Mach Learn (2008) 71: 185–217
DOI 10.1007/s10994-008-5053-y

Learning the structure of dynamic Bayesian networks from time series and steady state measurements

Harri Lähdesmäki · Ilya Shmulevich

Received: 21 May 2007 / Revised: 27 February 2008 / Accepted: 28 March 2008 / Published online: 12 April 2008

Abstract Dynamic Bayesian networks (DBN) are a class of graphical models that has become a standard tool for modeling various stochastic time-varying phenomena. In many applications, the primary goal is to infer the network structure from measurement data. Several efficient learning methods have been introduced for the inference of DBNs from time series measurements. Sometimes, however, it is either impossible or impractical to collect time series data, in which case, a common practice is to model the non-time series observations using static Bayesian networks (BN). Such an approach is obviously sub-optimal if the goal is to gain insight into the underlying dynamical model. Here, we introduce Bayesian methods for the inference of DBNs from steady state measurements. We also consider learning the structure of DBNs from a combination of time series and steady state measurements. We introduce two different methods: one that is based on an approximation and another one that provides exact computation. Simulation results demonstrate that dynamic network structures can be learned to an extent from steady state measurements alone and that inference from a combination of steady state and time series data has the potential to improve learning performance relative to the inference from time series data alone.

Keywords Dynamic Bayesian networks · Steady state analysis · Bayesian inference · Markov chain Monte Carlo · Trans-dimensional Markov chain Monte Carlo

Communicated by Kevin P. Murphy.

H. Lähdesmäki (✉) · I. Shmulevich
Institute for Systems Biology, 1441 North 34th Street, Seattle, WA 98103, USA
e-mail: harri.lahdesmaki@tut.fi

I. Shmulevich
e-mail: ishmulevich@systemsbiology.org

H. Lähdesmäki
Department of Signal Processing, Tampere University of Technology, Tampere, Finland

1 Introduction

Dynamic Bayesian networks (DBNs), also called dynamic probabilistic networks, are a general and flexible model class that is capable of representing complex temporal stochastic processes (Dean and Kanazawa 1989; Murphy 2002). DBNs and their non-temporal versions, i.e., static Bayesian networks (BN), have successfully been used in different modeling problems, such as in speech recognition, target tracking and identification, genetics, probabilistic expert systems, and medical diagnostic systems (see, e.g., Cowell et al. 1999, and the references therein). Recently, BNs and DBNs have also been intensively studied in the context of modeling genomic regulation, see, e.g., (Hartemink et al. 2001, 2002; Husmeier 2003; Imoto et al. 2003; Friedman 2004; Pournara and Wernisch 2004; Sachs et al. 2005; Bernard and Hartemink 2005; Werhli et al. 2006; Lähdesmäki et al. 2006).

In many applications, the underlying network structure is unknown. Therefore, the first and often the most important problem is to infer the model structure from measurements. There exists an extensive body of literature introducing efficient BN and DBN learning methods. Different model inference methods can be divided into two categories: methods that attempt to construct networks by estimating conditional independencies between nodes in the network (Pearl 2000), and methods that search candidate models through the space of network models using a statistical score combined with different search/estimation algorithms, such as greedy search or powerful standard stochastic estimation methods (Cooper and Herskovits 1992; Madigan and York 1995; Heckerman 1998). Here we follow the latter approach.

Temporal models are best learned from temporal data. However, experimental settings do not always permit collecting time series measurements, and may only capture so-called steady state measurements.[1] That is frequently the case in bioinformatics and computational systems biology studies, where regulatory network models are inferred from gene expression or proteomics data. One has previously been left with two alternative approaches. First, all the samples, both time series and steady state, can be used to infer static BNs. This approach is inherently limited to learning non-dynamic network models and is therefore sub-optimal if the underlying model is dynamic in nature. Second, one can use only the time series measurements for the inference of dynamic models. While this approach is principled, it results in an inefficient procedure as it ignores part of the measurements, which can contain a substantial amount of information about the dynamic behavior of the network. In this work we describe a rigorous Bayesian method for learning the structure of DBNs from steady state measurements. This is achieved in two different ways: either learning network structures from steady state measurements alone, or learning network structures from a combination of time series and steady state measurements. We introduce both approximate and exact learning methods. Simulation results are presented to show that the proposed methods provide an improved learning methodology.

To the best of our knowledge, no method capable of learning dynamic network models from steady state measurements, at least for DBNs, has been proposed previously. Relevant background material for our approach includes studies introducing BN and DBN inference methods (Cooper and Herskovits 1992; Madigan and York 1995; Heckerman 1998; Friedman et al. 1998; Husmeier 2003). Similar ideas of using steady state analysis in dynamic model inference have been developed for hidden Markov models (HMM) in (Robert et al. 2000). The goal of Robert et al. was to develop a Bayesian estimation method for the

[1] Steady state measurements can be considered as snapshots of the long-run behavior of a system. A more precise definition of steady state is given later in Sect. 3.

number of components in an HMM from time series data, where the first observation was assumed to be generated from the steady state distribution of the HMM. A modification of the method was also proposed for i.i.d. data, essentially corresponding to steady state data, but in that case the Bayesian inference was implemented using the steady state distribution directly, not the underlying dynamic HMM. In particular, here we consider learning the structure of the underlying DBN, which generates the Markovian process, from steady state measurements. In addition to structure learning from complete and incomplete time series measurements, Friedman et al. (1998) also considered learning DBNs in a similar setting as we do here. They assumed that data was sampled from an unknown time step. They also augmented the dynamic network model with an additional switch variable that, once set, freezes the state of the network. The structural expectation-maximization (EM) algorithm was used to infer hidden states of the system back to the beginning of the time series (time 0) and to find a maximum a posteriori (MAP) network structure. Note that some related work has also been reported by Nikovski (1998), who proposes a method for learning parameters of DBNs from incomplete time series data by imposing a (slightly different) constraint on stationarity. This approach is similar in spirit to what we propose here. The main differences are that we build our methods on the standard Markovian assumption ($P(\mathbf{X}[t+1]|\mathbf{X}[t])$, inherent in the model) and on steady state analysis implied by this transition model. Most importantly, we consider structural learning from steady state measurements.

Section 2 reviews the modeling framework. Steady state analysis of DBNs is described in Sect. 3. The inference methods are introduced in Sect. 4. Simulations and results are discussed in Sects. 5 and 6, respectively. Conclusions and discussion are given in Sect. 7.

2 Modeling framework

This section introduces background of BNs and DBNs necessary for further analysis.

2.1 Bayesian networks

A Bayesian network is defined by a graphical model structure $\mathcal{M}$ and a family of conditional distributions $\mathcal{F}$ and their parameters θ. The model structure $\mathcal{M}$ consists of a set of nodes V and a set of directed edges E connecting the nodes such that the resulting directed graph is acyclic (DAG). The nodes represent random variables in the network whereas the edges encode a set of conditional dependencies. In the parametric setting, the family of conditional distributions $\mathcal{F}$ is assumed to be known and hence is fully described by its parameters.

Let $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ denote a set of random variables that correspond to the nodes V in the network. Lower-case letters $x_1, x_2, \ldots, x_n$ are used to denote the value of the corresponding variables. Let $\mathbf{Pa}(X_i)$ denote the random variables corresponding to the parents of node i in the DAG. Then, the network structure $\mathcal{M}$ and the parameters θ of the conditional distributions together define a joint distribution over the random variables $\mathbf{X}$ as

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathbf{pa}(X_i)).$$

In the following, we assume that each random variable X_i can take on r_i values. Furthermore, we only focus on the family of (unconstrained) multinomial conditional distributions, although other parametric families are also possible.

Note that static BNs can be limiting in some applications because the network structures are acyclic. Although information flow in BNs is bi-directional relative to the directed edges, static BNs do not allow one to explicitly model the direct or indirect feedback loops explicitly that are frequently encountered in applications.

2.2 Dynamic Bayesian networks

DBNs, which are temporal extensions of BNs, extend the above concepts to stochastic processes. Let $\mathbf{X}[t] = \{X_1[t], X_2[t], \ldots, X_n[t]\}$ denote the random variables in $\mathbf{X}$ at time $t \in \{1, 2, \ldots\}$. We restrict our attention to homogeneous first-order Markov processes in $\mathbf{X}$, i.e., $P(\mathbf{X}[t]|\mathbf{X}[t-1], \ldots, \mathbf{X}[1]) = P(\mathbf{X}[t]|\mathbf{X}[t-1])$ for all $t > 1$ and for all values of $\mathbf{X}[1], \mathbf{X}[2], \ldots, \mathbf{X}[t]$. We also assume that each node $X_i[t]$ has all of its parents among variables $\mathbf{X}[t-1]$. Under these assumptions, the joint probability distribution of a finite length time series can be written as

$$P(\mathbf{x}[1], \mathbf{x}[2], \ldots, \mathbf{x}[T]) = P(\mathbf{x}[1]) \prod_{t=2}^{T} P(\mathbf{x}[t]|\mathbf{x}[t-1]) \tag{1}$$

$$= P(\mathbf{x}[1]) \prod_{t=2}^{T} \prod_{i=1}^{n} P(x_i[t]|\mathbf{pa}(X_i[t])). \tag{2}$$

Note that the role of the first sample $\mathbf{x}[1]$ is slightly different, which is discussed more in the following. Although (1) and (2) generalize easily to more than one time series, we will limit our discussion to a single time series for notational simplicity.

Note that the above constraints guarantee the underlying "time unrolled" network still to be acyclic but at the same time allow modeling feedback loops explicitly by directed edges. Although only first order Markov processes are considered here, the following discussion naturally extends to higher order Markov processes as well since the state space can always be extended to accommodate higher-order processes. DBNs can also be defined to contain edges within a time slice, i.e., $\mathbf{Pa}(X_i[t]) \subseteq \{\mathbf{X}[t], \mathbf{X}[t-1]\}$ instead of $\mathbf{Pa}(X_i[t]) \subseteq \{\mathbf{X}[t-1]\}$. While directed edges between consecutive time slices $\mathbf{X}[t-1]$ and $\mathbf{X}[t]$ represent causal flow, within time slice connections can be interpreted as instantaneous causality. Although we assume DBNs to have only between slice edges, our learning methods work equally well if we allow edges within a time slice, as long as the time unrolled network is a DAG (so that the graphical model structure represents a DBN).

3 Steady state analysis of DBNs

Equation (2) characterizes stochastic behavior of DBNs over a finite time interval. However, it is also important to consider the long-run behavior of DBNs. Since we focus on homogeneous discrete-valued DBNs, we can study their dynamics using finite-state Markov chains.

Let A denote the state transition matrix of a Markov chain corresponding to a DBN. Using the state vectors of a DBN to index A, let $A_{\mathbf{uv}}$ denote the probability that a DBN will move to state $\mathbf{v}$ given that the current state is $\mathbf{u}$, i.e.,

$$A_{\mathbf{uv}} = P(\mathbf{X}[t] = \mathbf{v}|\mathbf{X}[t-1] = \mathbf{u})$$

$$= \prod_{i=1}^{n} P(X_i[t] = v_i|\mathbf{Pa}(X_i[t]) = \mathbf{u}_{\mathbf{pa}_i}), \tag{3}$$

where v_i is the ith element of $\mathbf{v}$ and $\mathbf{u}_{\mathbf{pa}_i}$ denotes the elements of $\mathbf{u}$ that correspond to the parents of the ith node. In the case of multinomial distributions, (3) can be rewritten in terms of its parameters as

$$A_{\mathbf{uv}} = \prod_{i=1}^{n} \theta_{i,\mathbf{u}_{\mathbf{pa}_i},v_i}, \tag{4}$$

where $\theta_{i,\mathbf{u}_{\mathbf{pa}_i},v_i} = P(X_i[t] = v_i | \mathbf{Pa}(X_i[t]) = \mathbf{u}_{\mathbf{pa}_i})$.

Let $A^{(r)}_{\mathbf{uv}} = P(\mathbf{X}[t+r] = \mathbf{v} | \mathbf{X}[t] = \mathbf{u})$ denote the r-step transition probability of the homogeneous Markov chain. A state $\mathbf{v}$ is said to be accessible from state $\mathbf{u}$ if there exists an $r > 0$ such that $A^{(r)}_{\mathbf{uv}} > 0$. Two states $\mathbf{u}$ and $\mathbf{v}$ are said to communicate if $\mathbf{v}$ is accessible from $\mathbf{u}$ and if $\mathbf{u}$ is accessible from $\mathbf{v}$. The communication relation divides the states into equivalence classes: states inside an equivalence class communicate which each other but not with states outside the class. A Markov chain is said to be irreducible if the number of equivalence classes generated by the communication relation is equal to one, i.e., all states of the chain communicate. The period of a state $\mathbf{u}$, $d_{\mathbf{u}}$, is the greatest common factor of integers $\{r \mid A^{(r)}_{\mathbf{uu}} > 0\}$. A Markov chain is said to be aperiodic if $d_{\mathbf{u}} = 1$ for all the states $\mathbf{u}$.

The Markov chain is said to possess a steady state distribution if there exists a probability distribution π such that

$$\lim_{r\to\infty} A^{(r)}_{\mathbf{uv}} = \pi_{\mathbf{v}}$$

for all states $\mathbf{u}$ and $\mathbf{v}$. The fundamental theorem of Markov chains says that every finite-state homogeneous Markov chain that is irreducible and aperiodic (i.e., ergodic) possesses a unique steady state distribution (see, e.g., Çinlar 1997). Moreover, for any initial distribution $\pi^{(0)}$ of $\mathbf{u}$, the state probability after r steps $\pi^{(r)}_{\mathbf{v}}$ approaches $\pi_{\mathbf{v}}$ when $r \to \infty$. Let us next establish a useful result for DBNs.

Theorem 1 *A sufficient condition for the Markov chain corresponding to a DBN to possess a unique steady state distribution, independent of the initial distribution, is that* $\theta_{i,\mathbf{u}_{\mathbf{pa}_i},v_i} > 0$ *for all possible values of* i, $\mathbf{u}_{\mathbf{pa}_i}$, *and* v_i.

Proof Consider any two state vectors $\mathbf{u}$ and $\mathbf{v}$. It is clear that $A^{(1)}_{\mathbf{uv}} = P(\mathbf{X}[t] = \mathbf{v} | \mathbf{X}[t-1] = \mathbf{u}) = \prod_{i=1}^{n} \theta_{i,\mathbf{u}_{\mathbf{pa}_i},v_i} > 0$. Hence all states communicate and the DBN is irreducible. Similarly, $A^{(1)}_{\mathbf{uu}} > 0$ and therefore $d_{\mathbf{u}} = 1$ for all the states $\mathbf{u}$, which ensures aperiodicity. □

Note that Theorem 1 gives a sufficient (not necessary) condition for ergodic dynamics. It is easy to construct DBNs where some $\theta_{i,\mathbf{u}_{\mathbf{pa}_i},v_i} = 0$ but the corresponding Markov chain is still both irreducible and aperiodic. Existence of ergodic dynamics for semi-deterministic models, however, needs to be checked on a case-by-case basis. Also note that any semi-deterministic model can be approximated as accurately as desired by requiring $\theta_{i,\mathbf{u}_{\mathbf{pa}_i},v_i} > 0$ but letting $\theta_{i,\mathbf{u}_{\mathbf{pa}_i},v_i} \to 0$.

In the following, we assume that the underlying model possesses a unique steady state distribution π from which the steady state measurements are also sampled. In most applications, this is a reasonable assumption. For example, the assumption of having a unique steady state distribution is made implicitly in practically every biological application where, e.g., static transcriptional regulatory network models, such as BNs, are learned from steady state gene expression or protein level data (Pe'er et al. 2001; Hartemink et al. 2002; Imoto et al. 2003; Dobra et al. 2004; Pournara and Wernisch 2004; Wille et al. 2004; Sachs et al. 2005; Schäfer and Strimmer 2005; Werhli et al. 2006). For a more complete list

of previous work, see (Markowetz 2007). Since the underlying (biological) system is known to be dynamic, the data generating mechanism is then naturally modeled as the steady state distribution of the dynamical system. Moreover, since steady state measurements are typically sampled infrequently, they are best described as being isolated and non-successive samples from π. It is also worth noting that our the methods proposed here are valid even if the underlying system is not strictly ergodic. In particular, if a DBN is not irreducible but each closed communicating sub-class has its own unique stationary distribution, then the same methods (described below) can be applied sub-class wise.

It is important to note that the mapping from DBNs to steady state distributions is not a bijection. To show a counterexample, consider, for example, two three-node networks with particular network structures (E_{12}, E_{23}, E_{31}) and (E_{21}, E_{32}, E_{13}), respectively, where E_{ij} denotes a directed edge from $X_i[t-1]$ to $X_j[t]$. If all the nodes in both networks are associated with the same conditional distribution, then the corresponding steady state distributions are equivalent. Although this example is somewhat artificial, from the point of view of network inference from steady state measurements this means that the inference problem can have more than one optimal point estimate, i.e., it is ill-posed. This issue is automatically taken care of by introducing (fully) Bayesian inference methods that simply assign the same posterior probability to such score-equivalent networks, assuming equal prior probabilities. This is exactly analogous to learning static BNs from non-interventional measurements where only equivalence classes of BNs can be learned. The score-equivalence problem disappears when DBNs are learned from a combination of time series and steady state data.

In Sect. 4, we are also interested in solving for the steady state distribution π. Note that the row vector π can be obtained by solving $\pi A = \pi$, i.e., the left eigenvector corresponding to the eigenvalue $\lambda = 1$. Also note that instead of solving the whole eigenproblem associated with the stochastic matrix A, one only needs to solve for the eigenvector corresponding to the largest eigenvalue. For that purpose, one can use a variety of methods (see, e.g., Stewart 1994). We use an algorithmic variant of the Arnoldi iteration called the implicitly restarted Arnoldi method as implemented in ARPACK/Matlab (Lehoucq et al. 1998). Although efficient algorithms have been introduced for solving the dominant eigenvector, the problem becomes computationally demanding for large state transition matrices. That also causes the main computational bottleneck of the current implementation of the proposed inference methods. For further discussion on this issue and recent improvements, see Sect. 7.

4 Bayesian learning methods

In the Bayesian context the most natural and most often used scoring metric is the posterior probability of a network $\mathcal{M}$ given data $\mathcal{D}$, $P(\mathcal{M}|\mathcal{D})$. According to Bayes' rule, the posterior probability can be written as

$$P(\mathcal{M}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{M})P(\mathcal{M})}{P(\mathcal{D})},$$

where $P(\mathcal{D})$ is a constant that does not depend on $\mathcal{M}$. Consequently, both the marginal likelihood $P(\mathcal{D}|\mathcal{M})$ and the network prior $P(\mathcal{M})$ play a central role in the inference. In a full Bayesian analysis the marginal likelihood involves marginalization over the whole parameter space

$$P(\mathcal{D}|\mathcal{M}) = \int_\theta P(\mathcal{D}|\mathcal{M},\theta)P(\theta|\mathcal{M})d\theta. \tag{5}$$

Although the network prior is an important factor, especially in small sample settings, for the purposes of this study, we assume an uninformative (uniform) prior over network models. If prior knowledge of a particular problem domain exists, the priors can be used in the proposed method the same way they are used in the traditional Bayesian inference.

Traditionally, DBNs have been learned from time series measurements only. In several applications, especially those in computational biology, the collected data typically contain steady state measurements. Our goal here is to use such steady state measurements, either alone or together with time series measurements, to learn the structure of DBNs. In the following, measured data are denoted collectively as $\mathcal{D}$. To distinguish between different types of measurements we write $\mathcal{D} = (\mathcal{D}_A, \mathcal{D}_\pi)$, where $\mathcal{D}_A = (\mathbf{x}[1], \mathbf{x}[2], \ldots, \mathbf{x}[T])$ and $D_\pi = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M)$ denote time series and steady state measurements, respectively. In general, time series data can contain several, say R, instances each having length T_i, i.e., $\mathcal{D}_A = (\mathcal{D}_A^1, \mathcal{D}_A^2, \ldots, \mathcal{D}_A^R)$, where $\mathcal{D}_A^i = (\mathbf{x}^i[1], \mathbf{x}^i[2], \ldots, \mathbf{x}^i[T_i])$. We assume a single time series for notational convenience. In the following, we also assume fully observed data. We first start with a brief discussion on parameter priors, parameter learning, and traditional time series based model inference.

4.1 Parameter priors

Recall that each random variable X_i is assumed to have r_i possible values. Let $\{i_1, i_2, \ldots, i_{|\mathbf{Pa}(X_i)|}\}$ denote the indices of the parents of node i. The number of possible parent configurations for node i is $q_i = r_{i_1} r_{i_2} \cdots r_{i_{|\mathbf{Pa}(X_i)|}}$. For notational convenience, let us rewrite the parameters $\theta_{i,\mathbf{u}_{\mathbf{pa}_i},v_i}$ as θ_{ijk}, where $i \in \{1, 2, \ldots, n\}$, $j \in \{1, 2, \ldots, q_i\}$, and $k \in \{1, 2, \ldots, r_i\}$.[2]

Given a network structure $\mathcal{M}$, one needs to define a prior probability model for the corresponding parameters. That may be a difficult task given the large number of different network structures, $2^{(n^2)}$ in the case of DBNs of the form we consider here. To simplify things, we follow the common practice and assume the parameter priors to fulfill both so-called global and local independence. The global parameter independence is defined as $P(\theta|\mathcal{M}) = \prod_{i=1}^n P(\theta_i|\mathbf{Pa}(X_i))$, where $\theta_i = \{\theta_{ijk} \mid j \in \{1, 2, \ldots, q_i\}, k \in \{1, 2, \ldots, r_i\}\}$, whereas the local independence means $P(\theta_i|\mathbf{Pa}(X_i)) = \prod_{j=1}^{q_i} P(\theta_{ij}|\mathbf{Pa}(X_i))$, where $\theta_{ij} = \{\theta_{ijk} \mid k \in \{1, 2, \ldots, r_i\}\}$.

It can be shown that the local and global parameter independence together with so-called likelihood equivalence for static BNs imply the prior to be Dirichlet (Geiger and Heckerman 1997). Furthermore, and more importantly for DBNs, Dirichlet is the conjugate prior for multinomials. Given the above assumptions, the Dirichlet prior for each θ_{ij} with hyperparameters α is defined as

$$P(\theta_{ij}|\alpha) = \frac{\Gamma(\alpha_{ij})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1},$$

where $\theta_{ijk} \geq 0$, $\sum_{k=1}^{r_i} \theta_{ijk} = 1$, $\alpha_{ijk} \geq 0$, $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$, and $\Gamma(\cdot)$ is the Gamma function.

4.2 Inference from time series measurements

Given a network structure $\mathcal{M}$, let N_{ijk} denote the number of times variable configuration $(X_i[t] = k, \mathbf{pa}(X_i[t]) = j)$ occurs in time series data $\mathcal{D}_A$. Since the Dirichlet distribution is the conjugate prior for multinomials, the posterior distribution of θ_{ij} given the data

[2] Any bijective mapping can be used for $\mathbf{u}_{\mathbf{pa}_i}$ and j and for v_i and k.

$\mathcal{D}_A$, $P(\theta_{ij}|\alpha, \mathcal{D}_A)$, also has a Dirichlet distribution, but with parameters $\alpha_{ij1} + N_{ij1}, \alpha_{ij2} + N_{ij2}, \ldots, \alpha_{ijr_i} + N_{ijr_i}$. This explicitly shows that the hyperparameters can be interpreted as pseudo counts of cases $(X_i[t] = k, \mathbf{pa}(X_i[t]) = j)$.

Different (posterior) estimates of parameters θ_{ijk} can be defined. Three of them are considered here: maximum likelihood (ML), maximum a posteriori (MAP) and posterior mean, which can be written as (see, e.g., Murphy 2002)

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{N_{ij}}$$

for the ML,

$$\tilde{\theta}_{ijk} = \frac{\alpha_{ijk} + N_{ijk} - 1}{\alpha_{ij} + N_{ij} - r_i} \tag{6}$$

for the MAP and

$$\overline{\theta}_{ijk} = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}} \tag{7}$$

for the posterior mean, where $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

As we are also interested in the long-run behavior of DBNs, it is natural to study the conditions under which the learned model (network structure is fixed, only parameters are learned from time series data) possesses a unique steady state distribution. For small samples, ergodicity can be guaranteed by the hyperparameters of the prior distribution. A sufficient condition is formulated in the following theorem.

Theorem 2 *Given a network structure $\mathcal{M}$ and time series data set $\mathcal{D}_A$, a sufficient condition for the finite-state Markov chain corresponding to the network model $(\mathcal{M}, \tilde{\theta})$ (resp. $(\mathcal{M}, \overline{\theta})$) to possess a unique steady state distribution is that $\alpha_{ijk} > 1$ (resp. $\alpha_{ijk} > 0$) for all i, j and k.*

Proof The result follows directly from Theorem 1 and (6) and (7). □

Note that the above result does not hold for the ML estimates.

Under the above assumptions on the parameter priors and complete data, computation of the marginal likelihood is analytically tractable, and $P(\mathcal{D}_A|\mathcal{M})$ can be written as (Cooper and Herskovits 1992; Heckerman et al. 1995)

$$P(\mathcal{D}_A|\mathcal{M}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}. \tag{8}$$

4.3 Inference from time series and steady state measurements

As discussed above, steady state measurements are modeled as isolated and non-successive samples from π. Thus, time series and steady state measurements are independent conditional on a DBN $(\mathcal{M}, \theta)$. Furthermore, steady state measurements are conditionally independent as well. This latter independence assumption states that steady state measurements are sampled infrequently enough so that the correlation (over time) between any $\mathbf{x}_i$ and $\mathbf{x}_j$ is negligible. Although we do not consider any correlation structure between steady state measurements, such correlations could be easily taken into account via the r-step transition

probabilities $A^{(r)}_{\mathbf{uv}}$ (assuming r is known). Given both time series and steady state measurements, the marginal likelihood can be written as

$$P(\mathcal{D}_A, \mathcal{D}_\pi|\mathcal{M}) = \int_\theta P(\mathcal{D}_A, \mathcal{D}_\pi|\mathcal{M}, \theta) P(\theta|\mathcal{M}) d\theta$$
$$= \int_\theta P(\mathcal{D}_A|\mathcal{M}, \theta) P(\mathcal{D}_\pi|\mathcal{M}, \theta) P(\theta|\mathcal{M}) d\theta, \tag{9}$$

where we have used the conditional independence between $\mathcal{D}_A$ and $\mathcal{D}_\pi$, given $(\mathcal{M}, \theta)$. Unfortunately, the above integral is no longer analytically tractable in general. To elaborate, let us use the independence between $\mathbf{x}_i$ and $\mathbf{x}_j$ $(i \neq j)$ and write

$$P(\mathcal{D}_\pi|\mathcal{M}, \theta) = \prod_{i=1}^{M} P(\mathbf{x}_i|\mathcal{M}, \theta) = \prod_{i=1}^{M} \pi_{\mathbf{x}_i}, \tag{10}$$

where π is the steady state distribution of a DBN defined by $\mathcal{M}$ and θ. Note especially that each $P(\mathbf{x}_i|\mathcal{M}, \theta)$ depends on $\mathbf{x}_i$ in a complicated manner via the steady state distribution. Finally, if the first time series sample $\mathbf{x}[1]$ can be considered to be sampled from the steady state distribution, then that can be accounted for by multiplying (10) by $\pi_{\mathbf{x}[1]}$.

Below we introduce both approximate and "exact" inference methods in Sects. 4.4 and 4.6, respectively. Approximate methods are limited in that they can only be applied if both time series and steady state measurements are available whereas the exact computation has no such limitations.

4.4 Approximate inference methods

A commonly used approximation to (the logarithm of) the Bayesian score can be obtained by using the Bayesian information criterion (BIC) (Schwarz 1978) or, equivalently, the minimum description length (MDL) principle (Rissanen 1978)

$$\mathrm{BIC}(\mathcal{D}|\mathcal{M}) = \log P(\mathcal{D}|\mathcal{M}, \hat{\theta}) - \frac{d}{2} \log N$$

where $\hat{\theta}$ denotes the ML parameters for $\mathcal{M}$ given $\mathcal{D}$, $d = \sum_{i=1}^{n} q_i(r_i - 1)$ is the number of parameters in the model, and $N = T + M$ is the sample size. Unfortunately, no closed-form solution is available for the computation of $\hat{\theta}$ from a combination of time series and steady state measurements and hence an iterative optimization routine is required. In order to overcome that, we can alternatively consider another approximation where we replace $\hat{\theta}$ by the ML estimate, $\hat{\theta}_A$, that depends on the time series data only. Similar approximation can also be constructed for the MAP and posterior mean estimates, respectively. The use of the MAP or posterior mean instead of the ML estimate is also motivated by Theorem 2, which guarantees that the optimal parameters provide ergodic dynamics.

The above reasoning leads us to consider another alternative, which is a type of semi-BIC approximation and is defined as (when expressed without the logarithm)

$$\mathrm{SBIC}_{\overline{A}}(\mathcal{D}|\mathcal{M}) = \int_\theta P(\mathcal{D}_A|\mathcal{M}, \theta) P(\mathcal{D}_\pi|\mathcal{M}, \overline{\theta}_A) P(\theta|\mathcal{M}) d\theta$$
$$= P(\mathcal{D}_\pi|\mathcal{M}, \overline{\theta}_A) \int_\theta P(\mathcal{D}_A|\mathcal{M}, \theta) P(\theta|\mathcal{M}) d\theta$$
$$= P(\mathcal{D}_\pi|\mathcal{M}, \overline{\theta}_A) P(\mathcal{D}_A|\mathcal{M}), \tag{11}$$

where $\overline{\theta}_A$ is the posterior mean estimate of θ that depends only on the time series data, and the last equality follows from (5) and (8). Again, a similar approximation can be considered for the ML and MAP estimates as well. Note that plugging in the posterior mean estimates, $P(\mathcal{D}_\pi|\mathcal{M}, \overline{\theta}_A)$, does not result in over-fitting since $\mathcal{D}_\pi$ serves as an independent test data set for $\overline{\theta}_A$. An important aspect of the above approximation is that it provides accurate scoring for the time series measurements, which naturally contain more information about the network dynamics than steady state measurements.

A potential problem with the above approximations, or with approximations in general, is their accuracy, especially for small sample sizes. The SBIC-approximation, however, is expected to behave well due to the above mentioned reasons. We also introduce an exact method (exact up to an arbitrary simulation accuracy) later in Sect. 4.6 and compare the exact method with the SBIC-approximation via simulations in Sects. 5 and 6.

4.5 Bayesian posterior estimation using MCMC

Given one of the above scoring criteria, either exact or an approximation, a common approach is to find the highest scoring network (or a set of high scoring networks), i.e.,

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} P(\mathcal{M}|\mathcal{D}).$$

Exhaustive search is prohibitive for all but the smallest networks due to the huge number of network models. Therefore, one typically needs to rely on optimization or estimation procedures. All the methods we propose rely on stochastic estimation methods, in particular, Markov chain Monte Carlo (MCMC) (for a review, see Robert and Casella 2005).

The appropriateness of searching for only the highest scoring network may be questionable, at least in a small sample setting, since the posterior is likely to be relatively flat, i.e., the highest scoring network does not stand out as sufficiently unique. Therefore, in many applications, it is more relevant to consider the full posterior distribution over network models or, in practice, a set of high scoring networks. This can be done by sampling networks directly from the posterior $P(\mathcal{M}|\mathcal{D})$ using MCMC methods. The idea of MCMC methods for DBNs is to construct a Markov chain over network structures, $\{\mathcal{M}_\ell\}_{\ell=1,2,\ldots}$, such that it converges in distribution to the posterior $P(\mathcal{M}|\mathcal{D})$. If the chain $\{\mathcal{M}_\ell\}$ is again aperiodic and irreducible, then it converges to a stationary distribution. Thus, the goal is to construct a transition kernel for $\{\mathcal{M}_\ell\}$ such that the stationary distribution is the desired posterior.

The Metropolis-Hastings (MH) algorithm for BNs was first introduced in (Madigan and York 1995) and was called MC^3, MCMC for model composition. In the MC^3 algorithm, convergence to the desired posterior is obtained as follows. Given the current network $\mathcal{M}$, a new structure $\mathcal{M}'$ is sampled from a proposal distribution $Q(\mathcal{M}'|\mathcal{M})$. For the traditional time series based inference, the proposed structure is accepted with probability

$$R = \min\left\{1, \frac{P(\mathcal{M}'|\mathcal{D})}{P(\mathcal{M}|\mathcal{D})} \times \frac{Q(\mathcal{M}|\mathcal{M}')}{Q(\mathcal{M}'|\mathcal{M})}\right\}. \tag{12}$$

For the SBIC-approximation the above equation can be written as

$$R = \min\left\{1, \frac{\mathrm{SBIC}_{\overline{A}}(\mathcal{D}|\mathcal{M}')}{\mathrm{SBIC}_{\overline{A}}(\mathcal{D}|\mathcal{M})} \times \frac{Q(\mathcal{M}|\mathcal{M}')}{Q(\mathcal{M}'|\mathcal{M})}\right\}. \tag{13}$$

Note that in (12) $\mathcal{D} = \mathcal{D}_A$ whereas in (13) $\mathcal{D} = (\mathcal{D}_A, \mathcal{D}_\pi)$. The proposal distribution introduced in (Madigan and York 1995) is based on the concept of neighborhood of a given

1 Initialization: set $\mathcal{M}_1$
2 For $\ell = 1$ to $L + S - 1$
– Sample $u \sim \mathcal{U}_{[0,1]}$
– Sample $\mathcal{M}' \sim Q(\cdot|\mathcal{M}_\ell)$
– If $u < R$ (Equation (12) or (13))
• $\mathcal{M}_{\ell+1} = \mathcal{M}'$
– Else
• $\mathcal{M}_{\ell+1} = \mathcal{M}_\ell$

Fig. 1 Pseudo-code of the MC^3 algorithm

network $\mathcal{M}$, $\mathcal{N}(\mathcal{M})$, which for DBNs consists of all networks that can be obtained from $\mathcal{M}$ with a single edge removal or addition (Husmeier 2003). The proposal distribution then assigns a uniform probability to all the networks in $\mathcal{N}(\mathcal{M})$, i.e., $Q(\mathcal{M}'|\mathcal{M}) = 1/|\mathcal{N}(\mathcal{M})|$ for all $\mathcal{M}' \in \mathcal{N}(\mathcal{M})$ (otherwise zero). It is easy to see that for DBNs, $|\mathcal{N}(\mathcal{M})| = n^2$, regardless of $\mathcal{M}$. Therefore, this choice of proposal distribution (and neighborhood) guarantees it to be symmetric, $Q(\mathcal{M}'|\mathcal{M}) = Q(\mathcal{M}|\mathcal{M}')$. Consequently, the MH algorithm reduces to the Metropolis algorithm and e.g. (12) can be rewritten as

$$R = \min\left\{1, \frac{P(\mathcal{M}'|\mathcal{D})}{P(\mathcal{M}|\mathcal{D})}\right\}. \tag{14}$$

To summarize, given an initial network $\mathcal{M}_1$, new networks are sampled from $Q(\cdot|\mathcal{M}_\ell)$ and accepted with probability R. If the proposed network is accepted (resp., rejected), then we set $\mathcal{M}_{\ell+1} = \mathcal{M}'$ (resp., $\mathcal{M}_{\ell+1} = \mathcal{M}_\ell$). After a proper burn-in period L, a dependent sample $\{\mathcal{M}_{L+1}, \mathcal{M}_{L+2}, \ldots, \mathcal{M}_{L+S}\}$ is collected from the chain. A pseudo-code of the MC^3 algorithm is shown in Fig. 1.

In order to score steady state measurements in the SBIC-approximation, one needs to solve for the posterior mean parameter estimates $\overline{\theta}_A$ as well as the steady state distribution corresponding to $(\mathcal{M}', \overline{\theta}_A)$ during each MCMC iteration. Note, however, that the consecutive networks in a chain differ only by at most one edge and that allows a more efficient way of computing the Bayes factors in (12) and (13) (Madigan and York 1995) (see also (4) and (8)). Also note that the intractable term $P(\mathcal{D})$ and the uniform prior over networks cancel out; i.e., $\frac{P(\mathcal{M}'|\mathcal{D})}{P(\mathcal{M}|\mathcal{D})} = \frac{P(\mathcal{D}|\mathcal{M}')}{P(\mathcal{D}|\mathcal{M})}$.

The chain $\{\mathcal{M}_\ell\}$ allows us to estimate the full posterior $P(\mathcal{M}|\mathcal{D})$ over all $\mathcal{M}$. It is typically of interest to look at the marginal posterior probabilities of network edges $P(E_{ij}|\mathcal{D}) = \sum_{\mathcal{M}} \mathbb{I}_{ij}(\mathcal{M}) P(\mathcal{M}|\mathcal{D})$, where $\mathbb{I}_{ij}(\cdot)$ is the indicator function for the edge from node X_i to node X_j. These quantities can be directly estimated using a chain as $\hat{P}(E_{ij}|\mathcal{D}) = \frac{1}{S}\sum_{\ell=L+1}^{L+S} \mathbb{I}_{ij}(\mathcal{M}_\ell)$, which converges to the true posterior edge probability almost surely.

4.6 Exact model inference using trans-dimensional MCMC

The previous section introduced approximations to the marginal likelihood. An alternative approach attempts to solve the intractable integral directly without any approximations. A naive solution would try to go through all DBN network structures and apply a separate MCMC estimation (or numerical integration) procedure to (9). This is computationally intractable, given the enormous number of different network structures. Alternatively, one

could use the MC^3 algorithm (over $\mathcal{M}$) discussed in the previous section and combine that with another MCMC estimation (over θ). In other words, given the current model $\mathcal{M}_\ell$, one could construct another chain over its parameters θ to compute the marginal likelihood. However, this would result in an inefficient computation where chains $\{\mathcal{M}_\ell\}$ and $\{\theta_\ell\}$ were run independently in a nested fashion.

A standard solution is to construct a trans-dimensional MCMC sampler that exploits relationships between model parameters in different models. For this purpose, we propose to apply a so-called reversible jump MCMC (RJMCMC) method introduced in (Green 1995). For a review, see (Andrieu et al. 2001; Robert and Casella 2005). Instead of sampling network models and their parameters independently, a RJMCMC samples in the "product space" of $\{\mathcal{M}, \theta\}$, more precisely in

$$\mathcal{S} = \bigcup_{\mathcal{M}} (\{\mathcal{M}\} \times \theta_{\mathcal{M}}), \tag{15}$$

where the union is over all $2^{(n^2)}$ DBN model structures and $\theta_{\mathcal{M}}$ denotes the parameters of a specific model structure $\mathcal{M}$.

The method developed by Green (1995) is extremely flexible. Here we consider a particular implementation that suffices to address our problem. See also (Dellaportas and Forster 1999; Giudici et al. 2000; Pournara 2004) for related sampling approaches to undirected and directed decomposable network models. Assume that the current state of the chain is $\{\mathcal{M}, \theta\}$ and that a proposed new state is $\{\mathcal{M}', \theta'\}$ where the dimensionality of θ', $\dim(\theta')$, is higher than that of θ. Further, assume that θ' is obtained from θ and a variable $\varphi_{\mathcal{M},\mathcal{M}'}$ via a bijective mapping $\theta' = f_{\mathcal{M},\mathcal{M}'}(\theta, \varphi_{\mathcal{M},\mathcal{M}'})$, $\dim(\theta') = \dim(\theta) + \dim(\varphi_{\mathcal{M},\mathcal{M}'})$, and $\varphi_{\mathcal{M},\mathcal{M}'}$ itself is proposed from a distribution $q_{\mathcal{M},\mathcal{M}'}(\cdot|\theta)$. The acceptance probability for the proposed move that satisfies the detailed balance conditions is (Green 1995)

$$R = \min\{1, R_a\}, \tag{16}$$

where

$$R_a = \frac{P(\mathcal{M}', \theta'|\mathcal{D})\, Q(\mathcal{M}|\mathcal{M}')}{P(\mathcal{M}, \theta|\mathcal{D})\, Q(\mathcal{M}'|\mathcal{M}) q_{\mathcal{M},\mathcal{M}'}(\varphi_{\mathcal{M},\mathcal{M}'}|\theta)} \left| \frac{\partial f_{\mathcal{M},\mathcal{M}'}(\theta, \varphi_{\mathcal{M},\mathcal{M}'})}{\partial(\theta, \varphi_{\mathcal{M},\mathcal{M}'})} \right| \tag{17}$$

and $|\cdot|$ denotes the determinant. The corresponding move from $\{\mathcal{M}', \theta'\}$ to $\{\mathcal{M}, \theta\}$ is accepted with probability

$$R = \min\{1, R_d\}, \tag{18}$$

where $R_d = 1/R_a$ and $\varphi_{\mathcal{M},\mathcal{M}'}$ is obtained from the inverse transformation $(\theta, \varphi_{\mathcal{M},\mathcal{M}'}) = f^{-1}_{\mathcal{M},\mathcal{M}'}(\theta')$. Note that $P(\mathcal{M}', \theta'|\mathcal{D}) = P(\mathcal{D}|\mathcal{M}', \theta') P(\mathcal{M}', \theta')/P(\mathcal{D})$ and $P(\mathcal{D}|\mathcal{M}', \theta') = P(\mathcal{D}_A|\mathcal{M}', \theta') P(\mathcal{D}_\pi|\mathcal{M}', \theta')$ due to the assumption of conditional independence of $\mathcal{D}_A$ and D_π given a DBN.

We consider the same symmetric proposal distribution for network structures as in Sect. 4.5 that proposes to either add or delete one edge at a time. Hence, the Q terms cancel out from (16). Assume first that a move from $\{\mathcal{M}, \theta\}$ to $\{\mathcal{M}', \theta'\}$ involves adding the edge E_{ai}, i.e., edge from $X_a[t-1]$ to $X_i[t]$. In the case of binary networks, which are also considered in Sects. 5 and 6, we sample $\varphi_{\mathcal{M},\mathcal{M}'}$ from the uniform distribution over continuous volume $(0, 1)^{q_i}$, where q_i denotes the number of parent configurations for the ith node in $\mathcal{M}$. Since the volume of the unit hypercube is 1, we have $q_{\mathcal{M},\mathcal{M}'}(\varphi_{\mathcal{M},\mathcal{M}'}|\theta) \equiv 1$. The uniform proposal distribution provides a general approach, especially if no prior knowledge of parameters is available, as is typically the case. Note that each θ_{ij} has only one free

parameter because $\theta_{ij2} = 1 - \theta_{ij1}$. The previous conditional probabilities θ_i and proposed values $\varphi_{\mathcal{M},\mathcal{M}'}$ are transformed into new conditional probabilities θ_i' directly using the identity function such that the previous (resp., proposed) conditional probabilities are used for those input configurations for which the newly added parent node takes on value one (resp., two). Conditional probabilities of other nodes remain the same, i.e., $\theta_j' = \theta_j$ for $j \neq i$. Thus, the determinant of the Jacobian matrix is 1. Consequently, (16–17) reduce to

$$R = \min\left\{1, \frac{P(\mathcal{M}', \theta'|\mathcal{D})}{P(\mathcal{M}, \theta|\mathcal{D})}\right\}. \tag{19}$$

Again, the reverse move from $\{\mathcal{M}', \theta'\}$ to $\{\mathcal{M}, \theta\}$ is defined automatically by the bijective mapping $f_{\mathcal{M},\mathcal{M}'}$.

A similar construction can also be defined for networks with $r_i > 2$ but now modified to satisfy the constraints $\theta_{ijk} \geq 0$ and $\sum_{k=1}^{r_i} \theta_{ijk} = 1$. Assume again that the proposal move involves adding the edge E_{ai}. As above, let q_i denote the number of parent configurations for node X_i in the current network $\mathcal{M}$. The number of parent configurations for X_i in the proposed network $\mathcal{M}'$ is $q_i' = r_a q_i$. One can proceed, e.g., as above and use the previous conditional probabilities for those input configurations for which the newly added parent node X_a takes on value one. For the $q_i' - q_i$ "new" parent configurations we can propose new conditional probabilities ($\varphi_{\mathcal{M},\mathcal{M}'}$) uniformly randomly from the r_i-dimensional unit simplex (but excluding boundary of the simplex to avoid zero probabilities). This is equivalent to drawing $q_i' - q_i$ independent samples from the r_i-dimensional Dirichlet distribution with all hyperparameters equal to one. Note that this proposal procedure for $r_i > 2$ is a generalization of the binary case. The determinant of the Jacobian matrix is again 1 but now $q_{\mathcal{M},\mathcal{M}'}(\varphi_{\mathcal{M},\mathcal{M}'}|\theta) \neq 1$ and so it cannot be ignored.

In addition to reversible jumps between different network structures and their parameters, RJMCMC can also propose a so-called null move where the network structure remains the same but the parameters are updated using the standard MH step. Given the current state $\{\mathcal{M}, \theta\}$, new parameters in the standard MH step are sampled from a proposal distribution $q_{\mathcal{M},\mathcal{M}}(\cdot|\theta)$ and a null move from $\{\mathcal{M}, \theta\}$ to $\{\mathcal{M}, \theta'\}$ is accepted with probability

$$R = \min\left\{1, \frac{P(\mathcal{M}, \theta'|\mathcal{D}) q_{\mathcal{M},\mathcal{M}}(\theta|\theta')}{P(\mathcal{M}, \theta|\mathcal{D}) q_{\mathcal{M},\mathcal{M}}(\theta'|\theta)}\right\}. \tag{20}$$

In our implementation, the proposal distribution $q_{\mathcal{M},\mathcal{M}}(\cdot|\theta)$ first selects a node uniformly randomly and proposes new parameter values for the selected node. Given that the ith node is selected by $q_{\mathcal{M},\mathcal{M}}$, then new values for all θ_{ij1}, $j \in \{1, \ldots, q_i\}$, are proposed independently from the $(0, 1)$ truncated normal distribution with mean θ_{ij1} and standard deviation $\sigma = 0.1$. Note again that $\theta_{ij2} = 1 - \theta_{ij1}$. Finally, a reversible jump and null move are proposed with probability $\beta = 1/2$ and $1 - \beta = 1/2$, respectively.

A similar construction can again be obtained for networks with $r_i > 2$. For example, new conditional probabilities θ_{ij}' can be proposed sequentially such that θ_{ijk}' (for $k = 2, \ldots, r_i - 1$) is proposed from $(\theta_{ij(k-1)}', \theta_{ij(k+1)})$ truncated normal distribution. For $k = 1$ and $k = r_i$ left and right limits of the truncated normal distribution need to be 0 and 1, respectively.

To summarize, given an initial model $\mathcal{M}_1$ and the corresponding initial parameters θ_1, the RJMCMC proceeds with a reversible jump (resp., null) move with probability β (resp., $1 - \beta$). The proposed reversible jump (resp., null) move is accepted with the probability shown in (16) or (18) (resp., (20)). If the proposed reversible jump (resp., null) move is accepted, then we set $(\mathcal{M}_{\ell+1}, \theta_{\ell+1}) = (\mathcal{M}', \theta')$ (resp., $(\mathcal{M}_{\ell+1}, \theta_{\ell+1}) = (\mathcal{M}, \theta')$). If the proposed move is rejected, then we set $(\mathcal{M}_{\ell+1}, \theta_{\ell+1}) = (\mathcal{M}, \theta)$. After a proper burn-in period L, a dependent

1 Initialization: set $(\mathcal{M}_1, \theta_1)$
2 For $\ell = 1$ to $L + S - 1$
- Sample $u \sim \mathcal{U}_{[0,1]}$ and $v \sim \mathcal{U}_{[0,1]}$
- If $u < \beta$ ("jump")
  - Sample $\mathcal{M}' \sim Q(\cdot|\mathcal{M}_\ell)$
  - If $\dim(\mathcal{M}') > \dim(M_\ell)$ (add an edge)
    - ◇ Sample $\varphi_{\mathcal{M}_\ell,\mathcal{M}'} \sim q_{\mathcal{M}_\ell,\mathcal{M}'}(\cdot|\theta_\ell)$ and set $\theta' = f_{\mathcal{M}_\ell,\mathcal{M}'}(\theta_\ell, \varphi_{\mathcal{M}_\ell,\mathcal{M}'})$
    - ◇ If $v < R = \min\{1, R_a\}$ (Equation (16))
      - ○ $(\mathcal{M}_{\ell+1}, \theta_{\ell+1}) = (\mathcal{M}', \theta')$
    - ◇ Else
      - ○ $(\mathcal{M}_{\ell+1}, \theta_{\ell+1}) = (\mathcal{M}_\ell, \theta_\ell)$
  - Else (remove an edge)
    - ◇ $(\theta', \varphi_{\mathcal{M}',\mathcal{M}_\ell}) = f^{-1}_{\mathcal{M}',\mathcal{M}_\ell}(\theta_\ell)$
    - ◇ If $v < R = \min\{1, R_d\}$ (Equation (18) with variables corresponding to a move from $(\mathcal{M}_\ell, \theta_\ell)$ to $(\mathcal{M}', \theta')$)
      - ○ $(\mathcal{M}_{\ell+1}, \theta_{\ell+1}) = (\mathcal{M}', \theta')$
    - ◇ Else
      - ○ $(\mathcal{M}_{\ell+1}, \theta_{\ell+1}) = (\mathcal{M}_\ell, \theta_\ell)$
- Else ("null")
  - Sample $\theta' \sim q_{\mathcal{M}_\ell,\mathcal{M}_\ell}(\cdot|\theta_\ell)$
  - If $v < R$ (Equation (20))
    - ◇ $(\mathcal{M}_{\ell+1}, \theta_{\ell+1}) = (\mathcal{M}_\ell, \theta')$
  - Else
    - ◇ $(\mathcal{M}_{\ell+1}, \theta_{\ell+1}) = (\mathcal{M}_\ell, \theta_\ell)$

Fig. 2 Pseudo-code of the RJMCMC algorithm

sample $\{(\mathcal{M}_{L+1}, \theta_{L+1}), (\mathcal{M}_{L+2}, \theta_{L+2}), \ldots, (\mathcal{M}_{L+S}, \theta_{L+S})\}$ is collected. A pseudo-code of the RJMCMC algorithm is shown in Fig. 2.

As in the case of the SBIC-approximation algorithm, one needs to solve for the steady state distribution corresponding to a proposed network during each RJMCMC iteration. However, the consecutive networks in a chain again differ only by at most one edge and that allows a more efficient computation of (16–20). Finally, note that the above RJMCMC method provides us a way to estimate the full posterior $P(\mathcal{M}, \theta|\mathcal{D})$ over $\mathcal{S}$ (see (15)) as well as the marginal posterior probability $P(\mathcal{M}|\mathcal{D}) = \int_\theta P(\mathcal{M}, \theta|\mathcal{D})d\theta$ for all $\mathcal{M}$ without any approximations.

Assuming the current DBN $\{\mathcal{M}, \theta\}$ satisfies the sufficient condition of Theorem 1, then the above RJMCMC construction guarantees that so does the proposed network. First, the detailed balance condition is satisfied by construction of Green's RJMCMC method. Second, aperiodicity and irreducibility of the chain are seen using the same reasoning as, e.g., in (Robert et al. 2000; Richardson and Green 1997). Aperiodicity follows from the fact that for any arbitrarily small neighborhood of the current state $\{\mathcal{M}, \theta\}$ there is a positive probability that the state is in that neighborhood after one step of the RJMCMC procedure. Irreducibility of the chain follows from the fact that any model structure $\mathcal{M}'$ can be ob-

tained from $\mathcal{M}$ by repeatedly adding or deleting one edge at a time and the parameters θ are sampled from continuous distributions whose supports are the whole parameter spaces.

The above discussion assumes a model where time series and steady state measurements are generated by exactly the same model $(\mathcal{M}, \theta)$. In some applications, it might be realistic to assume a different model where $\mathcal{D}_A$ and $\mathcal{D}_\pi$ are associated with the same network structure but with different parameters θ_A and θ_π, i.e., parameters are not coupled. This modification can be accounted for with minor changes. Assuming an independent prior for θ_A and θ_π as in Sect. 4.1, the marginal likelihood factorizes as $P(\mathcal{D}_A, \mathcal{D}_\pi|\mathcal{M}) = P(\mathcal{D}_A|M)P(\mathcal{D}_\pi|\mathcal{M})$ (see (9)). Consequently, it also follows that the posterior probability factorizes similarly $P(\mathcal{M}|\mathcal{D}_A, \mathcal{D}_\pi) \propto P(\mathcal{M}|\mathcal{D}_A)P(\mathcal{M}|\mathcal{D}_\pi)/P(\mathcal{M})$. Note that these factorizations only apply when $\mathcal{D}_A$ and $\mathcal{D}_\pi$ are associated with separate parameters and parameters are a priori independent. When this factorization is true, posterior probabilities of a network can be obtained separately for the two data sets $\mathcal{D}_A$ and $\mathcal{D}_\pi$ and, thus, parameter coupling is not needed in the RJMCMC method. Because $P(\mathcal{M}|\mathcal{D}_A)$ can be computed in closed-form the posterior $P(\mathcal{M}|\mathcal{D}_A)$ can be sampled efficiently with the MC^3 algorithm. Sampling from $P(\mathcal{M}|\mathcal{D}_\pi)$, however, requires the RJMCMC method, but now without the parameter coupling. Note that the model parameters remain coupled also in the case where $\mathcal{D}_A$ and $\mathcal{D}_\pi$ are associated with different parameters but if there is prior knowledge of relationships between θ_A and θ_π, i.e., priors for θ_A and θ_π are dependent. The question of whether the two data sources $\mathcal{D}_A$ and $\mathcal{D}_\pi$ should be modeled as being generated by the same or separate parameters is problem dependent. A general class of problems where parameters can be considered to be different is one where time series and steady state measurements have different noise levels. Even in this case, however, one might have prior knowledge of θ_A and θ_π that could be reflected using a dependent prior for θ_A and θ_π, hence requiring the coupled approach. We consider both the coupled and uncoupled cases in our simulations.

4.7 Computational complexity

Each iteration of the standard time series based MCMC algorithm requires computing the acceptance probability shown in (12). Because $\mathcal{M}$ and $\mathcal{M}'$ (or $\mathcal{M}_\ell$ and $\mathcal{M}'$ in pseudo-code in Fig. 2) differ only by one edge and the score factorizes over nodes, the Bayes factor needs to be computed only for a single node, X_i say, that has different parents in $\mathcal{M}$ and $\mathcal{M}'$. From (8) we see that computation of the marginal likelihood for a single node requires counting instances N_{ijk} (time complexity proportional to the length of the time series, $O(T)$) and computing a product of $q_i \times r_i$ terms.

Each MCMC step of the SBIC-approximation requires computing the same node-wise Bayes factor as in the standard time series based analysis. In addition, the posterior mean estimate $\overline{\theta}_A$ that depends on time series data only is needed, which has the same time complexity as the node-wise Bayes factor. Scoring of the steady state measurements requires solving for the steady state vector π of the transition matrix A corresponding to the proposed network $\mathcal{M}'$ and parameters $\overline{\theta}_A$. Let s denote the size of the state space, i.e., $s = \prod_{i=1}^{n} r_i$. Each element of A is a product of n terms but $n-1$ of those terms remain the same between consecutive networks and thus updating A from $\mathcal{M}_\ell$ to $\mathcal{M}'$ has complexity $O(s^2)$. The standard exact solution for the eigenvector problem, such as the one based on the QR factorization, has complexity $O(s^3)$. Advanced algorithms for the eigenvector problem are more efficient but they are typically based on iterative optimization approaches. Thus, (average) asymptotic time complexity depends on particular problem at hand (e.g. properties/sparseness of A) as well as the convergence criterion. We found that ARPACK routine (Lehoucq et al. 1998) is remarkably more efficient than the standard approach. Overall, computational complexity

of the proposed SBIC-approximation is about the same as that of the standard Bayesian inference from time series data, except that the approximate method needs to solve the steady state distribution of a DBN during each step of the MCMC. The cost of solving for the steady state distribution is relatively small for small networks but becomes substantial for larger networks.

The computational complexity of the proposed RJMCMC method, in turn, is similar to that of the SBIC-approximation. Instead of computing the node-wise Bayes factor, time series based node-wise likelihood term $P(\mathcal{D}_A|\mathcal{M}', \theta')$ or $P(\mathcal{D}_A|\mathcal{M}, \theta')$ is needed that can again be computed in time $O(T)$. Scoring of the steady state measurements, which is the most time consuming operation (asymptotically) of each step, is comparable with that of the approximate method. However, the approximate and the standard Bayesian methods sample in the space of network structures whereas the proposed exact method samples both network structures and their parameters. Therefore, to guarantee sufficient convergence and thereby accurate estimation, the higher dimensional parameter space of the exact RJMCMC method typically requires approximately an order of magnitude longer chain than the less complex MCMC algorithm. See Sect. 7 for a discussion on improving the computational complexity.

As discussed above, exact algorithmic time complexity depends on several problem dependent parameters. The most important parameter is the length of the (parallel) chains which, in turn, needs to be chosen such that the chosen convergence criteria are met and posterior estimates are sufficiently accurate. Running times (per simulation) on a single standard desktop of our non-optimized Matlab routines for the type of simulations performed in this work are approximately as follows: hour(s) for the standard time series based analysis, "less than a day" for the approximate method, and "a couple of days" for the RJMCMC method.

4.8 Convergence assessments

MCMC methods provide powerful tools for simulating from complex target distributions. Moreover, the resulting chains are proven to converge to the desired densities under fairly mild conditions. Convergence results, however, are only guaranteed when the number of iterations approaches infinity. Therefore, for any application of MCMC, convergence of the resulting finite chains needs to be carefully assessed, and this represents one of the main difficulties for the application of MCMC methods. Convergence assessment is even more difficult for the MCMC-based model selection methods, such as the one explained above, since the number of different models can be prohibitively high and the length of the parameter vector varies from iteration to iteration.

It is commonly observed that no single convergence diagnostic is capable of providing sufficiently reliable convergence assessment. Therefore, it is suggested that two or more diagnostics are applied together. In this study we use a non-parametric convergence assessment method that has been developed particularly for model selection problems (Brooks et al. 2003) and a general purpose method that compares the estimated posterior probabilities of the network connections from two independent simulations (see, e.g., Husmeier 2005). Both Brooks et al.'s and Husmeier's diagnostics assume that one has several ($J \geq 2$) independent chains, which we also assume here. Let us briefly describe the method of Brooks et al. first.

The underlying idea is to compare the posterior probabilities of network models as estimated from J independent chains. In the case of DBNs, the cardinality of different network models is $2^{(n^2)}$ and thus, unless n is very small, it is impossible to monitor the posterior probability of all of them. A way around is to combine similar models and monitor the behavior of groups of models. As suggested by Brooks et al. (2003), each model $\mathcal{M}_i$ is labeled

by the number of edges, $\nu(\mathcal{M}_i)$, that it possesses and models with the same number of edges are grouped. This model indicator should reflect the complexity of the model.

In order to define a statistical test for the convergence (or rather divergence), the individual chains are subsampled to obtain approximately independent samples. Following the reasoning in (Brooks et al. 2003), an estimate of the thinning parameter for sub-sampling the jth chain is obtained as $\lambda^j = \frac{\log p}{\log \rho^j}$, where p is the accepted error level (of independency) and ρ^j is an estimate of the convergence rate. In the following, we use $p = 0.01$. An estimate of the convergence rate, in turn, can be obtained as follows. For each chain $j = 1, \ldots, J$, compute the state transition matrix between model groups $\{\mathcal{M}_i | \nu(\mathcal{M}_i) = k\}$ and $\{\mathcal{M}_i | \nu(\mathcal{M}_i) = l\}$ as $P^j_{kl} = N^j_{kl} / \sum_{l=0}^{n^2} N^j_{kl}$, where N^j_{kl} denotes the number of times a move from model group k to model group l is observed in the jth chain. The convergence rate ρ^j can be estimated by the second largest eigenvalue of P^j (with possible null rows removed). The final λ is then obtained by averaging the individual λ^js and the sub-sampling keeps only every λth sample of the original chains.

The actual convergence diagnostic is implemented using a goodness of fit test. Of the two alternative methods introduced in (Brooks et al. 2003) we choose to use the chi-squared test since that generally provides a more conservative diagnostic than the corresponding Kolmogorov-Smirnov test. Let N^j_ν denote the number of times model group ν is observed in the jth sub-sampled chain. Assuming the individual chains are homogeneous, i.e., $P(N^i_\nu) = P(N^j_\nu)$ for all ν, i and j, the expected number of counts can be defined as $E_\nu = \sum_{j=1}^{J} N^j_\nu / J$. The goodness of test diagnostic is then based on the following statistic

$$\chi^2(\ell) = \sum_{\nu=0}^{n^2} \sum_{j=1}^{J} \frac{(N^j_\nu - E_\nu)^2}{E_\nu}$$

which, under the homogeneity assumption, is asymptotically chi-squared distributed with $(J-1)n^2$ degrees of freedom. The above χ^2 statistic can be computed for different values of $\ell = L, L+\lambda, L+2\lambda, \ldots, L+S$ (assuming for simplicity that S is a multiple of λ). If the significance values corresponding to the above test get below a certain threshold, say, $\alpha = 0.1$ to be fairly stringent, then there is statistical evidence that the null hypothesis does not hold and the chains have not converged. Otherwise the test shows no significant evidence against the null hypothesis and the chains are considered to be homogeneous. We take a similar approach as in (Brooks et al. 2003) in that a drop below the threshold α for a few iterations in the beginning or middle of the chain does not cause the chain pair to get rejected. Rejection requires the significance value to be below α for a larger number of consecutive iterations (see Sect. 6 for illustrative examples).

The additional diagnostic we consider here (see Friedman and Koller 2003) plots the estimated posterior probabilities of the network edges from two independent simulations with different initial values and random number seeds. As above, by assuming that the two chains have converged to the same stationary distribution, marginal probabilities of network edges are also equivalent, and the scatter plot of the estimated posterior probabilities should lie tightly around the diagonal. Note that this test is similar with the method of Brooks et al. except that it is based on the full dependent sample and marginal probabilities of individual network connections (instead of model groups). No formal test has been developed for this diagnostic but one can employ, as a heuristic, an error threshold below which each pair of probabilities must be.

The aforementioned two diagnostics are used to assess the convergence of pairs of chains. Once a chain pair that does not get rejected by either of the tests is found, the mean of the

estimated posterior edge probabilities, $\hat{P}(E_{ij}|\mathcal{D})$, from those two chains is used as the final estimate. Note that although the chains are subsampled in one of the diagnostics, the final estimates are computed based on the full (dependent) sample.

5 Simulations

We assess the performance of the proposed methods and compare it to that of the standard state-of-the-art Bayesian inference method (see (8)) that can only handle time series measurements. Note that previously no principled method has been introduced that can infer dynamic networks from steady state measurements, either alone or in combination with time series measurements. In particular, we consider the $\text{SBIC}_{\overline{A}}$-approximation shown in (11) with the posterior mean parameter estimates and the RJMCMC method introduced in Sect. 4.6. The standard Bayesian inference method is applied to time series data alone whereas the $\text{SBIC}_{\overline{A}}$-method is applied to a combination of time series and steady state data. For the RJMCMC-based exact method we consider two scenarios: a combination of time series and steady state data, and steady state data alone. The same hyperparameter values are used in all the methods and they are set such that Theorem 2 holds for the approximate method, in particular $\alpha_{ijk} = 1$. This choice of pseudo counts also gives a non-informative parameter prior. As above, we assume a non-informative structure prior. The MCMC estimation procedure is used for both the traditional and proposed approximate methods whereas the RJMCMC estimation is used for the exact model inference. Burn-in and sampling period lengths are $L = 5 \times 10^5$ and $S = 5 \times 10^5$ (resp., $L = 10^6$ and $S = 10^6$) for smaller (resp., larger) network sizes. Convergence is assessed using the methods described in Sect. 4.8. In a more sophisticated approach, one would monitor the convergence (of multiple parallel chains) during the simulation and determine the sufficient number of MCMC iterations online. The simple approach of running each chain a predetermined number of times works well in our case.

All the performance comparisons are based on simulated data. Different network models are considered. The considered network sizes are $n = 6$ or $n = 8$ and binary-valued nodes are used throughout the simulations. The structure of a DBN is chosen uniformly randomly from the space of n-variable DBNs that contain a certain number of edges: 12 for $n = 6$, and 18 for $n = 8$. Parameters of the conditional distributions are chosen randomly such that for each i and j one of the θ_{ijk}s is set to 0.9, and the remaining entries have equal probability, i.e., 0.1. Similar parameter values are used, e.g., in (Husmeier 2003). If smaller values of conditional probabilities, say 0.7 and 0.3, are used, then the networks behave "more randomly" and performance of all the inference methods decreases, though they retain their relative performance and order. In a simulation that is motivated by computational biology studies, we also vary the value of θ_{ijk}s in order to produce a biologically more realistic setting. In the same way as a variable in a deterministic function can be fictitious, some of the parent variables in a random DBN may have no effect on the local conditional probabilities. Edges corresponding to such parent nodes are not considered when analyzing the inference results. There are on average about 1–2 (resp., 2–3) such edges in $n = 6$ (resp., $n = 8$) networks. The size of the time series and steady state samples are set to $T = 25$ and $M = 50$. The initial state of the time series, $\mathbf{x}[1]$, is chosen randomly from the steady state distribution of the corresponding DBN. Each simulation is repeated in a Monte Carlo fashion 20 times. For each iteration, a random DBN is chosen from which random data sets, $\mathcal{D}_A$ and $\mathcal{D}_\pi$, are drawn. For each MCMC and RJMCMC run separately, the initial DBN structure is chosen uniformly randomly from DBNs where each node has exactly one parent. The corresponding parameters are chosen uniformly randomly.

Results are summarized using receiver operating characteristics (ROC) curves (see, e.g., Fawcett 2006). Let $\Omega(\epsilon) = \{E_{ij} \mid \hat{P}(E_{ij}|\mathcal{D}) \geq \epsilon\}$ denote the set of edges whose estimated posterior probability is above a threshold $\epsilon \in [0, 1]$. For each simulation (for which we also know the underlying true network), we can use $\Omega(\epsilon)$ to compute the specificity and sensitivity for different values of ϵ and form the ROC curve. The ROC curves are averaged over the independent Monte Carlo repetitions using vertical averaging (Fawcett 2006). We also compute the area under the ROC curve in order to get a single number that measures the performance. In all the figures and tables below, the traditional method is denoted by "MCMC: $\mathcal{D} = (\mathcal{D}_A)$," and the new methods are denoted by "MCMC (appr.): $\mathcal{D} = (\mathcal{D}_A, \mathcal{D}_\pi)$," "RJMCMC: $\mathcal{D} = (\mathcal{D}_A, \mathcal{D}_\pi)$," and "RJMCMC: $\mathcal{D} = (\mathcal{D}_\pi)$," corresponding to the $\text{SBIC}_{\overline{A}}$-approximation, and the RJMCMC method when applied to both time series and steady state data sets, and to steady state data set alone, respectively.

All the methods have been implemented in Matlab by making use of the Bayes net toolbox written by Murphy (2001) and an additional Markov chain Monte Carlo software by Husmeier (2003).[3] Software implementing the proposed methods will be made publicly available.

6 Results

Before presenting the actual inference results, let us take a look at representative steady state distributions of DBNs. Figure 3 shows two illustrative steady state distributions along with the corresponding DBN network structures that correspond to the two cases $n = 6$ (a–b) and $n = 8$ (c–d), respectively. For each (random) DBN, the steady state distribution can be solved as explained in Sect. 3 and the steady state measurements are sampled directly from the corresponding distributions. In Fig. 3(b), for example, there are four states (corresponding to integer representations 24, 30, 31, and 32) at which the dynamic network spends most of its time in the long-run. Those four states capture approximately 45% of the mass of the steady state distribution.

6.1 Inference results

Figures 4(a) and (b) show the ROC curves for different learning methods and for the two different cases, $n = 6$ and $n = 8$, respectively. Both subplots show the results for the traditional method (dashed blue), the proposed approximate method (dotted green), and for the RJMCMC method when applied to a combination of time series and steady state data (solid red) and to steady state data alone (dash-dotted cyan).

First, the results show that dynamic network models can be learned to an extent from steady state measurements alone. By comparing inference results that are based on steady state data alone to those that (also) incorporate time series data, one can conclude that the amount of dynamic information in steady state measurements is smaller than that in time series measurements, as expected. However, the proposed RJMCMC method provides a statistically rigorous method for inferring dynamic network models from steady state data alone.

Second, ROC curves in the upper left corner (dashed blue, dotted green, and solid red graphs) illustrate the performance of the different MCMC and RJMCMC methods when

[3]The latter software is also publicly available at: http://www.bioss.sari.ac.uk/~dirk/software/DBmcmc.

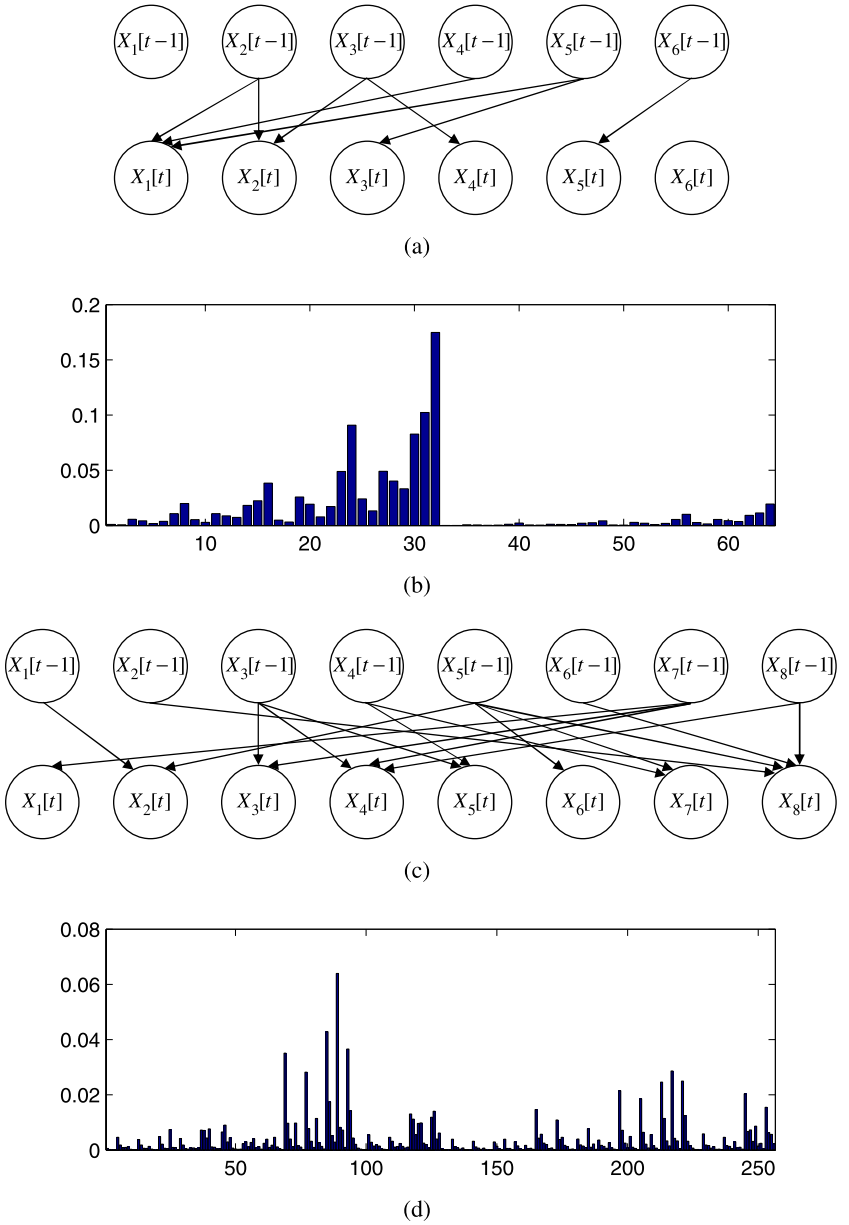

Fig. 3 Two illustrative (random) DBN network structures and the corresponding steady state distributions: (**a–b**) $n = 6$, and (**c–d**) $n = 8$. *Horizontal axis* in subplots (**b**) and (**d**) corresponds to the value of the state vector and is encoded as an integer

applied to either time series data alone or to both time series and steady state measurements. For small values of the complementary specificity (x-axis), the proposed approximate method seems to provide similar or only slightly better results than the traditional Bayesian inference. However, performance of the approximate method clearly exceeds that

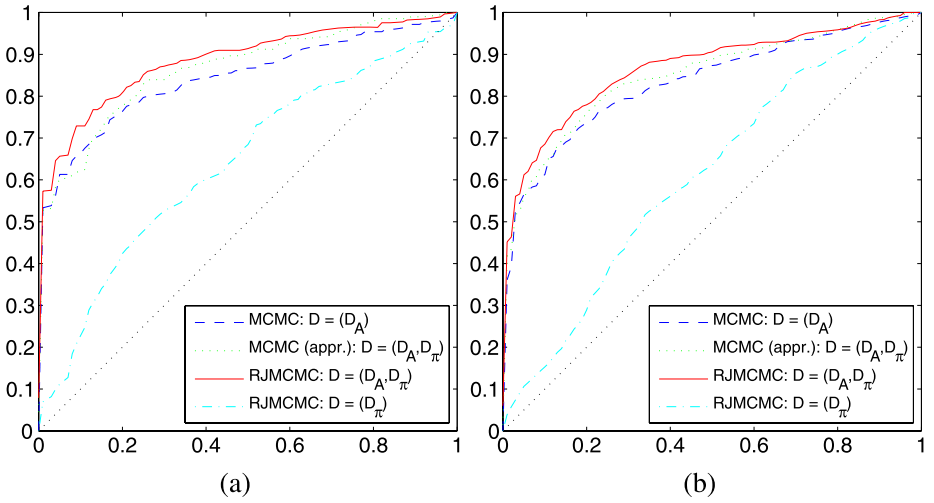

Fig. 4 The ROC curves for the DBN structure learning: (**a**) case $n = 6$, and (**b**) case $n = 8$. Different graphs are coded as follows: the traditional method (*dashed blue*), the proposed approximate method (*dotted green*), and the RJMCMC method when applied to both time series and steady state data (*solid red*) and to steady state data alone (*dash-dotted cyan*)

Table 1 The area under the curve (AUC) for the detection of network edges

Method \ simulation	Case $n = 6$	Case $n = 8$
MCMC: $\mathcal{D} = (\mathcal{D}_A)$	0.843	0.833
MCMC (appr.): $\mathcal{D} = (\mathcal{D}_A, \mathcal{D}_\pi)$	0.865	0.848
RJMCMC: $\mathcal{D} = (\mathcal{D}_A, \mathcal{D}_\pi)$	0.884	0.866
RJMCMC: $\mathcal{D} = (\mathcal{D}_\pi)$	0.639	0.609

of the traditional Bayesian method for larger values of the complementary specificity. The RJMCMC method, in turn, which computes the analytically intractable integral in marginal likelihood directly, consistently outperforms the traditional Bayesian inference as well as the proposed approximate method. A benefit of the approximate method over the RJMCMC method is that the additional complexity of sampling in the product space is avoided. However, decreased computational complexity comes with decreased accuracy.

Overall, the ROC curves show that the proposed methods provide improved inference results. The amount of improvement is moderate which, again, reflects the fact that the amount of (dynamic) information in steady state measurements is smaller than that in time series. However, Figs. 4(a) and (b) clearly show that using the proposed RJMCMC method with both time series and steady state measurements provides more accurate results.

The ROC curves can be summarized and represented as a single number by computing the area under the curve (AUC). AUC provides an intuitive scalar measure that is independent of the shape of the curve. AUCs are computed from the averaged ROC curves using the standard trapezoidal integration method. AUC measures for different methods and different scenarios are listed in Table 1. These performance scores are in good agreement with the ROC curves presented in Figs. 4(a) and (b).

While AUC measures in Table 1 provide useful information about average performance of different methods, it is also useful to assess their variability. To address this, we compute

Table 2 Paired difference and standard deviation in AUC measures for different methods

	Mean	Standard deviation
Simulation: case $n = 6$		
MCMC (appr.): $\mathcal{D} = (\mathcal{D}_A, \mathcal{D}_\pi)$ – MCMC: $\mathcal{D} = (\mathcal{D}_A)$	0.021	0.054
RJMCMC: $\mathcal{D} = (\mathcal{D}_A, \mathcal{D}_\pi)$ – MCMC: $\mathcal{D} = (\mathcal{D}_A)$	0.041	0.044
Simulation: case $n = 8$		
MCMC (appr.): $\mathcal{D} = (\mathcal{D}_A, \mathcal{D}_\pi)$ – MCMC: $\mathcal{D} = (\mathcal{D}_A)$	0.015	0.046
RJMCMC: $\mathcal{D} = (\mathcal{D}_A, \mathcal{D}_\pi)$ – MCMC: $\mathcal{D} = (\mathcal{D}_A)$	0.033	0.040

AUCs for individual experiments (i.e., before averaging the ROC curves) and compute their mean and standard deviation. However, since we are comparing over different experiments, variability in AUC measure includes not only the variance of different learning methods but also variability due to different experiments. Thus, a better comparison between different methods is obtained by comparing the (paired) differences between AUC measures of different methods in individual experiments (i.e., this is analogous to using the paired t-test instead of the standard t-test). We compare the difference between the proposed approximate method and the traditional Bayesian inference and between the proposed exact RJMCMC method and the traditional Bayesian inference. Table 2 summarizes these results by showing estimates of the mean and standard deviation. Mean (resp. standard deviation) is estimated using the sample average (resp. $1.483 \cdot \text{mad}$, where mad stands for the median absolute deviation from the median and the scaling term 1.483 makes the estimator approximately unbiased for normally distributed data).[4] In light of the estimated means and standard deviations, the performance difference between the approximate method and the traditional Bayesian inference is only marginal. The difference between the exact RJMCMC method and the traditional Bayesian inference, however, is more significant as the mean paired difference is about one standard deviation away from zero. Under Gaussian approximation, statistics shown in Table 2 can be used to compute significance values that assess the probability of observing this large differences in AUC scores given the null hypothesis that different methods indeed perform similarly (i.e., traditional hypothesis testing). Significance values (one-tailed) are $p = 0.347$, 0.177, 0.373, and 0.205 corresponding to the different cases in Table 2. (Meaning of these significance values is discussed more below.) Although performance differences are consistent as seen in Fig. 4, p-values do not satisfy the commonly used significance level of 0.05. Performance differences between different methods depend on the relative size of time series and steady state measurements and also on the underlying conditional probabilities θ. For example, as we will see shortly, by decreasing the length of the time series as well as the value of the underlying θ that is used to generate time series data makes even steady state inference to perform better than the standard time series inference.

In addition to the above simulations, we also ran other tests. First, in order to gain insight into behavior of RJMCMC methods relative to the standard MCMC, we applied the proposed exact (RJMCMC) inference method to time series data alone and compared the inference results to those of the standard Bayesian inference (MCMC). Inference results for

[4]Robust variance estimates are, on average, slightly smaller than the standard sample variance estimates. A robust estimator is less sensitive to deviations from a normal distribution and possible outliers due to poor convergence, which are still possible even after careful convergence diagnostics.

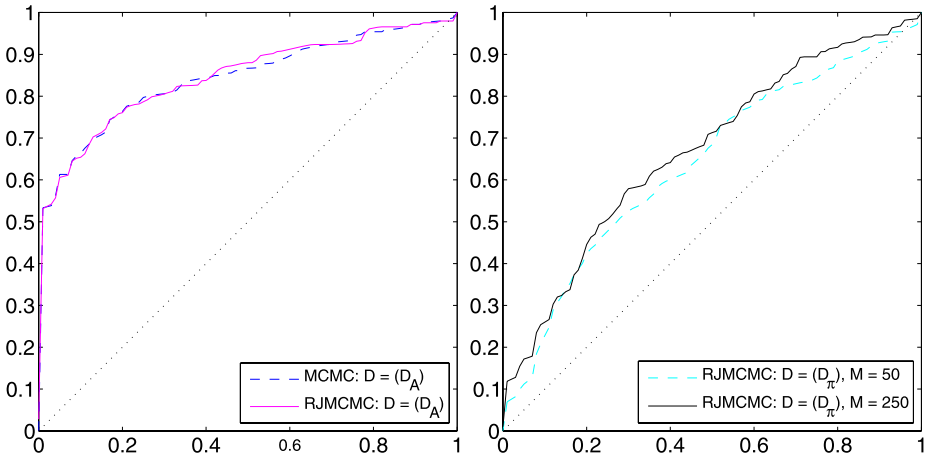

Fig. 5 The ROC curves for the two additional DBN structure learning tests. (**a**) the traditional method (*dashed blue*) and the RJMCMC method when applied to time series data alone (*solid magenta*). The curves are consistent with each other. (**b**) The RJMCMC method when applied to steady state data alone. Sample sizes of $M = 50$ (*dashed cyan*) and $M = 250$ (*solid black*) are considered

$n = 6$ networks in Fig. 5(a) show that the RJMCMC method when only time series data are available (solid magenta) produces practically identical results with the traditional method that is inherently limited to time series data only (dashed blue). This further confirms that the trans-dimensional MCMC method mixes and converges sufficiently well despite the additional complexity of sampling in the "product space" (see also details of the convergence assessment below). The same good agreement is also seen in terms of AUC measures, which are 0.8430 and 0.8453 for the MCMC and RJMCMC, respectively.

Second, to study the effect of steady state sample size, we also applied the RJMCMC method to a steady state data set having size $M = 250$. Figure 5(b) shows the results for the proposed RJMCMC method when applied to steady state data alone, $M = 50$ (dashed cyan) and $M = 250$ (solid black). For the larger steady state sample size we use slightly larger burn-in and sampling period lengths to ensure convergence, i.e., $L = 2.5 \times 10^6$ and $S = 2.5 \times 10^6$. The performance improves as the sample size increases from $M = 50$ to $M = 250$ although the improvement appears relatively small. The corresponding AUC measure improves from 0.6391 to 0.6688. The relatively small improvement again reflects the fact that steady state measurements contain less dynamic information than time series measurements. The above finding is also likely to reflect the non-unique nature of the network inference from steady state measurements. In other words, even infinite sample size, i.e., full knowledge of the underlying steady state distribution, does not in general guarantee that the corresponding ROC curve would be 'ideal.'

Next we consider a biologically more realistic simulation setting by decreasing the length of the time series as well as the value of θ that is used to generate time series data. This scenario is motivated by the problem of learning gene regulatory networks from gene expression time series and/or steady state protein level data. Because transcriptional regulation is largely carried out by proteins, gene expression measurements are considered to be noisier than protein level measurements (at least from gene regulatory network inference point of view). The amount of steady state protein level data is also typically higher than that of gene

expression time series. However, protein levels can currently be measured only for a handful of proteins simultaneously whereas gene expression measurements typically cover all the genes. With these aspects in mind, we now generated short ($T = 15$) and noisy ($\theta_{ijk} = 0.6$ or $\theta_{ijk} = 0.7$) time series while parameters of the steady state data were the same as before ($M = 50$ or $M = 250$ and $\theta_{ijk} = 0.9$). Structure of the underlying network is the same for both time series and steady state data and parameters are kept consistent in that if $\theta_{ijk} = 0.7$ (resp. $\theta_{ijk} = 0.3$) in time series then $\theta_{ijk} = 0.9$ (resp. $\theta_{ijk} = 0.1$) in the steady state. We again consider learning the structure of the underlying DBN using time series data, steady state data, and a combination of time series and steady state data. This simulation is again repeated for 20 randomly generated networks of size $n = 6$ and $n = 8$ as above.

Note that the above data set does not fit into our RJMCMC inference method directly because time series and steady state data are generated with different parameter values. We could modify our method (using a dependent parameter prior) such that the difference in parameter values for time series and steady state is fixed, say, 0.2 or 0.3. Such an approach can be somewhat unrealistic because that requires accurate knowledge of the underlying data generating mechanisms which might not be available in practical applications. Instead, we propose to use the approach where parameters are not coupled (discussed at the end of Sect. 4.6). That is, we apply the time series (standard MCMC) and steady state (RJMCMC) methods separately to $\mathcal{D}_A$ and $\mathcal{D}_\pi$, respectively. The posterior probability estimates are combined as $\hat{P}(E_{ij}|\mathcal{D}) \propto \hat{P}(E_{ij}|\mathcal{D}_A)\hat{P}(E_{ij}|\mathcal{D}_\pi)$, where E_{ij} again denotes the edge from node X_i to node X_j.

Figure 6 shows the ROC curves for different structure learning methods. Interestingly, when time series data are shorter and noisier, then even the RJMCMC estimation from steady state data alone (dash-dotted cyan) performs on average better than the state-of-the-art time series inference (dashed blue). Moreover, performance is improved even further when network structures are inferred from a combination of time series and steady state data (solid red). For the network size $n = 6$ we try two scenarios: $\theta_{ijk} = 0.7$ and $M = 50$, and $\theta_{ijk} = 0.7$ and $M = 250$. As above, larger steady state sample size improves the performance. For the network size $n = 8$ we try two scenarios $\theta_{ijk} = 0.6$ and $M = 50$, and $\theta_{ijk} = 0.7$ and $M = 50$, where the behavior is again as what one would expect: noisy and short time series provides less accurate inference results than a combination of noisy and short time series and higher quality steady state measurements. Similar significance value computation based on Gaussian approximation as above gives the following significance values 0.2849, 0.2138, 0.1602, and 0.1992 corresponding to the cases in Figs. 6(a)–(d). Significance levels are again only moderate.

To better understand the outcome of these significance tests, we also compared the performance of the standard state-of-the-art MCMC method to the random choice (i.e., the diagonal line in ROC curves). Using the noisy and short time series data sets from Figs. 6(a)–(d) results in similar significance values, i.e., 0.1992, 0.2946, and 0.1719. In other words, although performance of the standard MCMC method on noisy and short time series ($\theta_{ijk} = 0.6$ or $\theta_{ijk} = 0.7$ and $T = 15$) is not statistically significant (using the common level $\alpha = 0.05$ as a criterion), it consistently outperforms the random choice. Similar arguments apply for our proposed method when compared with previous methods.

6.2 Convergence assessment results

A critical step of MCMC-based stochastic estimation methods is convergence assessment. We used the diagnostics described in Sect. 4.8 and only accepted those chains that passed the two convergence tests. Figures 7(a) and (b) show illustrative convergence diagnostic plots

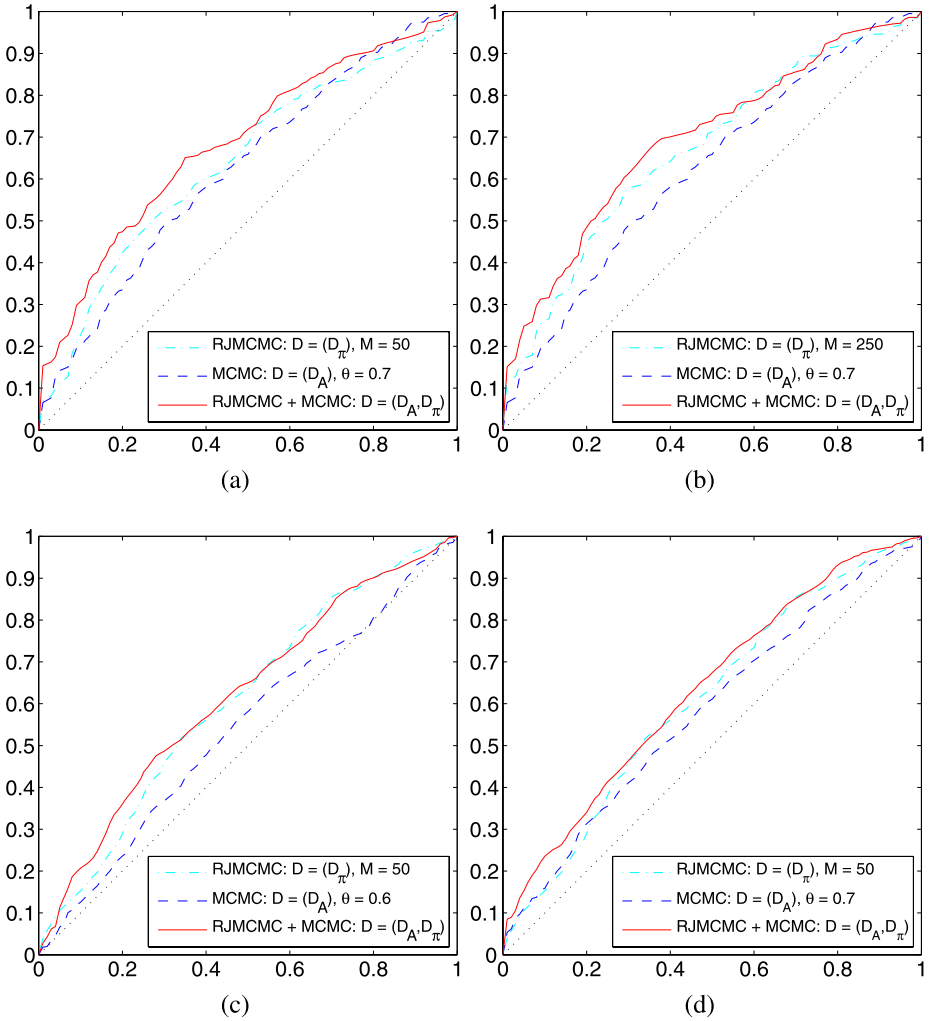

Fig. 6 The ROC curves for the biologically motivated structure learning simulations. The *first row*: the number of nodes is $n = 6$. The *second row*: the number of nodes is $n = 8$. The traditional method (*dashed blue*), the RJMCMC method when applied to steady state data alone (*dash-dotted cyan*), and a combination of time series and steady state data (*solid red*). (**a**) $\theta_{ijk} = 0.7$, $M = 50$, (**b**) $\theta_{ijk} = 0.7$, $M = 250$, (**c**) $\theta_{ijk} = 0.6$, $M = 50$, and (**d**) $\theta_{ijk} = 0.7$, $M = 50$

using the non-parametric method of Brooks et al. (2003) for cases $n = 6$ and $n = 8$, respectively. After the burn-in, we plot the significance value (p-value) of the convergence test for different values of the MCMC/RJMCMC iteration index ℓ in increments of the average sub-sampling parameter λ. Different methods are color-coded as in Fig. 4. As a threshold value we use $\alpha = 0.1$ that is shown with the dashed red line. In Fig. 7 the significance values of the test statistic are all close to one, indicating no evidence against the null hypothesis of homogeneity. That is commonly the case for these simulations and only a few MCMC runs are rejected. Note that the significance values in Fig. 7(b) are practically equal

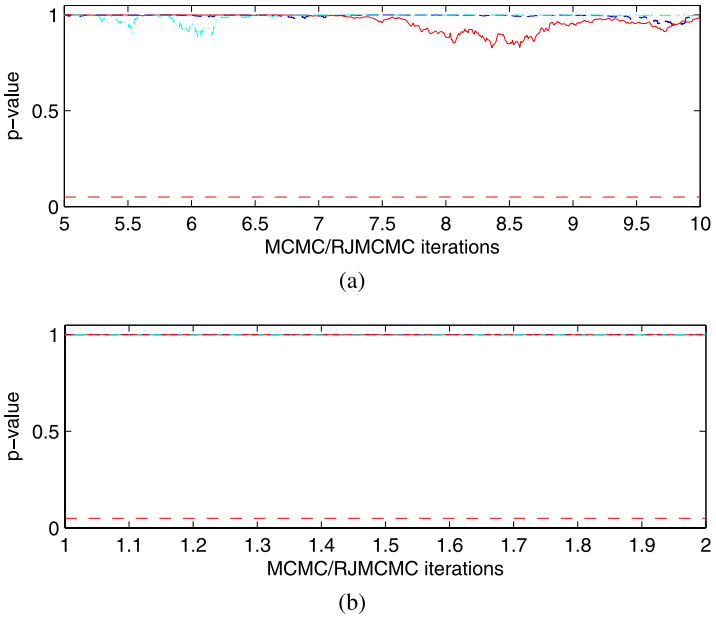

Fig. 7 The significance value (p-value) of the convergence test for different values of ℓ: (**a**) case $n = 6$, and (**b**) case $n = 8$. Different methods are color-coded as in Fig. 4. *Horizontal axis* shows (**a**) hundreds of thousands (**b**) millions of iterations

to one. This suggests that the increased number of (RJ)MCMC iterations for $n = 8$ networks ($L + S = 2 \times 10^6$ instead of $L + S = 10^6$) might not be necessary.

Figures 8(a) and (b) show the same convergence diagnostic plots for the two additional simulations. Although a larger steady state sample size makes the posterior distribution over network structures more "peaky," it also makes it more difficult for the RJMCMC to sample from the full search space. Consequently, the convergence is typically a bit more difficult to obtain for larger sample sizes and, therefore, the chain is run for a longer time ($L + S = 5 \times 10^6$ instead of $L + S = 10^6$). Figure 8(b) shows an illustrative plot when the size of the steady state sample is $M = 250$. It is also worth noting that, in general, the ease or difficulty of obtaining sufficiently converged chain pairs is sample data dependent. That is, for most of the sample data sets the first two chains pass the convergence diagnostics but there are also a few problematic ones for which additional chains are needed in order to obtain a chain pair that passes the tests. An illustrative diagnostic plot of a chain pair that is rejected as not having converged is shown in Fig. 11(a).

As an additional diagnostic, we also use a simple method that plots the estimated posterior probabilities of the network connections from two independent simulations (Husmeier 2005). Figures 9(a–d) and (e–h) show illustrative scatter plots for cases $n = 6$ and $n = 8$, respectively. Note that all the scatter plots are very close to the diagonal, indicating good convergence. The scatter plot in graphs (a) and (e) (and often (b) and (f) as well) show the least deviation from the diagonal line. This is expected since the estimation is based on MCMC rather than on RJMCMC and thus the additional complexity of sampling in the product space $\{\mathcal{M}, \theta\}$ is avoided. Graphs (a–b) and (e–f) suggest that even a shorter chain could suffice for the MCMC-based methods. Running the chains for the same number of steps, however, allows a straightforward and fair comparison of different methods. The

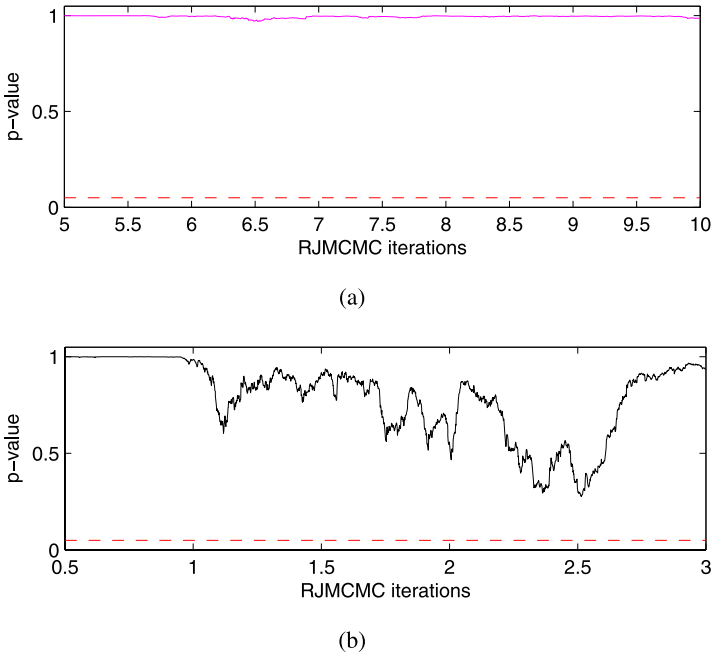

Fig. 8 The significance value (p-value) of the convergence test for the two additional tests. (**a**) the RJMCMC method when applied to time series data alone. (**b**) The RJMCMC method when applied to time series data alone, $M = 250$. *Horizontal axis* shows (**a**) hundreds of thousands (**b**) millions of iterations

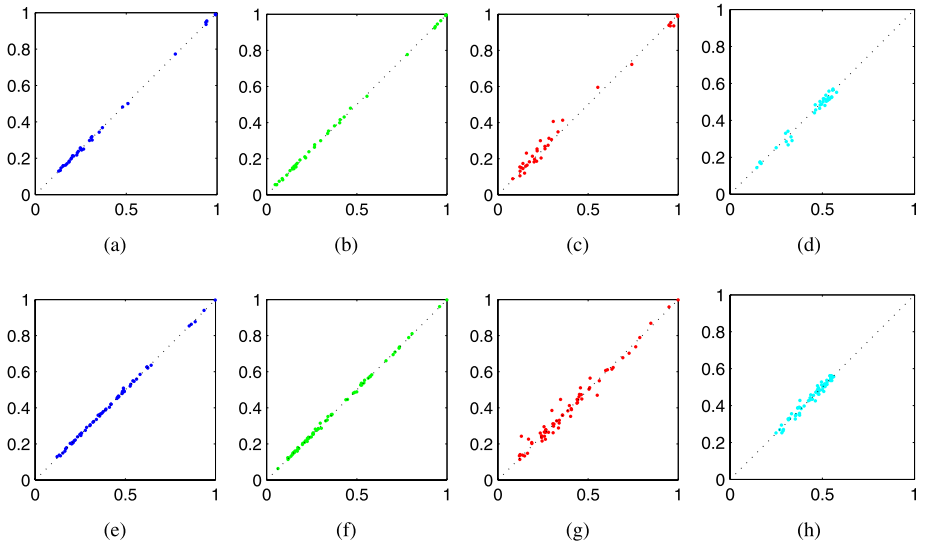

Fig. 9 Representative scatter plots of the estimated posterior probabilities of the network connections from two independent simulations: (**a–d**) case $n = 6$, and (**e–h**) case $n = 8$. Subplots (**a**) and (**e**) the traditional method, (**b**) and (**f**) the proposed approximate method, (**c**) and (**g**) the RJMCMC method when applied to both time series and steady state data, and (**d**) and (**h**) the proposed RJMCMC method when applied to steady state data alone

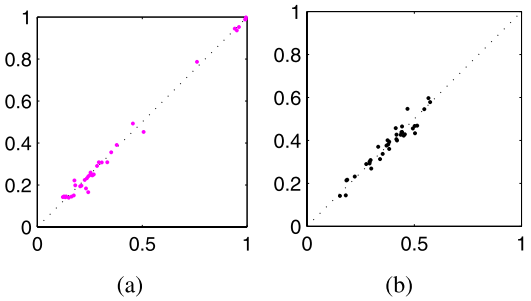

Fig. 10 Illustrative scatter plots of the estimated posterior probabilities of the network connections for the two additional simulations. (**a**) The RJMCMC method when applied to time series data alone, and (**b**) the proposed RJMCMC method when applied to steady state data alone, with $M = 250$

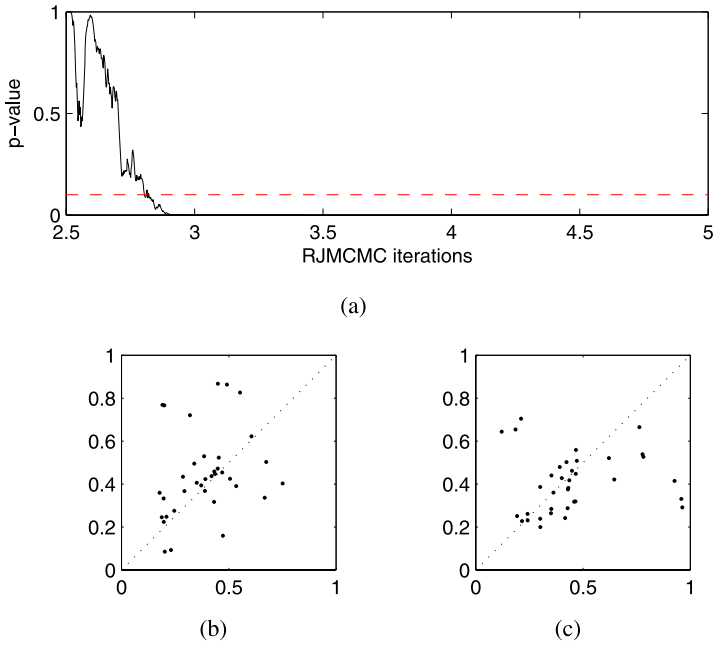

Fig. 11 Illustrative diagnostic plots of chain pairs that are rejected as not having converged. (**a**) The significance value (p-value) of the convergence test. (**b**) and (**c**) Scatter plots of the estimated posterior probabilities of the network connections. Graphs (**a**) and (**b**) correspond to the same chain pair

lower information content of the steady state measurements is also apparent in graphs (d) and (h), i.e., all the estimated posterior probabilities are below 0.6. Figures 10(a) and (b) show the same scatter plot convergence diagnostics for the two additional simulations. Illustrative scatter plots corresponding to chain pairs that are rejected as not having converged are shown in Figs. 11(b) and (c).

All of the above diagnostic tests show evidence for convergence. As a cautionary note for stochastic simulations, however, it is worth mentioning that the above tests are only necessary, not sufficient (Husmeier 2005), for convergence.

In order to gain additional insight into the different MCMC and RJMCMC chains, Figs. 12(a) and (b) show illustrative examples of the cumulative acceptance rates of MCMC/RJMCMC proposal moves for cases $n = 6$ and $n = 8$, respectively. Curves are again color-coded in the same way as in Figs. 4 and 5. Note that the cumulative acceptance rates

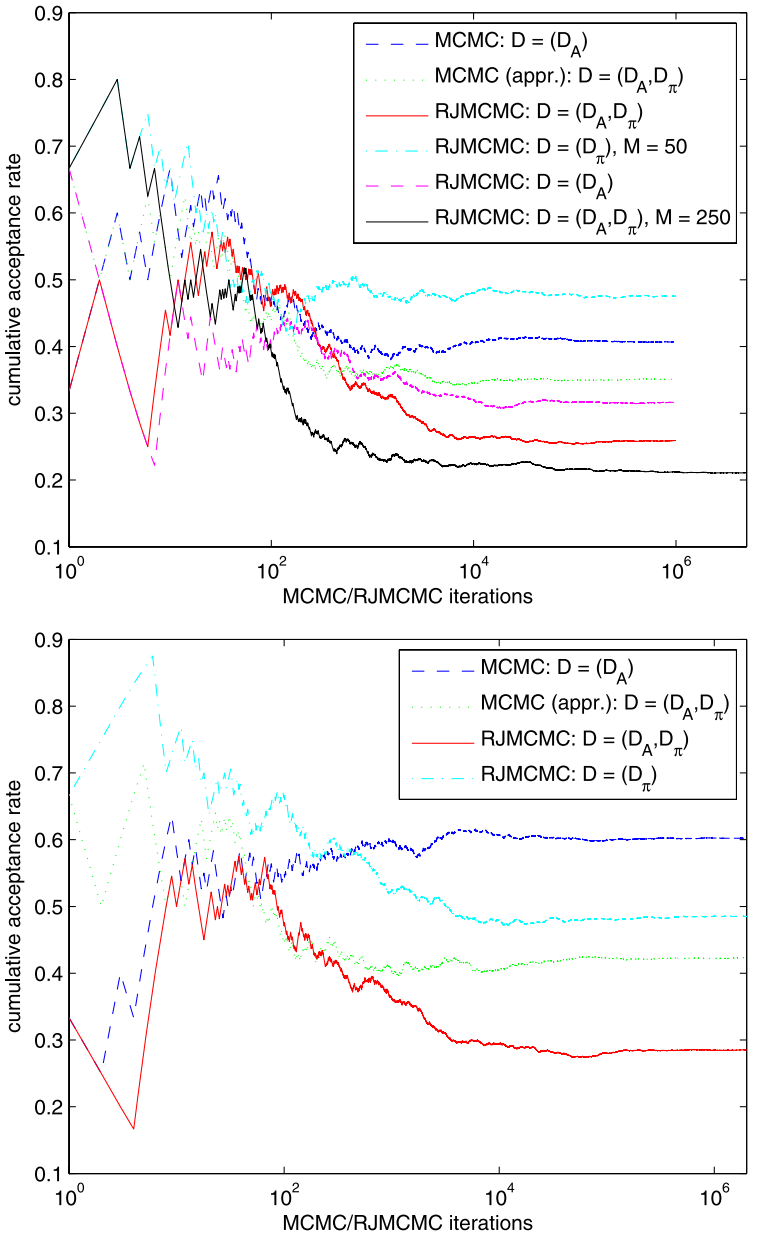

Fig. 12 Representative cumulative acceptance rates of MCMC/RJMCMC proposal moves for different methods. (**a**) Case $n = 6$ and (**b**) case $n = 8$

of the two additional simulations are included in Fig. 12(a). Also note that the first half of the samples are ignored as the burn-in period and only the last half are used in the estimation. A commonly used rule of thumb (Gelman et al. 1996) recommends acceptance rate of around 0.25 for 'high dimensional' models and around 0.5 for 'smaller dimensional' mod-

els, which are in concordance with the obtained acceptance rates. Recall that the RJMCMC methods also have the null move where only the parameters are updated.

7 Discussion and conclusions

The proposed exact inference method is implemented using RJMCMC which provides an extremely flexible framework for model selection problems. (Similar RJMCMC strategies have been discussed e.g. in (Dellaportas and Forster 1999; Giudici et al. 2000; Pournara 2004).) RJMCMC also allows various extensions to the methods described in this manuscript. For example, as the RJMCMC methods are not based on conjugate prior analysis, one can equally well consider other than the Dirichlet prior distributions. A more sophisticated approach can be constructed using, for example, hierarchical priors. It is worth noting that the posterior distribution of the conditional probabilities θ_{ijk} can be estimated as well. Furthermore, the proposed methodology can also be extended to handle hidden variables (e.g., more general state space models) and missing data. For that purpose, an approach that combines the proposed methods with, e.g., ideas from (Robert et al. 2000) for learning HMMs might be potentially valuable. An interesting future research problem will be to study modifications of the proposed methods to other, completely different, model classes as well. Also note that although the proposed inference methods are combined with MCMC and RJMCMC procedures, one can equally well adapt them to be used together with other search methods, such the greedy hill climbing.

The flexibility of the RJMCMC does not come without drawbacks. The "unlimited" freedom of choosing numerous proposal distributions makes their choice more demanding from the convergence point of view. Although RJMCMC, as well as MCMC, is guaranteed to converge under fairly mild conditions in the limit of infinite iterations, an unsophisticated choice of the proposal distributions can lead to situations where convergence is unreachable within a reasonable number of iterations (see, e.g., Robert and Casella 2005). This emphasizes the need for reliable convergence diagnostics. In order to further improve convergence and mixing of the chain, it could also be possible to use an adaptive MCMC scheme to adjust average cumulative acceptance rates. For example, some parameters of the proposal distribution(s), such as β and σ, could be adapted during the burn-in period so as to optimize convergence. If reliable prior information is available, it might also be beneficial to sample parameters from priors directly. A related problem is that the implicitly restarted Arnoldi method (as implemented in ARPACK/MATLAB) occasionally reported an error, indicating that the eigenvector (i.e., the steady state distribution) cannot be solved reliably. Such presumably numerical problems involved in solving for the eigenvector might be alleviated by a different choice of proposal distributions.

The current implementation of the proposed methods is limited to fairly small networks due to computational complexity involved in solving for the steady state distribution. For example, it can be too time consuming to apply the proposed methods (using the current implementation) to the cases $n > 12$ (depending on the values of r_i). Thus, an important future research direction will be to study more efficient ways of solving for the steady state distribution. That has recently attracted attention with regard to Google's PageRank application. A specific feature of the eigenvector problem in that application is that one knows the steady state distribution π of the current stochastic matrix A and the goal is to obtain the steady state distribution π' of another stochastic matrix A' that has been obtained from A by slightly changing some of its transition probabilities. Exactly the same problem is also met in the proposed network inference methods where minor changes are proposed to the

network, and thereby to the transition probabilities, at each iteration of (RJ)MCMC. The goal is then to develop a method for computing π' given π, A and A' that is computationally more efficient than a standard approach that solves π' directly from A'. Langville and Meyer (2006) have recently introduced an efficient iterative method to solve this problem, which is directly applicable to our problem as well.

Many of the efficient eigenvector solvers can also be parallelized and general purpose software packages are available for that purpose. Although parallelization does not change the asymptotic time complexity apart from (possible) linear time improvement that can make the range of applications just sufficiently wider. Parallelization might indeed prove useful in our case.

Alternatively, the inference methods only require the steady state probabilities for those states that are among the observed steady states $\mathcal{D}_\pi$. In other words, instead of solving the steady state probabilities for all $\prod_{i=1}^{n} r_i$ states one could do that only for (a maximum of) M states. More generally, the special structure of the state transition matrix A (each element of A is a product of n conditional probabilities, see e.g. (4)) can carry over to the steady state distribution as well and provide an efficient way of solving for the steady state probabilities analytically. Even if no analytical solution can be found it will be interesting to study approximations as well. That was the approach taken in (Ching et al. 2007) who developed an approximation method for solving the steady-state probability distribution of probabilistic Boolean networks (PBN). Although the method was developed for PBNs, the idea of Ching et al. was essentially to ignore small conditional probabilities from the computation of the state transition matrix and the corresponding steady state distribution. In addition, Ching et al. also developed an expected error bound for their approximation. Given the close relationship between DBNs and PBNs (Lähdesmäki et al. 2006), this method can be directly used in the context of DBNs as well. Assessment of possible advantages of the approximate method and error bound of Ching et al. (2007) in our application will be left for future studies.

In this article we introduced a novel, rigorous Bayesian method for learning the structure of DBNs from steady state measurements. The major advantage of the proposed method is that it is able to learn dynamic network models even if only non-time series measurements are available. Moreover, the proposed method can also be applied to a combination of time series and steady state data on which its performance is improved relative to the standard state-of-the-art Bayesian inference using time series data alone, although this improvement when measured in terms of the AUC score was not found to be statistically significant. Results reported in this manuscript confirm the expected fact that time series measurements contain more information about network dynamics than steady state measurements. However, steady state measurements do contain a substantial amount of information about network dynamics and, if properly used in the inference, their use can result in an improved learning algorithm.

Acknowledgements The authors wish to thank Antti Larjo for his careful reading and suggestions on the manuscript. This work was supported by grants R01 GM072855 and P50 GM076547 from NIH/NIGMS. We thank the anonymous reviewers for their valuable comments and suggestions.

References

Andrieu, C., Djurić, P. M., & Doucet, A. (2001). Model selection by MCMC computation. *Signal Processing*, *81*(1), 19–37.

Bernard, A., & Hartemink, A. (2005). Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data. In *Proceedings of pacific symposium on biocomputing (PSB 05)* (Vol. 10, pp. 459–470). Singapore: World Scientific.

Brooks, S. P., Giudici, P., & Philippe, A. (2003). Nonparametric convergence assessment for MCMC model selection. *Journal of Computational and Graphical Statistics*, *12*(1), 1–22.

Ching, W.-K., Zhang, S.-Q., Ng, M. K., & Akutsu, T. (2007). An approximation method for solving the steady-state probability distribution of probabilistic Boolean networks. *Bioinformatics*, *23*(12), 1511–1518.

Çinlar, E. (1997). *Introduction to stochastic processes* (1st ed.). Englewood Cliffs: Prentice Hall.

Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, *9*(4), 309–347.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., & Spiegelhalter, D. J. (1999). *Probabilistic networks and expert systems*. New York: Springer.

Dean, T., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, *5*(3), 142–150.

Dellaportas, P., & Forster, J. J. (1999). Markov chain Monte Carlo model determination for hierarchical and graphical log-linear models. *Biometrika*, *86*(3), 615–633.

Dobra, A., Hans, C., Jones, B., Nevins, J. R., Yao, G., & West, M. (2004). Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, *90*(1), 196–212.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874.

Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, *303*, 799–805.

Friedman, N., & Koller, D. (2003). Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, *50*, 95–125.

Friedman, N., Murphy, K., & Russell, S. (1998). Learning the structure of dynamic probabilistic networks. In *Proceedings of fourteenth conference on uncertainty in artificial intelligence (UAI)* (pp. 139–147). San Mateo: Morgan Kaufmann.

Geiger, D., & Heckerman, D. (1997). A characterization of the Dirichlet distribution through global and local parameter independence. *The Annals of Statistics*, *25*(3), 1344–1369.

Gelman, A., Roberts, G. O., & Gilks, W. R. (1996). Efficient Metropolis jumping rules. In J. M. Bernardo, J. O. Berger, A. P. Dawid, & A. F. M. Smith (Eds.), *Bayesian statistics* (Vol. 5, pp. 599–607). Oxford: Oxford University Press.

Giudici, P., Green, P. J., & Tarantola, C. (2000). *Efficient model determination for discrete graphical models* (Discussion paper No. 99-63). Available on-line at http://citeseer.ist.psu.edu/giudici00efficient.html.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, *82*(4), 711–732.

Hartemink, A., Gifford, D., Jaakkola, T., & Young, R. (2001). Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. In *Proceedings of pacific symposium on biocomputing (PSB 01)* (Vol. 6, pp. 422–433). Singapore: World Scientific.

Hartemink, A., Gifford, D., Jaakkola, T., & Young, R. (2002). Combining location and expression data for principled discovery of genetic regulatory network models. In *Proceedings of pacific symposium on biocomputing (PSB 02)* (Vol. 7, pp. 437–449). Singapore: World Scientific.

Heckerman, D. (1998). A tutorial on learning with Bayesian networks. In M. I. Jordan (Ed.), *Learning in graphical models* (pp. 301–354). Cambridge: MIT Press.

Heckerman, D., Geiger, D., & Chickering, D. (1995). Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, *20*(3), 197–243.

Husmeier, D. (2003). Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, *19*(17), 2271–2282.

Husmeier, D. (2005). Introduction to learning Bayesian networks from data. In D. Husmeier, R. Dybowski, & S. Roberts (Eds.), *Probabilistic modeling in bioinformatics and medical informatics* (pp. 17–57). Berlin: Springer.

Imoto, S., Kim, S., Goto, T., Miyano, S., Aburatani, S., & Tashiro, K. (2003). Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. *Journal of Bioinformatics and Computational Biology*, *1*(2), 231–252.

Lähdesmäki, H., Hautaniemi, S., Shmulevich, I., & Yli-Harja, O. (2006). Relationships between probabilistic Boolean networks and dynamic Bayesian networks as models of gene regulatory networks. *Signal Processing*, *86*(4), 814–834.

Langville, A. N., & Meyer, C. D. (2006). Updating Markov chains with an eye on Google's PageRank. *SIAM Journal on Matrix Analysis and Applications*, *27*(4), 968–987.

Lehoucq, R. B., Sorensen, D. C., & Yang, C. (1998). *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. Philadelphia: SIAM.

Madigan, D., & York, J. (1995). Bayesian graphical models for discrete data. *International Statistical Review*, *63*(2), 215–232.
Markowetz, F. (2007). *A bibliography on learning causal networks of gene interactions*. Available on-line at http://www.molgen.mpg.de/~markowet/docs/network-bib.pdf.
Murphy, K. P. (2001). The Bayes net toolbox for Matlab. *Computing Science and Statistics*, *33*, 1–20. Software is available on-line at http://bnt.sourceforge.net/.
Murphy, K. P. (2002). *Dynamic Bayesian networks: representation, inference and learning*. PhD thesis, University of California, Berkeley.
Nikovski, D. (1998). Learning stationary temporal probabilistic networks. In *Conference on automated learning and discovery*.
Pearl, J. (2000). *Causality: models, reasoning, and inference*. Cambridge: Cambridge University Press.
Pe'er, D., Regev, A., Elidan, G., & Friedman, N. (2001). Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, *17*(Suppl. 1), 215S–224S.
Pournara, I. (2004). *Reconstructing gene regulatory networks by passive and active Bayesian learning*. PhD thesis, Birkbeck College, University of London.
Pournara, I., & Wernisch, L. (2004). Reconstruction of gene networks using Bayesian learning and manipulation experiments. *Bioinformatics*, *20*(17), 2934–2942.
Richardson, S., & Green, P. (1997). On Bayesian analysis of mixtures with an unknown number of components, with discussion. *Journal of the Royal Statistical Society: Series B*, *59*(4), 731–792.
Rissanen, J. J. (1978). Modeling by shortest data description. *Automatica*, *14*, 465–471.
Robert, C. P., & Casella, G. (2005). *Monte Carlo statistical methods* (2nd ed.), Berlin: Springer.
Robert, C. P., Rydén, T., & Titterington, D. M. (2000). Bayesian inference in hidden Markov models through the reversible jump Markov chain Monte Carlo. *Journal of the Royal Statistical Society: Series B*, *62*(1), 57–75.
Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., & Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, *308*(5721), 523–529.
Schäfer, J., & Strimmer, K. (2005). An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, *21*(5), 754–764.
Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*(2), 461–464.
Stewart, W. J. (1994). *Introduction to the numerical solution of Markov chains* (1st ed.). Princeton: Princeton University Press.
Werhli, A. V., Grzegorczyk, M., & Husmeier, D. (2006). Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical Gaussian models and Bayesian networks. *Bioinformatics*, *22*(20), 2523–2531.
Wille, A., Zimmermann, P., Vranová, E., Fürholz, A., Laule, O., & Bleuler, S. (2004). Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana. *Genome Biology*, *5*(11), R92.