai, N., Tamada, Y., Imoto, S., & Miyano, S. (2005). Estimating gene regulatory networks and protein-protein interactions of saccharomyces cerevisiae from multiple genome-wide data. *Bioinformatics*, *21*(Suppl. 2), ii206–ii212.

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

Ott, S., Imoto, S., & Miyano, S. (2004). Finding optimal models for small gene networks. In *Pacific symposium on biocomputing* (Vol. 9, pp. 557–567).

Pearl, J. (2000). *Causality: models, reasoning and intelligent systems*. London: Cambridge University Press.

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

Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics*, *22*(4), 1701–1728.

Verma, T., & Pearl, J. (1990). Equivalence and synthesis of causal models. In *Proceedings of the sixth conference on uncertainty in artificial intelligence* (Vol. 6, pp. 220–227).

Werhli, A. V., & 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* (Article 15).