Bayesian Analysis (2015) **10**, Number 1, pp. 109–138

Bayesian Structure Learning in Sparse Gaussian Graphical Models

A. Mohammadi* and E. C. Wit†

Abstract. Decoding complex relationships among large numbers of variables with relatively few observations is one of the crucial issues in science. One approach to this problem is Gaussian graphical modeling, which describes conditional independence of variables through the presence or absence of edges in the underlying graph. In this paper, we introduce a novel and efficient Bayesian framework for Gaussian graphical model determination which is a trans-dimensional Markov Chain Monte Carlo (MCMC) approach based on a continuous-time birth-death process. We cover the theory and computational details of the method. It is easy to implement and computationally feasible for high-dimensional graphs. We show our method outperforms alternative Bayesian approaches in terms of convergence, mixing in the graph space and computing time. Unlike frequentist approaches, it gives a principled and, in practice, sensible approach for structure learning. We illustrate the efficiency of the method on a broad range of simulated data. We then apply the method on large-scale real applications from human and mammary gland gene expression studies to show its empirical usefulness. In addition, we implemented the method in the R package `BDgraph` which is freely available at http://CRAN.R-project.org/package=BDgraph.

Keywords: Bayesian model selection, Sparse Gaussian graphical models, Non-decomposable graphs, Birth-death process, Markov chain Monte Carlo, G-Wishart.

1 Introduction

Statistical inference of complex relationships among large numbers of variables with a relatively small number of observations appears in many circumstances. Biologists want to recover the underlying genomic network between thousands of genes, based on at most a few hundred observations. In market basket analysis analysts try to find relationships between only a small number of purchases of individual customers (Giudici and Castelo, 2003). One approach to these tasks is probabilistic graphical modeling (Lauritzen, 1996), which is based on the conditional independencies between variables. Graphical models offer fundamental tools to describe the underlying conditional correlation structure. They have recently gained in popularity in both statistics and machine learning with the rise of high-dimensional data (Jones et al., 2005; Dobra et al., 2011a; Meinshausen and Bühlmann, 2006; Wang and Li, 2012; Friedman et al., 2008; Ravikumar et al., 2010; Zhao and Yu, 2006; Wang, 2012, 2014). For the purpose of performing structure learning, Bayesian approaches provide a straightforward tool, explicitly incorporating underlying graph uncertainty.

*Dept. of Statistics, University of Groningen, Groningen, Netherlands, a.mohammadi@rug.nl
†Dept. of Statistics, University of Groningen, Groningen, Netherlands, e.c.wit@rug.nl

 DOI: 10.1214/14-BA889

In this paper, we focus on Bayesian structure learning in Gaussian graphical models for both decomposable and non-decomposable cases. Gaussian graphical determination can be viewed as a covariance selection problem (Dempster, 1972), where the non-zero entries in the off-diagonal of the precision matrix correspond to the edges in the graph. For a p-dimensional variable there are in total $2^{p(p-1)/2}$ possible conditional independence graphs. Even with a moderate number of variables, the model space is astronomical in size. The methodological problem as the dimension grows includes searching over the graph space to identify high posterior regions. High-dimensional regimes, such as genetic networks, have hundreds of nodes, resulting in over 10^{100} possible graphs. This motivates us to construct an efficient search algorithm which explores the graph space to distinguish important edges from irrelevant ones and detect the underlying graph with high accuracy. One solution is the trans-dimensional MCMC methodology (Green, 2003).

In the trans-dimensional MCMC methodology, the MCMC algorithm explores the model space to identify high posterior probability models and estimate the parameters simultaneously. A special case is the reversible-jump MCMC (RJMCMC) approach, proposed by Green (1995). This method constructs an ergodic discrete-time Markov chain whose stationary distribution is taken to be the joint posterior distribution of the model and the parameters. The process transits among models using an acceptance probability, which guarantees convergence to the target posterior distribution. If this probability is high, the process efficiently explores the model space. However, for the high-dimensional regime this is not always efficient. Giudici and Green (1999) extended this method for the decomposable Gaussian graphical models. Dobra et al. (2011a) developed it based on the Cholesky decomposition of the precision matrix. Lenkoski (2013), Wang and Li (2012), and Cheng et al. (2012) developed an RJMCMC algorithm, which combined the exchange algorithm (Murray et al., 2012) and the double Metropolis-Hastings algorithm (Liang, 2010) to avoid the intractable normalizing constant calculation.

An alternative trans-dimensional MCMC methodology is the birth-death MCMC (BDMCMC) approach, which is based on a continuous time Markov process. In this method, the time between jumps to a larger dimension (birth) or a smaller one (death) is taken to be a random variable with a specific rate. The choice of birth and death rates determines the birth-death process, and is made in such a way that the stationary distribution is precisely the posterior distribution of interest. Contrary to the RJMCMC approach, moves between models are always accepted, which makes the BDMCMC approach extremely efficient. In the context of finite mixture distributions with variable dimension this method has been used (Stephens, 2000), following earlier proposals by Ripley (1977) and Geyer and Møller (1994).

The main contribution of this paper is to introduce a novel Bayesian framework for Gaussian graphical model determination and design a BDMCMC algorithm to perform both structure learning (graph estimation) and parameter learning (parameters estimation). In our BDMCMC method, we add or remove edges via birth or death events. The birth and death events are modeled as independent Poisson processes. Therefore, the time between two successive birth or death events has an exponential distribution. The

birth and death events occur in continuous time and the relative rates at which they occur determine the stationary distribution of the process. The relationships between these rates and the stationary distribution is formalized in Section 3 (Theorem 3.1).

The outline of this paper is as follows. In Section 2, we introduce the notation and preliminary background material such as suitable prior distributions for the graph and precision matrix. In Section 3, we propose our Bayesian framework and design our BDMCMC algorithm. In addition, this section contains the specific implementation of our method, including an efficient way for computing the birth and death rates of our algorithm and a direct sampler algorithm from G-Wishart distribution for the precision matrix. In Section 4, we show the performance of the proposed method in several comprehensive simulation studies and large-scale real-world examples from human gene expression data and a mouse mammary gland microarray experiment.

2 Bayesian Gaussian graphical models

We introduce some notation and the structure of undirected Gaussian graphical models; for a comprehensive introduction see Lauritzen (1996). Let $G = (V, E)$ be an undirected graph, where $V = \{1, 2, ..., p\}$ is the set of nodes and $E \subset V \times V$ is the set of existing edges. Let

$$\mathcal{W} = \{(i, j) \mid i, j \in V,\ i < j\},$$

and $\overline{E} = \mathcal{W} \backslash E$ denotes the set of non-existing edges. We define a zero mean Gaussian graphical model with respect to the graph G as

$$\mathcal{M}_G = \left\{\mathcal{N}_p(0, \Sigma) \mid K = \Sigma^{-1} \in \mathbb{P}_G\right\},$$

where $\mathbb{P}_G$ denotes the space of $p \times p$ positive definite matrices with entries (i, j) equal to zero whenever $(i, j) \in \overline{E}$. Let $\mathbf{x} = (\mathbf{x}^1, ..., \mathbf{x}^n)$ be an independent and identically distributed sample of size n from model $\mathcal{M}_G$. Then, the likelihood is

$$P(\mathbf{x}|K, G) \propto |K|^{n/2} \exp\left\{-\frac{1}{2}\text{tr}(KS)\right\}, \tag{1}$$

where $S = \mathbf{x}'\mathbf{x}$.

The joint posterior distribution is given as

$$P(G, K|\mathbf{x}) \propto P(\mathbf{x}|G, K)P(K|G)P(G). \tag{2}$$

For the prior distribution of the graph there are many options, of which we propose two. In the absence of any prior beliefs related to the graph structure, one case is a discrete uniform distribution over the graph space $\mathcal{G}$,

$$P(G) = \frac{1}{|\mathcal{G}|}, \quad \text{for each} \quad G \in \mathcal{G}.$$

Alternatively, we propose a truncated Poisson distribution on the graph size $|E|$ with parameter γ,

$$p(G) \propto \frac{\gamma^{|E|}}{|E|!}, \quad \text{for each} \quad G = (V, E) \in \mathcal{G}.$$

Other choices of priors for the graph structure involve modelling the joint state of the edges as a multivariate discrete distribution (Carvalho et al., 2009) and (Scutari, 2013), encouraging sparse graphs (Jones et al., 2005) or having multiple testing correction properties (Scott and Berger, 2006).

For the prior distribution of the precision matrix, we use the G-Wishart (Roverato, 2002; Letac and Massam, 2007), which is attractive since it represents the conjugate prior for normally distributed data. It places no probability mass on zero entries of the precision matrix. A zero-constrained random matrix $K \in \mathbb{P}_G$ has the G-Wishart distribution $W_G(b, D)$, if

$$P(K|G) = \frac{1}{I_G(b, D)} |K|^{(b-2)/2} \exp\left\{-\frac{1}{2}\text{tr}(DK)\right\},$$

where $b > 2$ is the degree of freedom, D is a symmetric positive definite matrix, and $I_G(b, D)$ is the normalizing constant,

$$I_G(b, D) = \int_{\mathbb{P}_G} |K|^{(b-2)/2} \exp\left\{-\frac{1}{2}\text{tr}(DK)\right\} dK.$$

When G is complete the G-Wishart distribution $W_G(b, D)$ reduces to the Wishart distribution $W_p(b, D)$, hence, its normalizing constant has an explicit form (Muirhead, 1982). If G is decomposable, we can explicitly calculate $I_G(b, D)$ (Roverato, 2002). For non-decomposable graphs, however, $I_G(b, D)$ does not have an explicit form; we can numerically approximate $I_G(b, D)$ by the Monte Carlo method (Atay-Kayis and Massam, 2005) or Laplace approximation (Lenkoski and Dobra, 2011).

The G-Wishart prior is conjugate to the likelihood (1), hence, conditional on graph G and observed data $\mathbf{x}$, the posterior distribution of K is

$$P(K|\mathbf{x}, G) = \frac{1}{I_G(b^*, D^*)} |K|^{(b^*-2)/2} \exp\left\{-\frac{1}{2}\text{tr}(D^*K)\right\},$$

where $b^* = b + n$ and $D^* = D + S$, that is, $W_G(b^*, D^*)$.

Other choices of priors for the precision matrix are considered on a class of shrinkage priors (Wang and Pillai, 2013) using the graphical lasso approach (Wang, 2012, 2014). They place constant priors for the nonzero entries of the precision matrix and no probability mass on zero entries.

In the following section, we describe an efficient trans-dimensional MCMC sampler scheme for our joint posterior distribution (2).

3 The birth-death MCMC method

Here, we determine a continuous time birth-death Markov process particularly for Gaussian graphical model selection. The process explores over the graph space by adding or removing an edge in a birth or death event. The birth and death rates of edges occur in continuous time with the rates determined by the stationary distribution of the process.

Suppose the birth-death process at time t is at state (G, K) in which $G = (V, E)$ with precision matrix $K \in \mathbb{P}_G$. Let $\Omega = \cup_{\substack{G \in \mathcal{G} \\ K \in \mathbb{P}_G}} (G, K)$ where $\mathcal{G}$ denotes the set of all possible graphs. We consider the following continuous time birth-death Markov process on Ω:

Death: Each edge $e \in E$ dies independently of the others as a Poisson process with a rate $\delta_e(K)$. Thus, the overall death rate is $\delta(K) = \sum_{e \in E} \delta_e(K)$. If the death of an edge $e = (i, j) \in E$ occurs, then the process jumps to a new state (G^{-e}, K^{-e}) in which $G^{-e} = (V, E \setminus \{e\})$, and $K^{-e} \in \mathbb{P}_{G^{-e}}$. We assume K^{-e} is equal to matrix K except for the entries in positions $\{(i, j), (j, i), (j, j)\}$. Note we can distinguish i from j, since by our definition of an edge $i < j$.

Birth: A new edge $e \in \overline{E}$ is born independently of the others as a Poisson process with a rate $\beta_e(K)$. Thus, the overall birth rate is $\beta(K) = \sum_{e \in \overline{E}} \beta_e(K)$. If the birth of an edge $e = (i, j) \in \overline{E}$ occurs, then the process jumps to a new state (G^{+e}, K^{+e}) in which $G^{+e} = (V, E \cup \{e\})$, and $K^{+e} \in \mathbb{P}_{G^{+e}}$. We assume K^{+e} is equal to matrix K except for the entries in positions $\{(i, j), (j, i), (j, j)\}$.

The birth and death processes are independent Poisson processes. Thus, the time between two successive events is exponentially distributed, with mean $1/(\beta(K)+\delta(K))$. Therefore, the probability of a next birth/death event is

$$P(\text{birth for edge } e) = \frac{\beta_e(K)}{\beta(K) + \delta(K)}, \qquad \text{for each} \quad e \in \overline{E}, \tag{3}$$

$$P(\text{death for edge } e) = \frac{\delta_e(K)}{\beta(K) + \delta(K)}, \qquad \text{for each} \quad e \in E. \tag{4}$$

The following theorem gives a sufficient condition for which the stationary distribution of our birth-death process is precisely the joint posterior distribution of the graph and precision matrix.

Theorem 3.1. *The above birth-death process has stationary distribution $P(K, G|\mathbf{x})$, if for each $e \in \mathcal{W}$*

$$\delta_e(K) P(G, K \setminus (k_{ij}, k_{jj})|\mathbf{x}) = \beta_e(K^{-e}) P(G^{-e}, K^{-e} \setminus k_{jj}|\mathbf{x}). \tag{5}$$

Proof. Our proof is based on the theory derived by Preston (1976, Section 7 and 8). Preston proposed a special birth-death process, in which the birth and death rates

are functions of the state. The process evolves by two types of jumps: a *birth* is defined as the appearance of a single individual, whereas a *death* is the removal of a single individual. This process converges to a unique stationary distribution, if the balance conditions hold (Preston, 1976, Theorem 7.1). We construct our method in such a way that the stationary distribution equals the joint posterior distribution of the graph and the precision matrix. See the appendix for a detailed proof.

3.1 Proposed BDMCMC algorithm

Our proposed BDMCMC algorithm is based on a specific choice of birth and death rates that satisfies Theorem 3.1. Suppose we consider the birth and death rates as

$$\beta_e(K) = \frac{P(G^{+e}, K^{+e} \setminus (k_{ij}, k_{jj})|\mathbf{x})}{P(G, K \setminus k_{jj}|\mathbf{x})}, \quad \text{for each} \quad e \in \overline{E}, \tag{6}$$

$$\delta_e(K) = \frac{P(G^{-e}, K^{-e} \setminus k_{jj}|\mathbf{x})}{P(G, K \setminus (k_{ij}, k_{jj})|\mathbf{x})}, \quad \text{for each} \quad e \in E. \tag{7}$$

Based on the above rates, we determine our BDMCMC algorithm as below.

Algorithm 3.1. BDMCMC algorithm. Given a graph $G = (V, E)$ with a precision matrix K, iterate the following steps:

Step 1. Birth and death process

- **1.1.** Calculate the birth rates by (6) and $\beta(K) = \sum_{e \in \overline{E}} \beta_e(K)$,
- **1.2.** Calculate the death rates by (7) and $\delta(K) = \sum_{e \in E} \delta_e(K)$,
- **1.3.** Calculate the waiting time by $w(K) = 1/(\beta(K) + \delta(K))$,
- **1.4.** Simulate the type of jump (birth or death) by (3) and (4).

Step 2. According to the type of jump, sample from the new precision matrix.

The main computational parts of our BDMCMC algorithm are computing the birth and death rates (steps 1.1 and 1.2) and sampling from the posterior distribution of the precision matrix (step 2). In Section 3.2, we illustrate how to calculate the birth and death rates. In Section 3.3, we explain a direct sampling algorithm from the G-Wishart distribution for sampling from the posterior distribution of the precision matrix.

In our continuous time BDMCMC algorithm we sample in each step of jumping to the new state (e.g. $\{t_1, t_2, t_3, ...\}$ in Figure 1). For inference, we put the weight on each state to effectively compute the sample mean as a Rao-Blackwellized estimator (Cappé et al., 2003, subsection 2.5); See e.g. (13). The weights are equal to the length of the

waiting time in each state (e.g. $\{w_1, w_2, w_3, ...\}$ in Figure 1). Based on these waiting times, we estimate the posterior distribution of the graphs, which are the proportion to the total waiting times of each graph (see Figure 1 in the right and Figure 3). For more detail about sampling from continuous time Markov processes see Cappé et al. (2003, subsection 2.5).

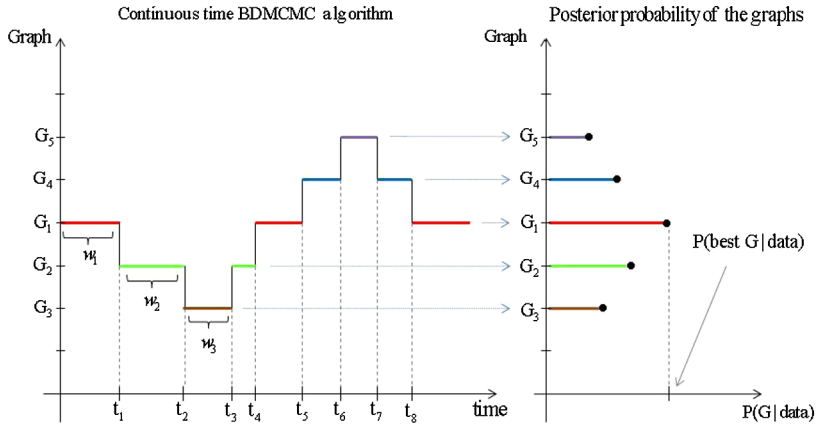


Figure 1: (Left) Continuous time BDMCMC algorithm where $\{t_1, t_2, t_3, ...\}$ are jumping times and $\{w_1, w_2, w_3, ...\}$ are waiting times. (Right) Posterior probability estimation of the graphs based on the proportions of their waiting times.

3.2 Step1: Computing the birth and death rates

In step 1 of our BDMCMC algorithm, the main task is calculating the birth and death rates (steps 1.1 and 1.2); Other steps are straightforward. Here, we illustrate how to calculate the death rates. The birth rates are calculated in a similar manner, since both birth and death rates (6) and (7) are the ratio of the conditional posterior densities.

For each $e = (i, j) \in E$, the numerator of the death rate is

$$P(G^{-e}, K^{-e} \setminus k_{jj}|\mathbf{x}) = \frac{P(G^{-e}, K^{-e}|\mathbf{x})}{P(k_{jj}|K^{-e} \setminus k_{jj}, G^{-e}, \mathbf{x})}.$$

The full conditional posterior for k_{jj} is (see Roverato 2002, Lemma 1)

$$k_{jj} - c \mid K^{-e} \setminus k_{jj}, G^{-e}, \mathbf{x} \sim W(b^*, D^*_{jj}),$$

where $c = K_{j,V\setminus j}(K_{V\setminus j,V\setminus j})^{-1}K_{V\setminus j,j}$. Following Wang and Li (2012) and some simplification, we have

$$P(G^{-e}, K^{-e}\setminus k_{jj}|\mathbf{x}) = \frac{P(G)}{P(\mathbf{x})}\frac{I(b^*, D^*_{jj})}{I_{G^{-e}}(b, D)}|K^0_{V\setminus j,V\setminus j}|^{(b^*-2)/2}\exp\left\{-\frac{1}{2}\mathrm{tr}(K^0D^*)\right\}, \quad (8)$$

where $I(b^*, D^*_{jj})$ is the normalizing constant of a G-Wishart distribution for $p = 1$ and $K^0 = K$ except for an entry 0 in the positions (i, j) and (j, i), and an entry c in the position (j, j).

For the denominator of the death rate we have

$$P(G, K \setminus (k_{ij}, k_{jj})|\mathbf{x}) = \frac{P(G, K|\mathbf{x})}{P((k_{ij}, k_{jj})|K \setminus (k_{ij}, k_{jj}), G, \mathbf{x})},$$

in which we need the full conditional distribution of (k_{ij}, k_{jj}). We can obtain the conditional posterior of (k_{ii}, k_{ij}, k_{jj}) and by further conditioning on k_{ii} and using the proposition in the appendix, we can evaluate the full conditional distribution of (k_{ij}, k_{jj}). Following Wang and Li (2012) and some simplification, we have

$$P(K \setminus (k_{ij}, k_{jj}), G|\mathbf{x}) = \frac{P(G)}{P(\mathbf{x})} \frac{J(b^*, D^*_{ee}, K)}{I_G(b, D)} |K^1_{V\setminus e, V\setminus e}|^{(b^*-2)/2} \exp\left\{-\frac{1}{2}\mathrm{tr}(K^1 D^*)\right\}, \tag{9}$$

where

$$J(b^*, D^*_{ee}, K) = (\frac{2\pi}{D^*_{jj}})^{\frac{1}{2}} I(b^*, D^*_{jj})(k_{ii} - k^1_{ii})^{\frac{b^*-2}{2}} \exp\left\{-\frac{1}{2}(D^*_{ii} - \frac{D^{*2}_{ij}}{D^*_{jj}})(k_{ii} - k^1_{ii})\right\},$$

and $K^1 = K$ except for the entries $K_{e,V\setminus e}(K_{V\setminus e,V\setminus e})^{-1}K_{V\setminus e,e}$ in the positions corresponding to $e = (i, j)$.

By plugging (8) and (9) into the death rates (7), we have

$$\delta_e(K) = \frac{P(G^{-e})}{P(G)} \frac{I_G(b, D)}{I_{G^{-e}}(b, D)} H(K, D^*, e), \tag{10}$$

in which

$$\begin{aligned} H(K, D^*, e) &= (\frac{D^*_{jj}}{2\pi(k_{ii} - k^1_{ii})})^{\frac{1}{2}} \\ &\times \exp\left\{-\frac{1}{2}\left[\mathrm{tr}(D^*(K^0 - K^1)) - (D^*_{ii} - \frac{D^{*2}_{ij}}{D^*_{jj}})(k_{ii} - k^1_{ii})\right]\right\}. \end{aligned} \tag{11}$$

For computing the above death rates, we require the prior normalizing constants which is the main computational part. Calculation time for the remaining elements is extremely fast.

Coping with evaluation of prior normalizing constants

Murray et al. (2012) proved that the exchange algorithm based on exact sampling is a powerful tool for general MCMC algorithms in which their likelihoods have additional parameter-dependent normalization terms, such as the posterior over parameters of an undirected graphical model. Wang and Li (2012) and Lenkoski (2013) illustrate how to

use the concept behind the exchange algorithm to circumvent intractable normalizing constants as in (10). With the existence of a direct sampler of G-Wishart, Lenkoski (2013) used a modification of the exchange algorithm to cope with the ratio of prior normalizing constants.

Suppose that (G, K) is the current state of our algorithm and we would like to calculate the death rates (10), first we sample $\tilde{K}$ according to $W_G(b, D)$ via an exact sampler, Algorithm 3.2 below. Then, we replace the death rates with

$$\delta_e(K) \quad = \quad \frac{P(G^{-e})}{P(G)} \frac{H(K, D^*, e)}{H(\tilde{K}, D, e)}, \tag{12}$$

in which the intractable prior normalizing constants have been replaced by an evaluation of H (given in (11)) in the prior, evaluated at $\tilde{K}$; For theoretical justifications of this procedure, see Murray et al. (2012) and Liang (2010).

3.3 Step 2: Direct sampler from precision matrix

Lenkoski (2013) developed an exact sampler method for the precision matrix, which borrows ideas from Hastie et al. (2009). The algorithm is as follows.

Algorithm 3.2. Direct sampler from precision matrix Lenkoski (2013). Given a graph $G = (V, E)$ with precision matrix K and $\Sigma = K^{-1}$:

Step 1. Set $\Omega = \Sigma$.

Step 2. Repeat for $i = 1, ..., p$, until convergence:

2.1 Let $N_i \subset V$ be the set of neighbors of node i in graph G. Form Ω_{N_i} and $\Sigma_{N_i,i}$ and solve

$$\hat{\beta}_i^* = \Omega_{N_i}^{-1} \Sigma_{N_i,i},$$

2.2 Form $\hat{\beta}_i \in R^{p-1}$ by padding the elements of $\hat{\beta}_i^*$ to the appropriate locations and zeroes in those locations not connected to i in graph G,

2.3 Update $\Omega_{i,-i}$ and $\Omega_{-i,i}$ with $\Omega_{-i,-i}\hat{\beta}_i$.

Step 3. Return $K = \Omega^{-1}$.

Throughout, we use the direct sampler algorithm for sampling from the precision matrix K.

4 Statistical performance

Here we present the results for three comprehensive simulation studies and two applications to real data sets. In Section 4.1, we show that our method outperforms alternative

Bayesian approaches in terms of convergence, mixing in the graph space and computing time; Moreover, the model selection properties compare favorably with frequentist alternatives. In Section 4.2, we illustrate our method on a large-scale real data set related to the human gene expression data. In Section 4.3, we demonstrate the extension of our method to graphical models which involve time series data. It shows how graphs can be useful in modeling real-world problems such as gene expression time course data. We performed all computations with the R package `BDgraph`, (Mohammadi and Wit, 2013).

4.1 Simulation study

Graph with 6 nodes

We illustrate the performance of our methodology and compare with two alternative Bayesian methods on a concrete small toy simulation example which comes from Wang and Li (2012). We consider a data generating mechanism with $p = 6$ within

$$\mathcal{M}_G = \left\{ \mathcal{N}_6(0, \Sigma) \mid K = \Sigma^{-1} \in \mathbb{P}_G \right\},$$

in which the precision matrix is

$$K = \begin{bmatrix} 1 & 0.5 & 0 & 0 & 0 & 0.4 \\ & 1 & 0.5 & 0 & 0 & 0 \\ & & 1 & 0.5 & 0 & 0 \\ & & & 1 & 0.5 & 0 \\ & & & & 1 & 0.5 \\ & & & & & 1 \end{bmatrix}.$$

Just like Wang and Li (2012) we let $S = nK^{-1}$ where $n = 18$, which represents 18 samples from the true model $\mathcal{M}_G$. As a non-informative prior, we take a uniform distribution for the graph and a G-Wishart $W_G(3, I_6)$ for the precision matrix.

To evaluate the performance of our BDMCMC algorithm, we run our BDMCMC algorithm with $60,000$ iterations and $30,000$ as a burn-in. All the computations for this example were carried out on an Intel(R) Core(TM) i5 CPU 2.67GHz processor.

We calculate the posterior pairwise edge inclusion probabilities based on the Rao-Blackwellization (Cappé et al., 2003, subsection 2.5) as

$$\hat{p}_e = \frac{\sum_{t=1}^{N} I(e \in G^{(t)}) w(K^{(t)})}{\sum_{t=1}^{N} w(K^{(t)})}, \quad \text{for each} \quad e \in \mathcal{W}, \tag{13}$$

where N is the number of iterations, $I(e \in G^{(t)})$ is an indicator function, such that $I(e \in G^{(t)}) = 1$ if $e \in G^{(t)}$ and zero otherwise, and $w(K^{(t)})$ is the waiting time in the

graph $G^{(t)}$ with the precision matrix $K^{(t)}$; see Figure 1. The posterior pairwise edge inclusion probabilities for all the edges $e = (i, j) \in \mathcal{W}$ are

$$\hat{p}_e = \begin{bmatrix} 0 & 0.98 & 0.05 & 0.02 & 0.03 & 0.92 \\ & 0 & 0.99 & 0.04 & 0.01 & 0.04 \\ & & 0 & 0.99 & 0.04 & 0.02 \\ & & & 0 & 0.99 & 0.06 \\ & & & & 0 & 0.98 \\ & & & & & 0 \end{bmatrix}.$$

The posterior mean of the precision matrix is

$$\hat{K} = \begin{bmatrix} 1.16 & 0.58 & -0.01 & 0.00 & -0.01 & 0.44 \\ & 1.18 & 0.58 & -0.01 & 0.00 & -0.01 \\ & & 1.18 & 0.58 & -0.01 & 0.00 \\ & & & 1.18 & 0.58 & -0.01 \\ & & & & 1.17 & 0.57 \\ & & & & & 1.16 \end{bmatrix}.$$

We compare the performance of our BDMCMC algorithm with two recently proposed trans-dimensional MCMC approaches. One is the algorithm proposed by Lenkoski (2013) which we call "Lenkoski". The other is an algorithm proposed by Wang and Li (2012) which we call "WL". The R code for the WL approach is available at `http://r-forge.r-project.org/projects/bgraph/`.

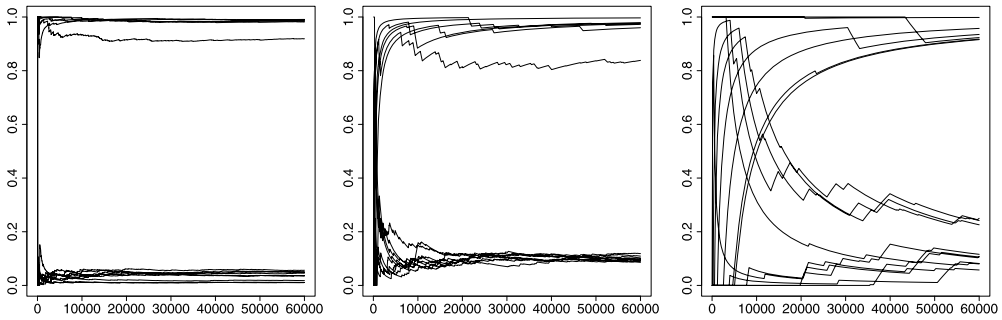
Compared to other Bayesian approaches, our BDMCMC algorithm is highly efficient due to its fast convergence speed. One useful test of convergence is given by the plot of the cumulative occupancy fraction for all possible edges, shown in Figure 2. Figure 2(a) shows that our BDMCMC algorithm converges after approximately $10,000$ iterations. Figure 2(b) and 2(c) show that the Lenkoski algorithm converges after approximately $30,000$, whereas the WL algorithm still does not converge after $60,000$ iterations.

Figure 3 reports the estimated posterior distribution of the graphs for BDMCMC, WL, and Linkoski algorithm, respectively. Figure 3(a) indicates that our algorithm visited around 450 different graphs and the estimated posterior distribution of the true graph is 0.66, which is the graph with the highest posterior probability. Figure 3(b) shows that the Lenkoski algorithm visited around 400 different graphs and the estimated posterior distribution of the true graph is 0.40. Figure 3(c) shows that the WL algorithm visited only 23 different graphs and the estimated posterior distribution of the true graph is 0.35.

To assess the performance of the graph structure, we compute the posterior probability of the true graph, and the calibration error (CE) measure, defined as follows

$$CE = \sum_{e \in \mathcal{W}} |\hat{p}_e - I(e \in G_{true})|, \tag{14}$$

where, for each $e \in \mathcal{W}$, $\hat{p}_e$ is the posterior pairwise edge inclusion probability in (13) and G_{true} is the true graph. The CE is positive with a minimum at 0 and smaller is better.



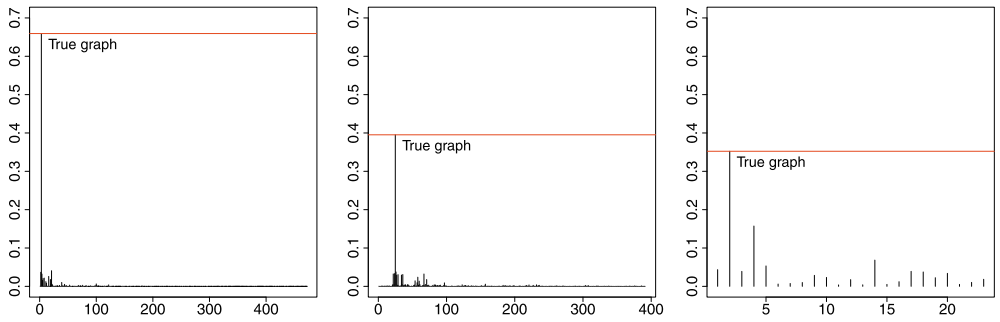
(a) BDMCMC (b) Lenkoski (c) WL

Figure 2: Plot of the cumulative occupancy fractions of all possible edges to check convergence in simulation example 4.1; BDMCMC algorithm in 2(a), Lenkoski algorithm(Lenkoski, 2013) in 2(b), and WL algorithm in Wang and Li (2012) 2(c).

Table 1 reports comparisons of our method with two other Bayesian approaches (WL and Lenkoski), reporting the mean values and standard errors in parentheses. We repeat the entire simulation 50 times. The first and second columns show the performance of the algorithms. Our algorithm performs better due to its faster convergence feature. The third column shows the acceptance probability (α) which is the probability of moving to a new graphical model. The fourth column shows that our algorithm is slower than the Lenkoski algorithm and faster than the WL approach. It can be argued that to make a fair comparison our method takes $60,000$ samples in 25 minutes while e.g. Lenkoski algorithm in 14 minutes takes only $60,000 \times 0.058 = 3,480$ efficient samples. For fair comparison we performed all simulations in R. However, our package `BDgraph` efficiently implements the algorithm with C++ code linked to R. For $60,000$ iterations, our C++ code takes only 17 seconds instead of 25 minutes in R, which means around 90 times faster than the R code. It makes our algorithm computationally feasible for high-dimensional graphs.

	P(true G | data)	CE	α	CPU time (min)
BDMCMC	0.66 (0.00)	0.47 (0.01)	1	25 (0.14)
Lenkoski	0.36 (0.02)	1.17 (0.08)	0.058 (0.001)	14 (0.13)
WL	0.33 (0.12)	1.25 (0.46)	0.003 (0.0003)	37 (0.64)

Table 1: Summary of performance measures in simulation example 4.1 for BDMCMC approach, Lenkoski (Lenkoski, 2013), and WL (Wang and Li, 2012). The table presents the average posterior probability of the true graph, the average calibration error (CE) which is defined in (14), the average acceptance probability (α), and the average computing time in minutes, with 50 replications and standard deviations in parentheses.



(a) BDMCMC (b) Lenkoski (c) WL

Figure 3: Plot of the estimated posterior probability of graphs in simulation example 4.1; BDMCMC algorithm in 3(a), Lenkoski algorithm (Lenkoski, 2013) in 3(b), and WL algorithm (Wang and Li, 2012) in 3(c).

Extensive comparison with Bayesian methods

We perform here a comprehensive simulation with respect to different graph structures to evaluate the performance of our Bayesian method and compare it with two recently proposed trans-dimensional MCMC algorithms; WL (Wang and Li, 2012) and Lenkoski (Lenkoski, 2013). Corresponding to different sparsity patterns, we consider 7 different kinds of synthetic graphical models:

1. *Circle:* A graph with $k_{ii} = 1$, $k_{i,i-1} = k_{i-1,i} = 0.5$, and $k_{1p} = k_{p1} = 0.4$, and $k_{ij} = 0$ otherwise.
2. *Star:* A graph in which every node is connected to the first node, with $k_{ii} = 1$, $k_{1i} = k_{i1} = 0.1$, and $k_{ij} = 0$ otherwise.
3. *AR(1):* A graph with $\sigma_{ij} = 0.7^{|i-j|}$.
4. *AR(2):* A graph with $k_{ii} = 1$, $k_{i,i-1} = k_{i-1,i} = 0.5$, and $k_{i,i-2} = k_{i-2,i} = 0.25$, and $k_{ij} = 0$ otherwise.
5. *Random:* A graph in which the edge set E is randomly generated from independent Bernoulli distributions with probability $2/(p-1)$ and the corresponding precision matrix is generated from $K \sim W_G(3, I_p)$.
6. *Cluster:* A graph in which the number of clusters is $\max\{2, [p/20]\}$. Each cluster has the same structure as a random graph. The corresponding precision matrix is generated from $K \sim W_G(3, I_p)$.
7. *Scale-free:* A graph which is generated by using the B-A algorithm (Albert and Barabási, 2002). The resulting graph has $p-1$ edges. The corresponding precision matrix is generated from $K \sim W_G(3, I_p)$.

For each graphical model, we consider four different scenarios: (1) dimension $p = 10$ and sample size $n = 30$, (2) $p = 10$ and $n = 100$, (3) $p = 50$ and $n = 100$, (4) $p = 50$ and $n = 500$.

For each generated sample, we fit our Bayesian method and two other Bayesian approaches (WL and Lenkoski) with a uniform prior for the graph and the G-Wishart prior $W_G(3, I_p)$ for the precision matrix. We run those three algorithms with the same starting points with $60,000$ iterations and $30,000$ as a burn in. Computation for this example was performed in parallel on 235 batch nodes with 12 cores and 24 GB of memory, running Linux.

To assess the performance of the graph structure, we compute the calibration error (CE) measure defined in (14) and the F_1-score measure (Baldi et al., 2000; Powers, 2011) which is defined as follows

$$F_1\text{-score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}, \tag{15}$$

where TP, FP, and FN are the number of true positives, false positives, and false negatives, respectively. The F_1-score lies between 0 and 1, where 1 stands for perfect identification and 0 for bad identification.

Table 2 reports comparisons of our method with two other Bayesian approaches, where we repeat the experiments 50 times and report the average F_1-score and CE with their standard errors in parentheses. Our method performs well overall as its F_1-score and its CE are the best in most of the cases, mainly because of its fast convergence rate. Both our method and the Lenkoski approach perform better compared to the WL approach. The main reason is that the WL approach uses a double Metropolis-Hastings (based on a block Gibbs sampler), which is an approximation of the exchange algorithm. On the other hand, both our method and the Lenkoski approach use the exchange algorithm based on exact sampling from the precision matrix. As we expected, the Lenkoski approach converges slower compared to our method. The reason seems to be the dependence of the Lenkoski approach on the choice of the tuning parameter, σ_g^2 (Lenkoski, 2013, step 3 in algorithm p. 124). In our simulation, we found that the convergence rate (as well acceptance probability) of the Lenkoski algorithm depends on the choice of σ_g^2. Here we choose $\sigma_g^2 = 0.1$ as a default. From a theoretical point of view, both our BDMCMC and the Lenkoski algorithms converge to the true posterior distribution, if we run them a sufficient amount of time. Thus, the results from this table just indicate how quickly the algorithms converge.

	F_1-score			CE		
	BDMCMC	Lenkoski	WL	BDMCMC	Lenkoski	WL
p=10 & n=30						
circle	**0.95** (0.00)	0.93 (0.01)	0.24 (0.01)	**2.5** (1.6)	4.9 (2.1)	15.8 (5)
star	0.15 (0.02)	**0.17** (0.02)	0.16 (0.01)	**11.3** (2.1)	14 (1.4)	13.6 (3.4)
AR1	**0.90** (0.01)	0.70 (0.01)	0.34 (0.02)	**4.4** (2.2)	9.7 (2.4)	12.5 (8.8)
AR2	0.56 (0.01)	**0.59** (0.02)	0.36 (0.01)	**11.5** (3.5)	12.8 (3.1)	16.3 (6.2)
random	**0.57** (0.03)	0.50 (0.01)	0.34 (0.01)	**11.4** (8.0)	15.3 (6.3)	14.1 (14.0)
cluster	**0.61** (0.02)	0.49 (0.01)	0.33 (0.01)	**10.3** (9.4)	14.3 (7.3)	13.5 (9.8)
scale-free	**0.53** (0.03)	0.45 (0.02)	0.31 (0.02)	**11.8** (8.8)	15.6 (6.9)	13.3 (6.5)
p=10 & n=100						
circle	**0.99** (0.00)	0.98 (0.00)	0.26 (0.01)	**1.0** (0.4)	2.2 (0.5)	15.6 (6.7)
star	0.21 (0.02)	0.18 (0.02)	**0.25** (0.02)	**9.3** (1.6)	11.4 (1.3)	11.4 (3.4)
AR1	**0.98** (0.00)	0.95 (0.00)	0.34 (0.01)	**1.5** (0.4)	5.2 (0.5)	13.0 (5.7)
AR2	0.89 (0.01)	**0.90** (0.01)	0.47 (0.01)	**4.1** (3.7)	5.6 (2.7)	14.0 (7.5)
random	**0.76** (0.01)	0.65 (0.02)	0.35 (0.01)	**7.0** (5.6)	10.7 (6.3)	13.8 (10.5)
cluster	**0.74** (0.02)	0.67 (0.02)	0.37 (0.02)	**6.4** (7.2)	9.9 (7.8)	12.4 (9.4)
scale-free	**0.69** (0.02)	0.56 (0.02)	0.33 (0.02)	**7.9** (8.0)	11.6 (7.0)	13.0 (7.8)
p=50 & n=100						
circle	**0.99** (0.01)	0.55 (0.10)	0.00 (0.00)	**2.5**(0.9)	75.3 (7.2)	50 (0.0)
star	**0.17** (0.04)	0.09 (0.04)	0.00 (0.00)	68.8 (4.4)	166.7 (4.5)	**49** (0.0)
AR1	**0.86** (0.04)	0.33 (0.09)	0.00 (0.00)	**19.0** (4.5)	159.1 (5.4)	49 (0.0)
AR2	**0.86** (0.04)	0.49 (0.17)	0.00 (0.00)	**28.6** (5.7)	117.5 (5.4)	97 (0.0)
random	**0.51** (0.09)	0.21 (0.07)	0.00 (0.00)	73.2 (18.5)	250.6 (36.7)	**49.2** (5.6)
cluster	**0.55** (0.11)	0.18 (0.07)	0.00 (0.00)	72.8 (18.2)	246.0 (44.4)	**47.8** (8.4)
scale-free	**0.49** (0.11)	0.19 (0.07)	0.00 (0.00)	72.4 (22.5)	243.1 (47.8)	**49** (0.0)
p=50 & n=500						
circle	**1.00** (0.01)	0.72 (0.09)	0.00 (0.00)	**1.7** (0.6)	55.8 (5.4)	50 (0.0)
star	**0.65** (0.05)	0.35 (0.05)	0.00 (0.00)	**31.7** (4.5)	92.3 (3.6)	49 (0.0)
AR1	**0.94** (0.02)	0.54 (0.07)	0.00 (0.00)	**7.2** (1.9)	84.9 (4.0)	49 (0.0)
AR2	**0.98** (0.01)	0.78 (0.11)	0.00 (0.00)	**4.8** (1.8)	61.7 (4.4)	97 (0.0)
random	**0.73** (0.09)	0.34 (0.10)	0.00 (0.00)	**34.3** (11.2)	149.3 (28.8)	50.7 (7.0)
cluster	**0.74** (0.09)	0.32 (0.13)	0.00 (0.00)	**32.2** (10.6)	142.2 (27.3)	48.5 (5.9)
scale-free	**0.73** (0.10)	0.33 (0.08)	0.00 (0.00)	**35.3** (13.6)	151.7 (26.9)	49 (0.0)

Table 2: Summary of performance measures in simulation example 4.1 for BDMCMC approach, Lenkoski (Lenkoski, 2013), and WL (Wang and Li, 2012). The table presents the F_1-score, which is defined in (15) and CE, which is defined in (14), for different models with 50 replications and standard deviations in parentheses. The F_1-score reaches its best score at 1 and its worst at 0. The CE is positive valued for which 0 is minimum and smaller is better. The best models for both F_1-score and CE are boldfaced.

Table 3 reports the average running time and acceptance probability (α) with their standard errors in parentheses across all 7 graphs with their 50 replications. It shows that our method compared to the Lenkoski approach is slower. The reason is that our method scans through all possible edges for calculating the birth/death rates, which is computationally expensive. On the other hand, in the Lenkoski algorithm, a new graph is selected by randomly choosing one edge which is computationally fast but not efficient. The table shows that the acceptance probability (α) for both WL and Lenkoski is small especially for the WL approach. Note the α here is the probability

that the algorithm moves to a new graphical model and it is not related to the double Metropolis-Hastings algorithm. The α in the WL approach is extremely small and it should be the cause of the approximation which has been used for the ratio of prior normalizing constants. As Murray et al. (2012) pointed out these kinds of algorithms can suffer high rejection rates. For the Lenkoski approach the α is relatively small, but compared with the WL method is much better. As in the Lenkoski approach, a new graph is proposed by randomly choosing one edge, yielding a relatively small acceptance probability.

		BDMCMC	Lenkoski	WL
p = 10	α	1	0.114 (0.001)	8.8e-06 (4.6e-11)
	Time	97 (628)	40 (225)	380 (11361)
p = 50	α	1	0.089 (0.045)	0.0000 (0.0000)
	Time	5408 (1694)	1193 (1000)	9650 (1925)

Table 3: Comparison of our BDMCMC algorithm with the WL approach (Wang and Li, 2012) and Lenkoski approach (Lenkoski, 2013). It presents the average computing time in minutes and the average probability of acceptance (α) with their standard deviations in parentheses.

Comparison with frequentist methods

We also compare the performance of our Bayesian method with two popular frequentist methods, the graphical lasso (glasso) (Friedman et al., 2008) and Meinshausen-Buhlmann graph estimation (mb) (Meinshausen and Bühlmann, 2006). We consider the same 7 graphical models with the same scenarios in the previous example.

For each generated sample, we fit our Bayesian method with a uniform prior for the graph and the G-Wishart prior $W_G(3, I_p)$ for the precision matrix. To fit the glasso and mb methods, however, we must specify a regularization parameter λ that controls the sparsity of the graph. The choice of λ is critical since different λ's may lead to different graphs. We consider the glasso method with three different regularization parameter selection approaches, which are the stability approach to regularization selection (stars) (Liu et al., 2010), rotation information criterion (ric) (Zhao et al., 2012), and the extended Bayesian information criterion (ebic) (Foygel and Drton, 2010). Similarly, we consider the mb method with two regularization parameter selection approaches, namely stars and the ric. We repeat all the experiments 50 times.

Table 4 provides comparisons of all approaches, where we report the averaged F_1-score with their standard errors in parentheses. Our Bayesian approach performs well as its F_1-score typically out performs all frequentist methods, except in the unlikely scenario of a high number of observations where it roughly equals the performance of the mb method with stars criterion. All the other approaches appear to perform well in some cases, and fail in other cases. For instance, when $p = 50$, the mb method with ric is the best for the AR(1) graph and the worst for the circle graph.

		glasso			mb	
	BDMCMC	stars	ric	ebic	stars	ric
p=10 & n=30						
circle	**0.95** (0.00)	0.00 (0.00)	0.01 (0.01)	**0.48** (0.00)	0.42 (0.01)	0.01 (0.01)
star	**0.15** (0.02)	0.01 (0.00)	**0.15** (0.02)	0.00 (0.00)	0.01 (0.02)	0.14 (0.02)
AR1	**0.90** (0.01)	0.20 (0.13)	0.61 (0.01)	0.17 (0.07)	0.46 (0.01)	**0.83** (0.01)
AR2	**0.56** (0.01)	0.09 (0.02)	**0.19** (0.02)	0.00 (0.00)	0.07 (0.02)	0.19 (0.02)
random	**0.57** (0.03)	0.36 (0.06)	0.48 (0.02)	0.08 (0.04)	0.45 (0.03)	**0.53** (0.03)
cluster	**0.61** (0.02)	0.45 (0.05)	0.54 (0.02)	0.07 (0.04)	0.50 (0.02)	**0.54** (0.02)
scale-free	**0.53** (0.03)	0.30 (0.05)	0.4 (0.02)	0.06 (0.02)	0.36 (0.03)	**0.46** (0.03)
p=10 & n=100						
circle	**0.99** (0.00)	0.00 (0.00)	0.50 (0.08)	0.45 (0.00)	**0.89** (0.08)	0.81 (0.09)
star	0.21 (0.02)	0.08 (0.02)	**0.29** (0.03)	0.01 (0.00)	0.07 (0.03)	**0.29** (0.03)
AR1	**0.98** (0.00)	0.90 (0.01)	0.57 (0.00)	0.56 (0.00)	**0.94** (0.00)	0.85 (0.00)
AR2	**0.89** (0.01)	0.34 (0.06)	0.63 (0.00)	0.08 (0.05)	0.41 (0.01)	**0.64** (0.01)
random	**0.76** (0.01)	0.61 (0.02)	0.57 (0.01)	0.45 (0.07)	**0.68** (0.02)	0.61 (0.02)
cluster	**0.74** (0.02)	0.66 (0.03)	0.59 (0.02)	0.53 (0.07)	**0.68** (0.03)	0.61 (0.03)
scale-free	**0.69** (0.02)	0.56 (0.02)	0.48 (0.008)	0.34 (0.07)	**0.63** (0.02)	0.52 (0.02)
p=50 & n=100						
circle	**0.99** (0.01)	0.28 (0.05)	0.00 (0.00)	**0.28** (0.01)	0.00 (0.00)	0.00 (0.00)
star	**0.17** (0.04)	0.14 (0.06)	0.06 (0.05)	0.00 (0.00)	**0.15** (0.04)	0.05 (0.041)
AR1	**0.86** (0.04)	0.56 (0.04)	0.59 (0.03)	0.49 (0.05)	0.82 (0.02)	**0.98** (0.02)
AR2	**0.86** (0.04)	0.59 (0.02)	0.02 (0.02)	0.00 (0.00)	**0.66** (0.02)	0.02 (0.02)
random	0.51 (0.09)	**0.52** (0.10)	0.40 (0.16)	0.04 (0.13)	**0.61** (0.21)	0.49 (0.21)
cluster	**0.55** (0.11)	0.54 (0.06)	0.42 (0.18)	0.13 (0.24)	**0.64** (0.22)	0.50 (0.22)
scale-free	**0.49** (0.11)	0.48 (0.10)	0.32 (0.18)	0.02 (0.09)	**0.60** (0.23)	0.40 (0.23)
p=50 & n=500						
circle	**1.00** (0.01)	**0.27** (0.05)	0.00 (0.00)	0.25 (0.01)	0.00 (0.00)	0.00 (0.00)
star	**0.65** (0.05)	0.29 (0.12)	0.60 (0.07)	0.01 (0.02)	0.31 (0.07)	**0.60** (0.07)
AR1	0.94 (0.02)	0.57 (0.02)	0.54 (0.02)	0.44 (0.02)	**0.97** (0.02)	**0.98** (0.01)
AR2	**0.98** (0.01)	0.69 (0.03)	0.64 (0.01)	0.66 (0.04)	**0.89** (0.02)	0.69 (0.02)
random	**0.73** (0.09)	0.62 (0.12)	0.46 (0.15)	0.56 (0.13)	**0.82** (0.24)	0.61 (0.24)
cluster	**0.74** (0.09)	0.65 (0.10)	0.51 (0.17)	0.58 (0.10)	**0.82** (0.23)	0.64 (0.23)
scale-free	**0.73** (0.10)	0.57 (0.14)	0.41 (0.15)	0.47 (0.15)	**0.82** (0.24)	0.62 (0.24)

Table 4: Summary of performance measures in simulation example 4.1 for BDMCMC approach, glasso (Friedman et al., 2008) with 3 criteria and mb (Meinshausen and Bühlmann, 2006) method with 2 criteria. The table reports F_1-score, which is defined in (15), for different models with 50 replications and standard deviations are in parentheses. The F_1-score reaches its best score at 1 and its worst at 0. The two top models are boldfaced.

To assess the performance of the precision matrix estimation, we use the Kullback-Leibler divergence (Kullback and Leibler, 1951) which is given as follows

$$\mathrm{KL} = \frac{1}{2}\left[\mathrm{tr}\left(K_{\mathrm{true}}^{-1}\hat{K}\right) - p - \log\left(\frac{|\hat{K}|}{|K_{\mathrm{true}}|}\right)\right], \tag{16}$$

where K_{true} is the true precision matrix and $\hat{K}$ is the estimate of the precision matrix.

Table 5 provides a comparison of all methods, where we report the averaged KL

with their standard errors in parentheses. Based on KL, the overall performance of our Bayesian approach is good as its KL is the best in all scenarios except one.

		glasso		
	BDMCMC	stars	ric	ebic
p=10 & n=30				
circle	**0.73** (0.12)	15.84 (0.03)	- -	10.34 (1.33)
star	0.57 (0.08)	0.31 (0.00)	**0.22** (0.00)	0.33 (0.01)
AR1	**0.70** (0.10)	3.63 (0.07)	1.59 (0.06)	2.77 (2.34)
AR2	**1.22** (0.07)	1.27 (0.00)	1.26 (0.00)	1.28 (0.00)
random	**0.67** (0.08)	8.32 (305)	- -	12.44 (1637)
cluster	**0.61** (0.06)	4.90 (2.37)	3.74 (3.23)	5.72 (7.35)
scale-free	**0.65** (0.07)	5.83 (12.35)	- -	6.59 (26.62)
p=10 & n=100				
circle	**0.14** (0.00)	15.95 (0.01)	- -	9.60 (0.56)
star	0.13 (0.00)	0.15 (0.00)	**0.10** (0.00)	0.17 (0.00)
AR1	**0.12** (0.00)	2.88 (0.16)	0.81 (0.01)	0.37 (0.00)
AR2	**0.28** (0.01)	1.24 (0.01)	1.14 (0.00)	1.25 (0.02)
random	**0.16** (0.00)	4.47 (1.09)	3.30 (0.76)	3.92 (2.55)
cluster	**0.13** (0.00)	4.46 (12.62)	3.62 (8.17)	4.47 (30.31)
scale-free	**0.16** (0.00)	4.14 (1.27)	3.01 (0.70)	3.68 (1.94)
p=50 & n=100				
circle	**0.67** (0.13)	117.12 (32.12)	- -	115.88 (5.88)
star	1.75 (0.21)	**1.05** (0.08)	1.27 (0.07)	1.49 (0.16)
AR1	**1.17** (0.23)	8 (1.10)	8.92 (0.50)	6.20 (0.93)
AR2	**1.97** (0.33)	6.56 (0.19)	7.29 (0.06)	7.27 (0.09)
random	**2.01** (0.42)	20.83 (6.44)	- -	30.07 (12.54)
cluster	**1.94** (0.42)	19.82 (3.75)	- -	26.47 (5.08)
scale-free	**1.96** (0.45)	20.91 (6.71)	- -	28.98 (8.40)
p=50 & n=500				
circle	**0.11** (0.01)	111.75 (30.56)	- -	111.32 (3.26)
star	**0.34** (0.04)	0.78 (0.04)	0.63 (0.03)	0.96 (0.06)
AR1	**0.15** (0.02)	4.87 (0.45)	3.51 (0.23)	1.75 (0.08)
AR2	**0.19** (0.03)	5.50 (0.2)	6.42 (0.13)	4.01 (0.10)
random	**0.26** (0.06)	18.89 (5.58)	17.14 (6.59)	17.80 (8.79)
cluster	**0.24** (0.05)	21.09 (21.51)	- -	21.34 (28.62)
scale-free	**0.25** (0.07)	19.86 (7.69)	- -	19.61 (16.65)

Table 5: Summary of performance measures in simulation example 4.1 for BDMCMC approach and glasso (Friedman et al., 2008) with 3 criteria. The table reports the KL measure, which is defined in (16), for different models with 50 replications and standard deviations are in parentheses. The KL is positive valued for which 0 is minimum and smaller is better. The best models are boldfaced.

4.2 Application to human gene expression data

We apply our proposed method to analyze the large-scale human gene expression data which was originally described by Bhadra and Mallick (2013); Chen et al. (2008), and Stranger et al. (2007). The data are collected by Stranger et al. (2007) using Illumina's Sentrix Human-6 Expression BeadChips to measure gene expression in B-lymphocyte

cells from Utah (CEU) individuals of Northern and Western European ancestry. They consider 60 unrelated individuals whose genotypes are available from the Sanger Institute website (ftp://ftp.sanger.ac.uk/pub/genevar). The genotype is coded as 0, 1, and 2 for rare homozygous, heterozygous and homozygous common alleles. Here the focus is on the 3125 Single Nucleotide Polymorphisms (SNPs) that have been found in the 5' UTR (untranslated region) of mRNA (messenger RNA) with a minor allele frequency ≥ 0.1. There were four replicates for each individual. Since the UTR has been subject to investigation previously, it should have an important role in the regulation of the gene expression. The raw data were background corrected and then quantile normalized across replicates of a single individual and then median normalized across all individuals. We chose the 100 most variable probes among the 47,293 total available probes corresponding to different Illumina TargetID. Each selected probe corresponds to a different transcript. Thus, we have n = 60 and p = 100. The data are available in the R package, `BDgraph`. Bhadra and Mallick (2013) have analyzed the data by adjusting the effect of SNPs using an expression quantitative trait loci (eQTL) mapping study. They found 54 significant interactions among the 100 traits considered. Previous studies have shown that these data are an interesting case study to carry out prediction.

We place a uniform distribution as an uninformative prior on the graph and the G-Wishart $W_G(3, I_{100})$ on the precision matrix. We run our BDMCMC algorithm for $60,000$ iterations with a $30,000$ sweeps burn-in.

The graph with the highest posterior probability is the graph with 281 edges, which includes almost all the significant interactions discovered by Bhadra and Mallick (2013). Figure 4.2 shows the selected graph with 86 edges, for which the posterior inclusion probabilities in (13) is greater then 0.6. Edges in the graph show the interactions among the genes. Figure 4.3 shows the image of the the all posterior inclusion probabilities for visualization.

4.3 Extension to time course data

Here, to demonstrate how well our proposed methodology can be extended to other types of graphical models, we focus on graphical models involving time series data (Dahlhaus and Eichler, 2003; Abegaz and Wit, 2013). We show how graphs can be useful in modeling real-world problems such as gene expression time course data.

Suppose we have a T time point longitudinal microarray study across p genes. We assume a stable dynamic graph structure for the time course data as follows:

$$x_t \sim N_p(f(t), K^{-1}), \qquad \text{for} \quad t = 1, ..., T, \tag{17}$$

in which vector $f(t) = \{f_i(t)\}_p$ with $f_i(t) = \beta_i' h(t) = \sum_{r=1}^{m} \beta_{ir} h_r(t)$, $\beta_i = (\beta_{i1}, ..., \beta_{im})'$, $h(t) = (h_1(t), ..., h_m(t))'$, and m is the number of basic elements. $h(t)$ is a cubic spline basis which should be continuous with continuous first and second derivatives (Hastie et al., 2009, chapter 5). The aim of this model is to find a parsimonious description of both time dynamics and gene interactions.

For this model, we place a uniform distribution as an uninformative prior on the

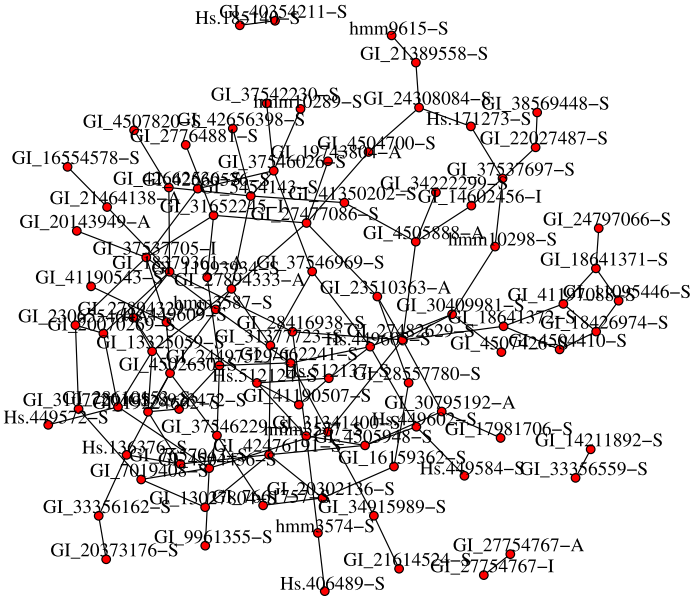


Figure 4: The inferred graph for the human gene expression data set. It reports the selected graph with 86 significant edges for which their posterior inclusion probabilities (13) are more then 0.6.

graph and a G-Wishart $W_G(3, I_p)$ on the precision matrix. For a prior distribution for β_i, we choose $N_p(\mu_{0i}, B_{0i})$, with $i = 1, ..., p$. Thus, based on our likelihood and priors, the conditional distribution of β_i is

$$\beta_i|\mathbf{x}, K, G \sim N_p(\mu_i, B_i), \tag{18}$$

in which

$$B_i = (B_{0i}^{-1} + K_{ii} \sum_{t=1}^{T} h(t)h^T(t))^{-1},$$
$$\mu_i = B_i(B_{0i}^{-1}\mu_{0i} + \sum_{t=1}^{T} h(t)(x_t' K_{V,i} - K_{i,V\setminus i} f_{-i}(t))).$$

Thus, to account for the time effect in our model, we require one more Gibbs sampling step in the BDMCMC algorithm for updating β_i, for $i = 1, ..., p$.

To evaluate the efficiency of the method, we focus on the mammary gland gene expression time course data from Stein et al. (2004). The data reports a large time course

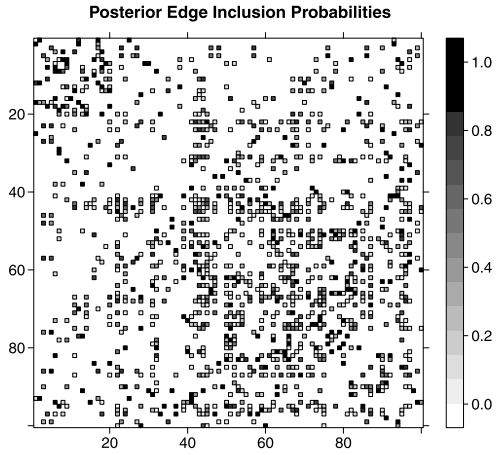


Figure 5: Image visualization of the posterior pairwise edge inclusion probabilities for all possible edges in the graph.

Affymetrix microarray experiment across different developmental stages performed by using mammary tissue from female mice. There are $12,488$ probe sets representing ~ 8600 genes. In total, the probe sets are measured across 54 arrays with three mice used at each of 18 time points. The time points are in the four main stages, as follows: `virgin`, 6, 10, and 12 weeks; `pregnancy`, 1, 2, 3, 8.5, 12.5, 14.5, and 17.5; `lactation`, 1, 3, and 7; `involution`, 1, 2, 3, 4, and 20. By using cluster analysis, we identify 30 genes which provide the best partition among the developmental stages. Those genes play a role in the transitions across the main developmental events. The mammary data is available in the R package `smida`; for more details about the data see Wit and McClure (2004, chapter one). Abegaz and Wit (2013) analyze this data based on a sparse time series chain graphical model. By using our proposed methodology, we infer the interactions between the crucial genes.

By placing a uniform prior on the graph and the G-Wishart $W_G(3, I_{30})$ on the precision matrix, we run our BDMCMC algorithm for $60,000$ iterations using $30,000$ as burn-in. Figure 4.3 shows the selected graph based on the output of our BDMCMC algorithm. The graph shows the edges with a posterior inclusion probability greater then 0.6. As we can see in Figure 4.3, the genes with the highest number of edges are LCN2, HSD17B, CRP1, and RABEP1, which each have 7 edges. Schmidt-Ott et al. (2007) suggested that gene LCN2 (lipocalin 2) plays an important role in the innate immune response to bacterial infection and also functions as a growth factor. For the gene HSD17B (17-β hydroxysteroid dehydrogenase), past studies suggest that this gene family provides each cell with the necessary mechanisms to control the level of intracellular androgens and/or estrogens (Labrie et al., 1997). Gene CRP1 was identified by Abegaz and Wit (2013) as a likely hub.

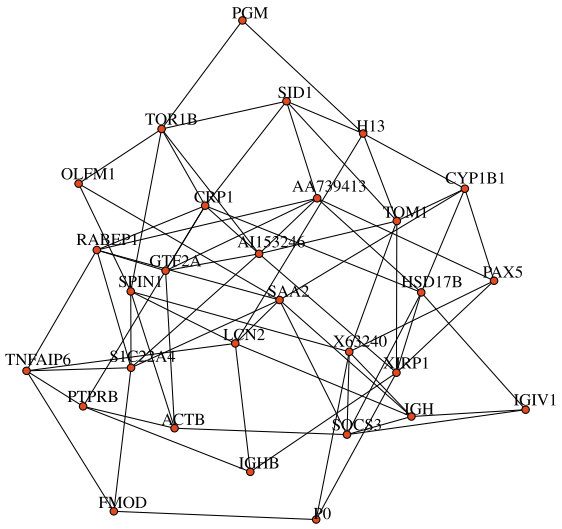


Figure 6: The inferred graph for the mammary gland gene expression data set. It reports the selected graph with 56 significant edges for which their posterior inclusion probabilities (13) are more then 0.6.

5 Discussion

We introduce a Bayesian approach for graph structure learning based on Gaussian graphical models using a trans-dimensional MCMC methodology. The proposed methodology is based on the birth-death process. In Theorem 3.1, we derived the conditions for which the balance conditions of the birth-death MCMC method holds. According to those conditions we proposed a convenient BDMCMC algorithm, whose stationary distribution is our joint posterior distribution. We showed that a scalable Bayesian method exists, which, also in the case of large graphs, is able to distinguish important edges from irrelevant ones and detect the true model with high accuracy. The resulting graphical model is reasonably robust to modeling assumptions and the priors used.

As we have shown in our simulation studies (4.1), in Gaussian graphical models, any kind of trans-dimensional MCMC algorithm which is based on a discrete time Markov process (such as reversible jump algorithms by Wang and Li, 2012 and Lenkoski, 2013) could suffer from high rejection rates, especially for high-dimensional graphs. However in our BDMCMC algorithm, moves between graphs are always accepted. In general, although our trans-dimensional MCMC algorithm has significant additional computing cost for birth and death rates, it has clear benefits over reversible jump style moves when graph structure learning in a non-hierarchical setting is of primary interest.

In Gaussian graphical models, Bayesian structure learning has several computational

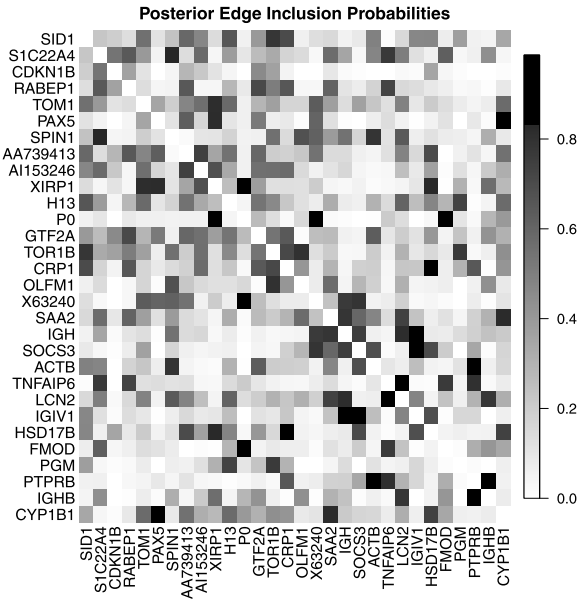


Figure 7: Image visualization of the posterior pairwise edge inclusion probabilities of all possible edges in the graph.

and methodological issues as the dimension grows: (1) convergence, (2) computation of the prior normalizing constant, and (3) sampling from the posterior distribution of the precision matrix. Our Bayesian approach efficiently eliminates these problems. For convergence, Cappé et al. (2003) demonstrate the strong similarity of reversible jump and continuous time methodologies by showing that, on appropriate rescaling of time, the reversible jump chain converges to a limiting continuous time birth-death process. In Section 4.1 we show the fast convergence feature of our BDMCMC algorithm. For the second problem, in Subsection 3.2, by using the ideas from Wang and Li (2012) and Lenkoski (2013) we show that the exchange algorithm circumvents the intractable normalizing constant. For the third problem, we used the exact sampler algorithm which was proposed by Lenkoski (2013).

Our proposed method provides a flexible framework to handle the graph structure and it could be extended to different types of priors for the graph and precision matrix. In Subsection 4.3, we illustrate how our proposed model can be integrated in types of graphical models, such as a multivariate time series graphical models. Although we have focused on normally distributed data, in general, we can extend our proposed method to other types of graphical models, such as log-linear models (see e.g. Dobra et al., 2011a and Lenkoski and Dobra, 2011), non-Gaussianity data by using copula transition (see e.g. Dobra et al., 2011b), or copula regression models (see e.g. Pitt et al., 2006). This will be considered in future work.

Appendix 1: Proof of theorem 3.1

Before we derive the detailed balance conditions for our BDMCMC algorithm we introduce some notation.

Assume the process is at state (G, K), in which $G = (V, E)$ with precision matrix $K \in \mathbb{P}_G$. The process behavior is defined by the *birth rates* $\beta_e(K)$, the *death rates* $\delta_e(K)$, and the birth and death *transition kernels* $T^G_{\beta_e}(K;.)$ and $T^G_{\delta_e}(K;.)$. For each $e \in \overline{E}$, $T^G_{\beta_e}(K;.)$ denotes the probability that the process jumps from state (G, K) to a point in the new state $\cup_{K^* \in \mathbb{P}_{G^{+e}}}(G^{+e}, K^*)$. Hence, if $\mathcal{F} \subset \mathbb{P}_{G^{+e}}$ we have

$$T^G_{\beta_e}(K;\mathcal{F}) = \frac{\beta_e(K)}{\beta(K)} \int_{k_e : K \cup k_e \in \mathcal{F}} b_e(k_e; K) dk_e. \tag{19}$$

Likewise, for each $e \in E$, $T^G_{\delta_e}(K;.)$ denotes the probability that the process jumps from state (G, K) to a point in the new state $\cup_{K^* \in \mathbb{P}_{G^{-e}}}(G^{-e}, K^*)$. Therefore, if $\mathcal{F} \subset \mathbb{P}_{G^{-e}}$ we have

$$\begin{aligned} T^G_{\delta_e}(K;\mathcal{F}) &= \sum_{\eta \in E : K \setminus k_\eta \in \mathcal{F}} \frac{\delta_\eta(K)}{\delta(K)} \\ &= \frac{\delta_e(K)}{\delta(K)} I(K^{-e} \in \mathcal{F}). \end{aligned}$$

Detailed balance conditions. In our birth-death process, $P(K, G|\mathbf{x})$ satisfies detailed balance conditions if

$$\int_{\mathcal{F}} \delta(K) dP(K, G|\mathbf{x}) = \sum_{e \in \overline{E}} \int_{\mathbb{P}_{G^{-e}}} \beta(K^{-e}) T^G_{\beta_e}(K^{-e}; \mathcal{F}) dP(K^{-e}, G^{-e}|\mathbf{x}), \tag{20}$$

and

$$\int_{\mathcal{F}} \beta(K) dP(K, G|\mathbf{x}) = \sum_{e \in E} \int_{\mathbb{P}_{G^{+e}}} \delta(K^{+e}) T^G_{\delta_e}(K^{+e}; \mathcal{F}) dP(K^{+e}, G^{+e}|\mathbf{x}), \tag{21}$$

where $\mathcal{F} \subset \mathbb{P}_G$.

The first expression says edges that enter the set $\mathcal{F}$ due to the deaths must be matched by edges that leave that set due to the births, and vice versa for the second part.

To prove the first part (20), we have

$$\begin{aligned} LHS &= \int_{\mathcal{F}} \delta(K) dP(G, K|\mathbf{x}) \\ &= \int_{\mathbb{P}_G} I(K \in \mathcal{F}) \delta(K) dP(G, K|\mathbf{x}) \end{aligned}$$

$$
\begin{aligned}
&= \int_{\mathbb{P}_G} I(K \in \mathcal{F}) \sum_{e \in E} \delta_e(K) dP(G, K|\mathbf{x}) \\
&= \sum_{e \in E} \int_{\mathbb{P}_G} I(K \in \mathcal{F}) \delta_e(K) dP(G, K|\mathbf{x}) \\
&= \sum_{e \in E} \int I(K \in \mathcal{F}) \delta_e(K) P(G, K|\mathbf{x}) \prod_{i=1}^{p} dk_{ii} \prod_{(i,j) \in E} dk_{ij}.
\end{aligned}
$$

For the RHS, by using (19) we have

$$
\begin{aligned}
RHS &= \sum_{e \in \overline{E}} \int_{\mathbb{P}_{G^{-e}}} \beta(K^{-e}) T^{G}_{\beta_e}(K^{-e}; \mathcal{F}) dP(K^{-e}, G^{-e}|\mathbf{x}) \\
&= \sum_{e \in \overline{E}} \int_{\mathbb{P}_{G^{-e}}} \beta_e(K) \int_{k_e : K^{-e} \cup k_e \in \mathcal{F}} b_e(k_e; K) dk_e dP(K^{-e}, G^{-e}|\mathbf{x}) \\
&= \sum_{e \in \overline{E}} \int_{\mathbb{P}_{G^{-e}}} \int_{k_e} I(K \in \mathcal{F}) \beta_e(K) b_e(k_e; K) dk_e dP(K^{-e}, G^{-e}|\mathbf{x}) \\
&= \sum_{e \in \overline{E}} \int I(K \in \mathcal{F}) \beta_e(K) b_e(k_e; K) P(K^{-e}, G^{-e}|\mathbf{x}) \prod_{i=1}^{p} dk_{ii} \prod_{(i,j) \in E} dk_{ij}.
\end{aligned}
$$

By putting

$$
\delta_e(K) P(G, K|\mathbf{x}) = \beta_e(K) b_e(k_e; K) P(K^{-e}, G^{-e}|\mathbf{x}),
$$

we have LHS=RHS. Now, in the above equation

$$
P(G, K|\mathbf{x}) = P(G, K \setminus (k_{ij}, k_{jj})|\mathbf{x}) P((k_{ij}, k_{jj})|K \setminus (k_{ij}, k_{jj}), G, \mathbf{x}),
$$

and

$$
P(G^{-e}, K^{-e}|\mathbf{x}) = P(G^{-e}, K^{-e} \setminus k_{jj}|\mathbf{x}) P(k_{jj}|K^{-e} \setminus k_{jj}, G^{-e}, \mathbf{x}).
$$

We simply choose the proposed density for the new element $k_e = k_{ij}$ as follows:

$$
b_e(k_e; K) = \frac{P((k_{ij}, k_{jj})|K \setminus (k_{ij}, k_{jj}), G, \mathbf{x})}{P(k_{jj}|K^{-e} \setminus k_{jj}, G^{-e}, \mathbf{x})}.
$$

Therefore, we reach the expression in Theorem 3.1. The proof for the second part (21) is the same.

Appendix 2: Proposition

Let A be a 2×2 random matrix with Wishart distribution $W(b, D)$ as below

$$
P(A) = \frac{1}{I(b, D)} |A|^{(b-2)/2} \exp\left\{ -\frac{1}{2} \mathrm{tr}(DA) \right\},
$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix}, \qquad D = \begin{bmatrix} d_{11} & d_{12} \\ d_{12} & d_{22} \end{bmatrix}.$$

Then

(i) $a_{11} \sim W(b+1, D_{11.2})$ where $D_{11.2} = d_{11} - d_{22}^{-1} d_{21}^2$,

(ii)

$$\begin{aligned} P(a_{12}, a_{22}|a_{11}) &= \frac{P(A)}{P(a_{11})} \\ &= \frac{1}{J(b, D, a_{11})} |A|^{(b-2)/2} \exp\left\{-\frac{1}{2}\text{tr}(DA)\right\}, \end{aligned}$$

where

$$J(b, D, a_{11}) = \left(\frac{2\pi}{d_{22}}\right)^{\frac{1}{2}} I(b, d_{22}) a_{11}^{\frac{(b-1)}{2}} \exp\left\{-\frac{1}{2} D_{11.2} a_{11}\right\}.$$

Proof. For proof of part (i), see Muirhead (1982, Theorem 3.2.10). The result for part (ii) is immediate by using part (i).

References

Abegaz, F. and Wit, E. (2013). "Sparse time series chain graphical models for reconstructing genetic networks." *Biostatistics*, 14(3): 586–599. 127, 129

Albert, R. and Barabási, A.-L. (2002). "Statistical mechanics of complex networks." *Reviews of modern physics*, 74(1): 47. 121

Atay-Kayis, A. and Massam, H. (2005). "A Monte Carlo method for computing the marginal likelihood in nondecomposable Gaussian graphical models." *Biometrika*, 92(2): 317–335. 112

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A., and Nielsen, H. (2000). "Assessing the accuracy of prediction algorithms for classification: an overview." *Bioinformatics*, 16(5): 412–424. 122

Bhadra, A. and Mallick, B. K. (2013). "Joint High-Dimensional Bayesian Variable and Covariance Selection with an Application to eQTL Analysis." *Biometrics*, 69(2): 447–457. 126, 127

Cappé, O., Robert, C., and Rydén, T. (2003). "Reversible jump, birth-and-death and more general continuous time Markov chain Monte Carlo samplers." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(3): 679–700. 114, 115, 118, 131

Carvalho, C. M., , and Scott, J. G. (2009). "Objective Bayesian model selection in Gaussian graphical models." *Biometrika*, 96(3): 497–512. 112

Chen, L., Tong, T., and Zhao, H. (2008). "Considering dependence among genes and markers for false discovery control in eQTL mapping." *Bioinformatics*, 24(18): 2015–2022. 126

Cheng, Y., Lenkoski, A., et al. (2012). "Hierarchical Gaussian graphical models: Beyond reversible jump." *Electronic Journal of Statistics*, 6: 2309–2331. 110

Dahlhaus, R. and Eichler, M. (2003). "Causality and graphical models in time series analysis." *Oxford Statistical Science Series*, 115–137. 127

Dempster, A. (1972). "Covariance selection." *Biometrics*, 28(1): 157–175. 110

Dobra, A., Lenkoski, A., and Rodriguez, A. (2011a). "Bayesian inference for general Gaussian graphical models with application to multivariate lattice data." *Journal of the American Statistical Association*, 106(496): 1418–1433. 109, 110, 131

Dobra, A., Lenkoski, A., et al. (2011b). "Copula Gaussian graphical models and their application to modeling functional disability data." *The Annals of Applied Statistics*, 5(2A): 969–993. 131

Foygel, R. and Drton, M. (2010). "Extended Bayesian Information Criteria for Gaussian Graphical Models." In Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 23*, 604–612. 124

Friedman, J., Hastie, T., and Tibshirani, R. (2008). "Sparse inverse covariance estimation with the graphical lasso." *Biostatistics*, 9(3): 432–441. 109, 124, 125, 126

Geyer, C. J. and Møller, J. (1994). "Simulation procedures and likelihood inference for spatial point processes." *Scandinavian Journal of Statistics*, 359–373. 110

Giudici, P. and Castelo, R. (2003). "Improving Markov chain Monte Carlo model search for data mining." *Machine Learning*, 50(1-2): 127–158. 109

Giudici, P. and Green, P. (1999). "Decomposable graphical Gaussian model determination." *Biometrika*, 86(4): 785–801. 110

Green, P. (1995). "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination." *Biometrika*, 82(4): 711–732. 110

Green, P. J. (2003). "Trans-dimensional Markov chain Monte Carlo." *Oxford Statistical Science Series*, 179–198. 110

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer. 117, 127

Jones, B., Carvalho, C., Dobra, A., Hans, C., Carter, C., and West, M. (2005). "Experiments in stochastic computation for high-dimensional graphical models." *Statistical Science*, 20(4): 388–400. 109, 112

Kullback, S. and Leibler, R. A. (1951). "On information and sufficiency." *The Annals of Mathematical Statistics*, 22(1): 79–86. 125

Labrie, F., Luu-The, V., Lin, S.-X., Claude, L., Simard, J., Breton, R., and Bélanger, A. (1997). "The key role of 17β-hydroxysteroid dehydrogenases in sex steroid biology." *Steroids*, 62(1): 148–158. 129

Lauritzen, S. (1996). *Graphical models*, volume 17. Oxford University Press, USA. 109, 111

Lenkoski, A. (2013). "A direct sampler for G-Wishart variates." *Stat*, 2(1): 119–128. 110, 116, 117, 119, 120, 121, 122, 123, 124, 130, 131

Lenkoski, A. and Dobra, A. (2011). "Computational aspects related to inference in Gaussian graphical models with the G-Wishart prior." *Journal of Computational and Graphical Statistics*, 20(1): 140–157. 112, 131

Letac, G. and Massam, H. (2007). "Wishart distributions for decomposable graphs." *The Annals of Statistics*, 35(3): 1278–1323. 112

Liang, F. (2010). "A double Metropolis–Hastings sampler for spatial models with intractable normalizing constants." *Journal of Statistical Computation and Simulation*, 80(9): 1007–1022. 110, 117

Liu, H., Roeder, K., and Wasserman, L. (2010). "Stability Approach to Regularization Selection (StARS) for High Dimensional Graphical Models." In *Advances in Neural Information Processing Systems*, 1432–1440. 124

Meinshausen, N. and Bühlmann, P. (2006). "High-dimensional graphs and variable selection with the lasso." *The Annals of Statistics*, 34(3): 1436–1462. 109, 124, 125

Mohammadi, A. and Wit, E. C. (2013). *BDgraph: Graph estimation based on birth-death MCMC*. R package version 2.10.
http://CRAN.R-project.org/package=BDgraph 118

Muirhead, R. (1982). *Aspects of multivariate statistical theory*, volume 42. Wiley Online Library. 112, 134

Murray, I., Ghahramani, Z., and MacKay, D. (2012). "MCMC for doubly-intractable distributions." *arXiv preprint arXiv:1206.6848*. 110, 116, 117, 124

Pitt, M., Chan, D., and Kohn, R. (2006). "Efficient Bayesian inference for Gaussian copula regression models." *Biometrika*, 93(3): 537–554. 131

Powers, D. M. (2011). "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation." *Journal of Machine Learning Technologies*, 2(1): 37–63. 122

Preston, C. J. (1976). "Special birth-and-death processes." *Bulletin of the International Statistical Institute*, 46: 371–391. 113, 114

Ravikumar, P., Wainwright, M. J., Lafferty, J. D., et al. (2010). "High-dimensional Ising model selection using L1-regularized logistic regression." *The Annals of Statistics*, 38(3): 1287–1319. 109

Ripley, B. D. (1977). "Modelling spatial patterns." *Journal of the Royal Statistical Society. Series B (Methodological)*, 172–212. 110

Roverato, A. (2002). "Hyper Inverse Wishart Distribution for Non-decomposable Graphs and its Application to Bayesian Inference for Gaussian Graphical Models." *Scandinavian Journal of Statistics*, 29(3): 391–411. 112, 115

Schmidt-Ott, K. M., Mori, K., Li, J. Y., Kalandadze, A., Cohen, D. J., Devarajan, P., and Barasch, J. (2007). "Dual action of neutrophil gelatinase–associated lipocalin." *Journal of the American Society of Nephrology*, 18(2): 407–413. 129

Scott, J. G. and Berger, J. O. (2006). "An exploration of aspects of Bayesian multiple testing." *Journal of Statistical Planning and Inference*, 136(7): 2144–2162. 112

Scutari, M. (2013). "On the Prior and Posterior Distributions Used in Graphical Modelling." *Bayesian Analysis*, 8(1): 1–28. 112

Stein, T., Morris, J. S., Davies, C. R., Weber-Hall, S. J., Duffy, M.-A., Heath, V. J., Bell, A. K., Ferrier, R. K., Sandilands, G. P., and Gusterson, B. A. (2004). "Involution of the mouse mammary gland is associated with an immune cascade and an acute-phase response, involving LBP, CD14 and STAT3." *Breast Cancer Research*, 6(2): R75–91. 128

Stephens, M. (2000). "Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods." *Annals of Statistics*, 28(1): 40–74. 110

Stranger, B. E., Nica, A. C., Forrest, M. S., Dimas, A., Bird, C. P., Beazley, C., Ingle, C. E., Dunning, M., Flicek, P., Koller, D., et al. (2007). "Population genomics of human gene expression." *Nature genetics*, 39(10): 1217–1224. 126

Wang, H. (2012). "Bayesian graphical lasso models and efficient posterior computation." *Bayesian Analysis*, 7(4): 867–886. 109, 112

— (2014). "Scaling It Up: Stochastic Search Structure Learning in Graphical Models." http://www.stat.sc.edu/~wang345/RESEARCH/Wang2013WP.pdf 109, 112

Wang, H. and Li, S. (2012). "Efficient Gaussian graphical model determination under G-Wishart prior distributions." *Electronic Journal of Statistics*, 6: 168–198. 109, 110, 115, 116, 118, 119, 120, 121, 123, 124, 130, 131

Wang, H. and Pillai, N. S. (2013). "On a class of shrinkage priors for covariance matrix estimation." *Journal of Computational and Graphical Statistics*, 22(3): 689–707. 112

Wit, E. and McClure, J. (2004). *Statistics for Microarrays: Design, Analysis and Inference*. John Wiley & Sons. 129

Zhao, P. and Yu, B. (2006). "On model selection consistency of Lasso." *The Journal of Machine Learning Research*, 7: 2541–2563. 109

Zhao, T., Liu, H., Roeder, K., Lafferty, J., and Wasserman, L. (2012). "The Huge Package for High-dimensional Undirected Graph Estimation in R." *The Journal of Machine Learning Research*, 13(1): 1059–1062. 124

Acknowledgments

The authors wish to thank two referees and an associate editor for their suggestions which led to significant improvement of this paper. The authors are also grateful to Alex Lenkoski for the code for the direct sampler algorithm from the G-Wishart distribution and to Hao Wang for his suggestions.