A generalized likelihood-based Bayesian approach for scalable joint regression and covariance selection in high dimensions

Srijata Samanta · Kshitij Khare · George Michailidis

Received: 4 June 2021 / Accepted: 27 April 2022 / Published online: 3 June 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
The paper addresses joint sparsity selection in the regression coefficient matrix and the error precision (inverse covariance) matrix for high-dimensional multivariate regression models in the Bayesian paradigm. The selected sparsity patterns are crucial to help understand the network of relationships between the predictor and response variables, as well as the conditional relationships among the latter. While Bayesian methods have the advantage of providing natural uncertainty quantification through posterior inclusion probabilities and credible intervals, current Bayesian approaches either restrict to specific subclasses of sparsity patterns and/or are not scalable to settings with hundreds of responses and predictors. Bayesian approaches that only focus on estimating the posterior mode are scalable, but do not generate samples from the posterior distribution for uncertainty quantification. Using a bi-convex regression-based generalized likelihood and spike-and-slab priors, we develop an algorithm called joint regression network selector (JRNS) for joint regression and covariance selection, which (a) can accommodate general sparsity patterns, (b) provides posterior samples for uncertainty quantification, and (c) is scalable and orders of magnitude faster than the state-of-the-art Bayesian approaches providing uncertainty quantification. We demonstrate the statistical and computational efficacy of the proposed approach on synthetic data and through the analysis of selected cancer data sets. We also establish high-dimensional posterior consistency for one of the developed algorithms.

Keywords
High-dimensional data · Sparse multivariate regression and covariance selection · Posterior consistency · Generalized likelihood

1 Introduction
We consider joint variable and precision matrix selection in high-dimensional multivariate regression models with multiple responses. In particular, we consider two sets of variables: the $n \times p$ matrix $X$ whose rows $x_1, \ldots, x_n \in \mathbb{R}^p$ comprise of $n$ samples on $p$ predictor variables and the $n \times q$ matrix $Y$ whose rows $y_1, \ldots, y_n \in \mathbb{R}^q$ comprise of $n$ matched samples (same set of entities) on $q$ response variables. We are interested in inferring a graphical model on the variables from the $Y$ data, while accounting for the effect of the $X$ data. The corresponding multivariate regression model is given by:

$$Y = XB + \varepsilon$$

where $\varepsilon$ is an $n \times q$ matrix whose rows $\varepsilon_1, \ldots, \varepsilon_n \in \mathbb{R}^q$ comprise of the $n$ noise vectors and $B$ is a $p \times q$ matrix of regression coefficients. To make things concrete, assume that $y_1, y_2, \ldots, y_n$ are independent, and

$$y_i \sim \mathcal{N}_q\left(B^T x_i, \Omega^{-1}\right) \text{ for } i = 1, 2, \ldots, n.$$  

This is equivalent to assuming that the noise vectors $\varepsilon_1, \ldots, \varepsilon_n$ are i.i.d. $\mathcal{N}_q(0, \Omega^{-1})$. The $q \times q$ matrix $\Omega$ captures the dependence between the response variables conditional on the predictor variables, and the $p \times q$ matrix $B$ captures the effect of the predictor variables on the response variables.

For example, in molecular biology applications, multiple Omics modalities are profiled on the same set of samples. Then, following the central dogma of biology, the predictor variables $X$ could correspond to the DNA level (e.g., copy number or methylation data), while the response variables...
Y to the transcriptomic level (mRNA expression). Another possibility is that the predictors correspond to the transcriptomic level and the responses to the proteomic level. Thus, the regression coefficients in B encode transcriptional or translational dependencies, while the entries of Ω reflect statistical associations within a molecular compartment.

We focus on the problem under a high-dimensional setting, wherein p and/or q is larger than or comparable to the sample size n. In such sample starved settings, imposing sparsity in B and Ω offers a simple and effective approach for reducing the effective number of parameters. The sparsity patterns in B and Ω often have specific scientific interpretations and can help researchers understand the underlying relationships between variables in the data set. To summarize, our goal is simultaneous sparse estimation of B and Ω and the use of estimated sparsity patterns to understand relevant dependence structures.

The above problem has been studied in the literature. On the frequentist side, various penalized likelihood-based methods have been proposed. Many of these methods use an ℓ1 penalty that encourages sparsity both in B and Ω. Various optimization algorithms have been employed, see Friedman et al. (2008), Rothman et al. (2010), Lee and Liu (2012), Cai et al. (2013) and Lin et al. (2016) and references therein. Note that the conditional log-likelihood for response Y given X can be written as:

\[
\ell(Y|X, B, \Omega) = \text{constant} + \frac{n}{2} \log \det \Omega - \frac{1}{2} \text{tr} \left( \Omega \sum_{i=1}^{n} (y_i - B^T x_i)(y_i - B^T x_i)^T \right) \tag{2}
\]

and is not jointly convex in B and Ω. Hence, many popular algorithms (e.g., block coordinate descent) for optimizing a penalized version of this log-likelihood may fail to converge to the global optimum, especially in settings where \(p > n\), as pointed out in Lee and Liu (2012). However, the log-likelihood is bi-convex, i.e., it is convex in B for fixed Ω and in Ω for fixed B. The bi-convexity is leveraged in Lin et al. (2016) to develop a two-block coordinate descent algorithm, which converges to a stationary point of the objective function, assuming that all iterates need to be within a ball of certain radius \(R(p, q, n)\) (a function of the model dimensions p, q and the sample size n) that in addition contains the true data-generating parameters. This condition is then shown to hold with high probability.

Some recent papers such as Sohn and Kim (2012), Yuan and Zhang (2014); McCarter and Kim (2014) consider an alternate parameterization \((\tilde{B}, \Omega)\) where \(\tilde{B} = -B^T \Omega\). The likelihood can be shown to be jointly convex in \((\tilde{B}, \Omega)\), and the respective algorithms in these papers provide sparse estimates of \(\tilde{B}\) and \(\Omega\) by using appropriate ℓ1 penalties. However, sparsity in \(\tilde{B}\) does not in general correspond to sparsity in B. In many applications, the linear model with the \((B, \Omega)\) parameterization is the natural modeling tool, and sparsity in the regression coefficient matrix B has a specific scientific interpretation. This interpretability may be lost if sparsity is instead imposed on \(\tilde{B}\). See (Lin et al. 2016, Sect. 5) for a detailed discussion.

Bayesian methods offer a natural framework for addressing uncertainty quantification of model parameters through the posterior distribution, and several Bayesian approaches have also been proposed in the literature. Brown et al. (1998) propose a Bayesian approach for the joint estimation of B and Ω, but restrict the sparsity pattern in B to be such that each row of B is completely sparse or completely dense. Richardson et al. (2010) allow for a general sparsity pattern in B, but restrict Ω to be a diagonal matrix. Bhadra and Mallick (2013) use spike-and-slab prior distributions to induce sparsity in B (conditional on Ω), and G-Wishart prior distributions coupled with independent Bernoulli priors on the sparsity pattern in Ω. Similar to Brown et al. (1998), they restrict the rows of B to be completely sparse or completely dense and also restrict the sparsity pattern in Ω to corresponding to a decomposable graph. In a related work Consonni et al. (2017), the authors develop an objective Bayesian approach for directed acyclic graph estimation in the presence of covariates. This approach induces sparsity in the Cholesky factor of Ω and corresponds to directly inducing sparsity in Ω, when the underlying sparsity pattern is decomposable. In a recent work, Deshpande et al. (2019) propose a scalable Bayesian approach using spike-and-slab Laplace prior distributions to induce sparsity in B and Ω. The work employs an expectation conditional maximization algorithm to find the (sparse) posterior mode and thereby obtain sparse estimates of B and Ω. However, methods to generate samples from the posterior distribution are not explored, and hence, uncertainty quantification in the form of posterior credible regions/intervals is not available. In Li et al. (2021), the authors propose a Gaussian likelihood-based fully Bayesian procedure for the simultaneous estimation of the mean vector and the inverse covariance matrix, which provides measures of uncertainty. However, in moderate-/high-dimensional settings it might run into scalability issues as pointed out in Sect. 4.

Note that in most of the above-cited literature, and in this paper, two layers of variables are considered. The matrix B captures the effect of the top layer (predictors) on the bottom layer (responses), while the matrix Ω captures the conditional covariance structure of the bottom layer. It is possible to consider a scenario where we have a chain of multiple layers of variables, each layer affecting the layer below it (see the setting in Lin et al. (2016)), thereby giving rise to multiple pairs of B and Ω matrices. Due to the factorization of the likelihood-based on the Markov property.
induced by the chain structure, any method developed for the two-layer setting can be extended in a reasonably straightforward way to the multiple layer setting, subject to some model parameter identifiability restrictions (see Sect. 3.4 in Lin et al. (2016)).

In Ha et al. (2020b), the authors consider the multiple layer setting and develop a generalized likelihood-based Bayesian approach for simultaneous sparse estimation of the $B$ and $\Omega$ pair. In Peng et al. (2009) and Khare et al. (2015), the use of a regression-based generalized likelihood has been shown to significantly improve the computational efficiency compared to Gaussian likelihood-based methods for standard graphical models (no presence of predictors). In Ha et al. (2020b), sparsity-inducing spike-and-slab prior distributions are used for the entries of $B$ and $\Omega$, and an MCMC algorithm (called BANS) based on add/delete/swap moves in the space of sparsity patterns is developed to generate approximate samples from the posterior distribution. In simulation experiments, scenarios with up to ten layers of variables and up to 20 variables in each layer are considered. However, the algorithm starts to run into serious computational issues when the number of variables in each layer is bumped up to 200 (with two layers). One reason for this is the need for several matrix inversions to compute Metropolis–Hastings-based rejection probabilities in each iteration of the BANS algorithm (see Sects. 2.2 and 4 for more details). A faster algorithm called BANS-parallel, wherein computations corresponding to each response variable can be parallelized, has also been developed in Ha et al. (2020b). However, this approach ignores the symmetry in $\Omega$, which negatively affects the quality of the estimates and again requires matrix inversions for various Metropolis–Hastings steps (see Remark 1 at the end of Sect. 2). In short, existing Bayesian approaches suffer from at least one of the following drawbacks: (i) restrict to a subclass of sparsity patterns; (ii) focus on estimating the posterior mode and not on sampling from the posterior distribution; (iii) are not computationally scalable due to excessive use of matrix inversions.

The goal of this paper is to develop a computationally scalable generalized likelihood-based Bayesian procedure for joint regression and precision matrix selection, which can account for arbitrary sparsity patterns in $B$ and $\Omega$, and provide uncertainty quantification. First, we leverage ideas in Khare et al. (2015) for standard graphical models (no predictors) to the current setting and construct a regression-based generalized likelihood that is bi-convex in $\Omega$ and $B$. The generalized likelihood in Ha et al. (2020b), which corresponds to the predictor-adjusted version of the generalized likelihood in Peng et al. (2009), is neither jointly convex, nor bi-convex. In the standard graphical model setting, it has been demonstrated in Khare et al. (2015) that convexity plays an important role in improved algorithmic and empirical performance of the generalized likelihood as compared to the one in Peng et al. (2009). Next, we develop a Gibbs sampling algorithm (referred to as the joint algorithm) to sample from the corresponding posterior (using spike-and-slab priors for entries of $B$ and $\Omega$). With entry-wise updates of $B$ and $\Omega$ involving standard distributions, we completely avoid the matrix inversions needed for the Metropolis–Hastings steps in BANS and the resulting algorithm is significantly computationally faster. As an illustrative example, with 200 responses and 200 predictors, the proposed MCMC algorithm, coded in R/Rcpp, completes 3000 iterations (each iteration cycles through all the entries of $B$ and $\Omega$) in less than 5 minutes. In the same setting, BANS (implemented using R/Rcpp code available on Github) was only able to finish less than 100 iterations in 4 days.

Several frequentist methods in the literature, such as those in Lee and Liu (2012) and Cai et al. (2013), consider a stepwise approach for sparse estimation of $B$ and $\Omega$. In this approach, $q$ regressions corresponding to each of the responses are used to obtain sparse estimates of columns of $B$. The resulting estimate of $B$ is used to compute plug-in covariate-adjusted responses, subsequently provided to a standard graphical model estimation procedure to obtain a sparse estimate of $\Omega$. As a Bayesian analog of this approach, and for faster computation, we develop a stepwise algorithm for joint regression/covariance selection. In this approach, we first focus on estimation/selection for $B$ by treating $\Omega$ as a diagonal matrix and combining the resulting Gaussian likelihood with spike-and-slab priors on entries of $B$. An appropriate posterior estimate $\hat{B}$ of $B$ is then used to compute covariate-adjusted responses (or pseudo-errors) $\hat{y}_i = y_i - \hat{B}^T x_i$. In the second step of the algorithm, the generalized likelihood of Khare et al. (2015), along with spike-and-slab priors on the entries of $\Omega$, is used for selecting the sparsity pattern in $\Omega$. The computational advantage obtained by ignoring the cross-correlation in the responses for the $B$ estimation step clearly comes with the cost of some loss of statistical efficiency. However, we rigorously establish high-dimensional posterior model selection and estimation consistency of the resulting estimates in Sect. 3.1. As expected, the simulation experiments in Sect. 4 in general demonstrate a loss in statistical accuracy and roughly two times improvement in computational performance as compared to the joint algorithm.

The remainder of the paper is organized as follows: The joint and the stepwise algorithms are developed in Sects. 2 and 3, respectively. High-dimensional posterior consistency results for the stepwise algorithm are provided in Sect. 3.1. An extensive simulation study evaluating the empirical performance of the proposed algorithms is presented in Sect. 4, and an analysis of a cancer data set is presented in Sect. 5. Proofs of the technical results along with additional simulation details are provided in Supplementary document.
2 Joint sparsity selection for $B$ and $\Omega$ using a bi-convex generalized likelihood

We develop a generalized likelihood-based Bayesian approach for jointly estimating the sparsity patterns in $B$ and $\Omega$. Consider the log-likelihood denoted by $\ell(Y \mid X, B, \Omega)$ in (2). One reason why block updates corresponding to optimization/MCMC algorithms for the corresponding penalized objective functions/posteriors run into computational issues, even in moderate-dimensional settings, is the presence of the log det $\Omega$ term, which leads to expensive matrix inverse computations. Let $y_j$ denote the $j^{th}$ column of the $n \times q$ data matrix $Y$. Hence, $y_j$ is the collection of all the $n$ observations corresponding to the $j^{th}$ response. Then, the conditional density of $y_j$ given all the other responses (and of course, conditional on $X$) is given by

\[
\left(\frac{\omega_{jj}}{2\pi}\right)^{n/2} \exp\left\{ -\frac{1}{2\omega_{jj}} \left\| (y_j - XB)_j \right\|^2 \right\}
\]

where $B_j$ and $\Omega_j$ denote the $j^{th}$ columns of $B = ((b_{jk}))$ and $\Omega = ((\omega_{jk}))$, respectively, and $\|a\|^2 = a^T a$. The above follows by noting that when regressing the $j^{th}$ variable against the other variables, the regression coefficient of the $k^{th}$ variable is given by $-\omega_{jk}/\omega_{kk}$. Using ideas in Besag (1975), a generalized likelihood for $(B, \Omega)$ can be defined by taking the product of these conditional densities. This is the regression-based generalized likelihood used for the BANS algorithm in Ha et al. (2020b), and its form is given by:

\[
L_{G,BANS}(Y \mid X, B, \Omega) = \left(\frac{\prod_{j=1}^{q}(\omega_{jj})^{n/2}}{(2\pi)^{n/2}}\right) \exp\left\{ -\sum_{j=1}^{q} \frac{1}{2\omega_{jj}} \left\| (Y - XB)\Omega_j \right\|^2 \right\}.
\]

Taking logarithm of the expression in (4) shows that the problematic log det $\Omega$ term in the log-Gaussian likelihood (2) is now replaced by a much simpler $\sum_{j=1}^{q} \log \omega_{jj}$ term in the generalized log-likelihood. However, the bi-convexity is lost, i.e., given $B$, the function log $L_{G,BANS}$ is not convex in $\Omega$. In the simpler setting of Gaussian graphical models with no predictors (i.e., no $B$), it was shown in Khare et al. (2015) that this lack of convexity can lead to severe convergence issues (in a penalized optimization context) and a convex version of the generalized likelihood was constructed. We adapt this idea in the more general setting of joint regression and precision matrix estimation in a Bayesian context.

In particular, by ‘weighting’ each observation with $\omega_{jj}^{-1/2}$ for the expression in (3), i.e., using the conditional density of $\omega_{jj}^{-1/2} y_j$ and then taking the product over every $1 \leq j \leq p$, we get the generalized likelihood:

\[
\tilde{L}_{G,\text{joint}}(Y \mid X, B, \Omega) = \prod_{j=1}^{q} \frac{(\omega_{jj})^{n}}{(2\pi)^{n/2}} \exp\left\{ -\sum_{j=1}^{q} \frac{1}{2\omega_{jj}} \left\| (Y - XB)\Omega_j \right\|^2 \right\}.
\]

Next, we discuss some important features related to $\tilde{L}_{G,\text{joint}}$ and its use for Bayesian inference.

- The exponent in $\tilde{L}_{G,\text{joint}}$ now becomes a quadratic form in $\Omega$, and the power of $\omega_{jj}$ is now $n$ instead of $n/2$ (as compared to $L_{G,BANS}$). Hence, log $\tilde{L}_{G,\text{joint}}$ is bi-convex (convex in $\Omega$ given $B$, convex in $B$ given $\Omega$) and in general analytically more tractable than log $L_{G,BANS}$.

- Note that our primary goal, as far as $\Omega$ is concerned, is sparsity selection. Hence, following Meinshausen and Bühlmann (2006), Peng et al. (2009); Khare et al. (2015), we relax the constraint of positive definiteness for $\Omega$ to the simpler constraint of just having positive diagonal entries. This relaxation leads to significant improvement in computational scalability. If a positive-definite estimate of $\Omega$ is needed for a downstream application, it can be obtained by a quick refitting step restricting to the selected sparsity pattern. The same relaxation of the positive-definiteness constraint is also used for the BANS algorithm in Ha et al. (2020b). Such a relaxation is not possible for the Gaussian likelihood because of the presence of the det $\Omega$ term.

- Although a generalized likelihood is not a probability density anymore, it can still be regarded as a data-based weight function, and as long as the product of the generalized likelihood and the specified prior density is integrable over the parameter space, one can construct a posterior distribution and carry out Bayesian inference (see Bissiri et al. (2016); Alquier (2020) and the references therein).

To induce sparsity, we use spike-and-slab prior distributions (mixture of point mass at zero and a normal density) for the entries of $B$ and the off-diagonal entries of $\Omega$, and exponential priors for the diagonal entries of $\Omega$. Specifically, for $B = ((b_{jk}))$ and $\Omega = ((\omega_{ij}))$, we use the following priors:
\( b_{rs} \sim (1 - q_1)\delta_0 + q_1 N(0, \tau_1^2), \ 1 \leq r \leq p, \)
\[ 1 \leq s \leq q, \]
\( w_{st} \sim (1 - q_2)\delta_0 + q_2 N(0, \tau_2^2), \ 1 \leq s < t \leq q, \)
\( \omega_{ts} \sim \lambda \exp(-\lambda \omega_{ts}), \ 1 \leq s \leq q, \)

where \( b_{rs} \)'s and \( \omega_{rs} \)'s are independently distributed and \( \delta_0 \) denotes the distribution with its entire mass at 0. Further, the hyperparameters \( q_1, q_2 \in (0, 1) \) denote the respective mixing probabilities for entries of \( B \) and \( \Omega \), and the hyperparameters \( \tau_1^2, \tau_2^2 \) are the respective prior slab variances.

The resulting generalized posterior distribution \( \pi_{R, \text{joint}} \) is intractable in the sense that closed-form computation or direct sampling is not feasible. However, straightforward calculations show that:

- the full conditional posterior distribution of each entry of \( B \) (given all the other parameters and the data) is a mixture of a point mass at zero and an appropriate normal density. For \( 1 \leq r \leq p \), \( 1 \leq s \leq q \),

\[ (b_{rs}|Y, B_{-(rs)}, \Omega) \sim (1 - q_1^*)\delta_0 + q_1^* N \left( \frac{C_2}{C_1}, \frac{1}{C_1} \right) \]

where

\[ 1 - q_1^* = C_0(1 - q_1), \]
\[ C_0 = \left[ (1 - q_1) + \frac{q_1}{\tau_1 \sqrt{C_1}} \exp \left( \frac{C_2^2}{2C_1} \right) \right]^{-1}, \]
\[ C_1 = \sum_{k=1}^{q} \sum_{l=1}^{n} \omega_{lk} x_{ir}^2 + \frac{1}{\tau_2^2}, \]
\[ C_2 = \sum_{k=1}^{q} \sum_{l=1}^{n} \omega_{lk} \left( \sum_{l=1}^{n} \omega_{lk} y_{il} \right) x_{ir} \]
\[ - \sum_{k=1}^{q} \omega_{lk} x_{ir} \left[ \sum_{l \neq s} B_j^T x_i \omega_{lk} + \omega_{lk} \sum_{j \neq r} b_{js} x_{ij} \right]. \]

- the full conditional posterior distribution of each off-diagonal entry of \( \Omega \) (given all the other parameters and the data) is a mixture of a point mass at zero and an appropriate normal density. For \( 1 \leq s < t \leq q \),

\[ (\omega_{st}|Y, \Omega_{-(st)}, B) \sim (1 - q_2^*)\delta_0 + q_2^* N \left( \frac{-D_2}{D_1}, \frac{1}{D_1} \right) \]

where

\[ 1 - q_2^* = D_0(1 - q_2), \]
\[ D_0 = \left[ (1 - q_2) + \frac{q_2}{\tau_2 \sqrt{D_1}} \exp \left( \frac{D_2^2}{2D_1} \right) \right]^{-1}, \]
\[ D_1 = S_{ss} + S_{tt} + \frac{1}{\tau_2^2}, \]
\[ D_2 = \sum_{l \neq s} \omega_{ts} S_{ls} + \sum_{l \neq t} \omega_{st} S_{lt}, \]
\[ S = (Y - XB)^T (Y - XB). \]

The full conditional posterior density of each diagonal entry of \( \Omega \) (given all the other parameters and the data) is given as:

\[ \pi_{R, \text{joint}}(\omega_{ss}|Y, \Omega_{-(ss)}, B) \propto \omega_{ss}^{n} \exp \left\{ -\frac{1}{2} S_{ss}(\omega_{ss} + (\lambda + n_1 \epsilon_{is} f_{is})/S_{ss})^2 \right\} \]

where

\[ \epsilon_{is} = y_{is} - B_j^T x_i \] and \( f_{is} = \sum_{l \neq s} \omega_{ls} \epsilon_{il}. \)

This is an univariate density with the unique mode at

\[ \frac{\sqrt{(f_{s}(\lambda))^2 + 4n S_{ss} - f_{s}(\lambda)}}{2S_{ss}}. \]

where \( f_{s}(\lambda) = \sum_{i=1}^{n} \epsilon_{is} f_{is} + \lambda. \)

These properties allow us to construct a Metropolis-within-Gibbs sampler, which we call the joint regression network selector (JRNS), to sample from the joint generalized posterior density of \( B \) and \( \Omega \). One iteration of JRNS, given the current value of \( (B, \Omega) \), is described in Algorithm 1. Essentially, all entries of \( B \) and off-diagonal entries of \( \Omega \) are sampled from the respective full conditional distribution. A Metropolis-Hastings approach is used for the diagonal entries \( \omega_{ss} \) of \( \Omega \). In particular, a proposal is generated from a normal density centered at the conditional posterior mode for \( \omega_{ss} \). The proposed value is accepted or rejected based on the relevant Metropolis-based acceptance probability computed using the proposal normal density and the full conditional of \( \omega_{ss} \). A more detailed description of this algorithm is presented in Sect. 3 of the Supplement.
Algorithm 1 Joint Regression Network Selector

```plaintext
procedure JRNS(B, Ω, X, Y)
    for r = 1 to p do
        for s = 1 to q do
            Sample b_{jk} from the mixture distribution, \( (1 - q_s^j) \theta_0 + q_s^j N(C_j, C_k) \)
        end for
        \( E = (Y - XB) \)
        \( S = E^T E \)
        for \( s = 1 \) to \( q - 1 \) do
            for \( t = s + 1 \) to \( q \) do
                Sample \( \omega_{st} \) from the mixture distribution, \( (1 - q_s^t) \theta_0 + q_s^t N(-D_p, \frac{1}{p}) \)
            end for
            for \( s = 1 \) to \( q \) do
                Set \( D_1, D_2, q_s^2 \)
                Sample \( \omega_{qs} \) from the mixture distribution, \( (1 - q_s^2) \theta_0 + q_s^2 N(-D_p, \frac{1}{p}) \)
            end for
            if \( f_s(\pi(\omega_s Y, \Omega, B)) \) and compute mode
            Calculate acceptance probability, \( \rho \)
            Accept proposed value, \( v \) with probability \( \rho \)
            end for
        end for
    end for
    return \( B \)
    return \( \Omega \)
end procedure
```

2.1 Sparsity selection and estimation using MCMC output

The output \((B^{(i)}, \Omega^{(i)})_{i=1}^M\) from JRNS for an appropriate number \(M\) of iterations (after the burn-in period), can be used as follows to estimate the sparsity patterns in the corresponding parameters. We follow the majority voting approach to construct such an estimate, wherein we include only those variables whose generalized posterior-based marginal inclusion probabilities are at least 1/2 (Barbieri and Berger (2004)). Let \( \gamma_{jk} = I(b_{jk} \neq 0) \) represent the sparsity indicator of \( b_{jk} \) for \( j = 1, 2, \ldots, p \) and \( k = 1, 2, \ldots, q \). Then, \( \gamma = (\gamma_{jk}) \) represents the sparsity pattern in \( B \). For each \((j, k)\), let \( \hat{\pi}_{jk} = P(b_{jk} \neq 0 | Y) \) which is approximated by:

\[
\hat{\pi}_{jk} = \frac{1}{M} \sum_{i=1}^{M} I(b_{jk}^{(i)} \neq 0). \tag{8}
\]

The quantity in (8) is the proportion of iterations for which \( b_{jk}^{(i)} \neq 0 \) out of the \( M \) iterations. If \( \hat{\pi}_{jk} \geq 1/2 \), the \((j, k)\)-th entry is considered nonzero in the estimated sparsity pattern of \( B \). It is to be noted that by ergodic theorem the MCMC approximation given above in (8) converges to the generalized posterior probability, \( \hat{\pi}_{jk} \) of \( b_{jk} \) being nonzero as \( M \to \infty \). For large values of \( M \), it is very close to \( \hat{\pi}_{jk} \).

Then, an estimate of \( \gamma_{jk} \) is approximately obtained as:

\[
\hat{\gamma}_{jk} = \begin{cases} 
1, & \text{if } \frac{1}{M} \sum_{i=1}^{M} I(b_{jk}^{(i)} \neq 0) \geq \frac{1}{2} \\
0, & \text{otherwise.}
\end{cases}
\]

A similar majority voting approach based on the generalized posterior-based marginal inclusion probabilities can be used to estimate the sparsity pattern in \( \Omega \). In particular, if \( \eta_{rs} = I(\omega_{rs} \neq 0) \) represents the sparsity indicator for \( \omega_{rs} \) for \( 1 \leq r < s \leq q \), then an estimate of \( \eta_{rs} \) is approximately obtained as:

\[
\hat{\eta}_{rs} = \begin{cases} 
1, & \text{if } \frac{1}{M} \sum_{i=1}^{M} I(\omega_{rs}^{(i)} \neq 0) \geq \frac{1}{2} \\
0, & \text{otherwise.}
\end{cases}
\]

Further, an estimate of the magnitudes of the selected nonzero entries of \( B \) can also be obtained as follows. If \( \hat{\gamma}_{jk} = 1 \), then

\[
\hat{b}_{jk} = \frac{\sum_{i=1}^{M} b_{jk}^{(i)} I(b_{jk}^{(i)} \neq 0)}{\sum_{i=1}^{M} I(b_{jk}^{(i)} \neq 0)}.
\]

An estimate of the magnitudes of the selected nonzero entries of \( \Omega \) can also be obtained similarly. As stated earlier, the positive definiteness constraint on \( \Omega \) is relaxed for faster sparsity selection. An examination of the output of JRNS for many of our simulation settings in Sect. 4 consistently revealed positive-definite \( \Omega \) iterates. However, there is no general guarantee that these iterates or the resulting estimate of \( \Omega \) will be positive definite.

If one wants to enforce positive-definiteness, it can be achieved through a post-processing step (see Lee et al. (2020)), which focuses on the induced posterior of \( h(\Omega) \), where

\[
h(\Omega) = \begin{cases} 
\Omega & \text{if } \epsilon_{\min}(\Omega) > \epsilon, \\
\Omega + (\epsilon - \epsilon_{\min}(\Omega))I_q & \text{if } \epsilon_{\min}(\Omega) \leq \epsilon.
\end{cases}
\]

for some suitably chosen \( \epsilon > 0 \).

Note that \( h(\Omega) \) is guaranteed to be positive definite and has the exact same off-diagonal entries as \( \Omega \). Hence, if \( (\Omega^{(i)})_{i=1}^{M} \) are the \( \Omega \) components of the iterates produced by the JRNS or stepwise algorithm, sparsity selection, inclusion probabilities and credible intervals for the off-diagonal entries are unchanged if one uses \( h(\Omega^{(i)})_{i=1}^{M} \) instead of \( (\Omega^{(i)})_{i=1}^{M} \). The transformation to \( h(\Omega) \) only affects the estimation of the diagonal entries \( \{\omega_{ss}\}_{s=1}^{q} \). This additional eigenvalue check for computing \( h(\Omega) \) takes \( O(q^3) \) computations and hence does not change the computational complexity of JRNS (see (10)) and marginally increases the wall-clock time (less than 5% in all our simulation settings).
Another approach to ensure positive definiteness is to use the refitting idea from the penalized sparsity selection literature (see, for example, Ma and Michailidis (2016)). The estimators generated from penalized sparsity selection methods often suffer from (magnitude) bias issues, and one way to fix this is to obtain a constrained MLE of the desired parameter (by restricting to the estimated sparsity pattern). Using this idea in our context, we compute the estimated sparsity pattern \( \hat{\eta} \) in \( \Omega \) and the regression coefficient matrix estimator \( \hat{B} \) from the MCMC output as described above. Now, we use the pseudo-errors (rows of \( Y - X \hat{B} \)) as approximate samples from a \( \mathcal{N}_q(0, \Omega^{-1}) \) distribution and use the glasso function in \( R \) to compute the constrained MLE of \( \Omega \) restricted to the sparsity pattern \( \hat{\eta} \).

**Hyperparameter selection:** Selecting hyperparameters is an important issue in any Bayesian approach. In sample-starved settings discussed in this paper, the choice of hyperparameters may have a significant impact on the resulting estimates (see Supplementary Sect. 4.2 for more illustrations or details). A standard approach to choose the hyperparameters will be to use cross-validation wherein we consider a grid of values for the hyperparameters and select the set of values based on the minimum prediction error. However, this method can be computationally expensive. If one does not have the computational resources or time to carry out the cross-validation technique, another approach is to make some sensible objective choices as discussed below.

For JRNS, we have the prior mixture probabilities \( q_1, q_2 \), the prior slab variances \( \tau^2_1 \) and \( \tau^2_2 \) and \( \lambda \) as hyperparameters. For \( q_1 \) and \( q_2 \), one can always consider a flat \( U(0, 1) \) prior in which case we will get a beta-update for \( q_1 \) and \( q_2 \) in every iteration of the Gibbs sampler. Another choice of \( q_1 \) and \( q_2 \), which is motivated by the theoretical results in this paper and also in Cao et al. (2019), Narisetty and He (2014), is to take \( q_1 = 1/p \) and \( q_2 = 1/q \). We use these choices in the simulation studies and obtain good results. For \( \tau^2_1 \) and \( \tau^2_2 \), one may choose values around 1, or for a more principled choice one may choose objective inverse-gamma priors with shape = \( 10^{-4} \) and rate = \( 10^{-8} \) as suggested in Wang (2012). These will result in straightforward inverse-gamma updates for \( \tau^2_1 \) and \( \tau^2_2 \) in each iteration. One may consider the Gamma prior with the same shape and rate values for \( \lambda \) as well.

### 2.2 JRNS and BANS: a computational cost comparison

Next, we discuss the computational cost associated with the proposed JRNS algorithm and compare it with the computational cost for the BANS algorithm in Ha et al. (2020b). The structural differences in the generalized likelihoods used by the two algorithms have been described in the discussion surrounding equations (4) and (5). As we describe below, there are also crucial differences between the two approaches at the computational level that lead to a significant difference in overall computational costs.

- Algorithm 1, as described in Sect. 3 of the supplementary document, provides the detailed pseudo-code for one iteration of the JRNS algorithm. The matrix multiplications in Lines 2 and 3 take at most \( O(pq^2 + q^3 \rho^2) \) operations (computation of \( X'X \) and \( X'Y \) needs to be done only once prior to starting the iterations, and hence is not included). For each of the \( pq \) repetitions of the dual for loops in Lines 4 and 5, the most expensive steps are the computation of \( C_2 \) in Line 12, which takes \( O(q) \) operations, and the update of the \( s^{th} \) row of \( M_2 \) in Line 20 which takes \( O(p) \) operations. The computational cost of all the other steps does not depend on \( n, p, q \) and involves \( O(1) \) operations in all. Hence, the overall cost of Lines 4 to 22 is at most

\[
\text{operations. The matrix multiplications in Lines 23 and 24 need } O(npq + q^3 \rho^2) \text{ operations. For each of the } \binom{q}{2} \text{ repetitions of the dual for loops in Lines 25 and 26, the most expensive step is the computation of } D_2 \text{ in Line 33, which takes } O(q) \text{ operations. The computational cost of all the other steps does not depend on } n, p, q \text{ and takes } O(1) \text{ operations in all. The update of the diagonal entry in Lines 42 to 46 takes } O(q) \text{ operations and is only repeated in the outer for loop. Also, the update of } \Omega^2 \text{ in Line 49 requires } O(q^3) \text{ operations. Hence, the overall cost of Lines 23 to 49 is at most}
\]

\[
\binom{q}{2} O(q) + q O(q) + q^3 = O(q^3)
\]

The overall cost of one iteration of the JRNS algorithm can be obtained by adding the values in (9) and (10). **Note that this is an upper bound, as the sparsity in B and \( \Omega \) can reduce the cost of many vector/matrix products in Algorithm 1.**

- The BANS algorithm (Ha et al. 2020b, Supplemental Sect. S3) uses a Metropolis-Hastings-based approach to do a neighborhood exploration in the graph spaces for the sparsity patterns in \( B \) and \( \Omega \). This algorithm is implemented in the ch.chaingraph function available on Github with Supplementary material for Ha et al. (2020b) In particular, the \( \Omega \) update in each iteration of the BANS algorithm cycles through each response variable and proposes an add-delete or swap operation among its current neighbors or non-neighbors. This proposal is accepted or rejected based on a Metropolis-Hastings-based probability. The nonzero entries in the appropriate rows of \( \Omega \) are then generated from relevant
multivariate normal distributions. To implement this procedure, the authors start by computing the inverse of a \( q \times q \) matrix (Line 73 of \texttt{chaingraph.R} in Ha et al. (2020a)). The inversion requires \( q^3 \) operations. Since this is done for all \( q \) response variables, the costs of these inversions add up to \( q^4 \) operations. There are, of course, additional costs to consider for the computation of the acceptance probability and multivariate normal sampling described previously, which requires more albeit smaller matrix inversions and matrix multiplications of its own. A similar approach and inversion of \( p \times p \) matrices for all \( p \) predictor variables are needed in the \( B \) update, which leads a computational cost of \( p^4 \) operations (Lines 169–173 of \texttt{chaingraph.R} in Ha et al. (2020a)). The overall computational cost for one iteration of the BANS algorithm is therefore of the order of \( p^4 + q^4 \).

The above analysis shows that each iteration of the JRNS algorithm is an order of magnitude faster than each iteration of the BANS algorithm. The multiple inversions in the BANS algorithm is an order of magnitude faster than each iteration of JRNS. As opposed to jointly estimating the sparsity pattern in \( B \) and \( \Omega \), called BANS-parallel, which has been constructed by ignoring the symmetry in \( \Omega \) to parallelize the computations for each row. While a similar parallel version can also be constructed for JRNS, we find that even without parallelization, JRNS is computationally faster than the BANS-parallel algorithm. For instance, in a simulation setting with \( n = 100, p = 30, q = 60 \), 3000 MCMC iterations take around 50 seconds for the BANS-parallel as opposed to 5 seconds for the regular JRNS algorithm. Also, it is well known from the vanilla graphical models literature (see, for example, Peng et al. (2009), Khare et al. (2015)) that this non-symmetric approach can lead to statistical inefficiencies, and we do not pursue it further.

## 3 Stepwise estimation of the sparsity patterns of \( B \) and \( \Omega \)

Next, we present a computationally faster alternative to the JRNS procedure. As opposed to jointly estimating the sparsity patterns of \( B \) and \( \Omega \) using the generalized likelihood in (5), we first estimate the sparsity pattern in \( B \) by looking at the \( q \) individual regressions inherent in the multivariate regression model (1) and use this to obtain an estimate of the sparsity pattern in \( \Omega \). We provide a detailed description below.

### Step 1: Estimating the sparsity pattern in \( B \) using individual regressions

Let \( y_{j} \) denote the \( j^{th} \) column of the response matrix \( Y \). \( B_{j} \) denote the \( j^{th} \) column of the regression coefficient matrix \( B \), and \( e_{j} \) denote the \( j^{th} \) column of the error matrix \( e \), for \( j = 1, 2, \ldots, q \). Note that \( y_{j} \) is the collection of the \( n \) observations for the \( j^{th} \) response variable, and the entries of \( e_{j} \) are i.i.d. \( N(0, \sigma_{j}^{2}) \), where \( \sigma_{j}^{2} \) is the \( j^{th} \) diagonal entry of \( \Omega^{-1} \). The multivariate regression model \( Y = XB + e \) in (1) can be equivalently represented as a collection of the \( q \) individual regressions

\[
y_{j} = XB_{j} + e_{j} \quad \text{for } j = 1, 2, \ldots, q.
\]

Clearly, the vectors \( y_{1}, y_{2}, \ldots, y_{q} \) are dependent, and this dependence is precisely captured by the precision matrix \( \Omega \). However, in this section, we will be agnostic to this dependence and consider a generalized likelihood for \( B, (\sigma_{j}^{2})_{j=1}^{q} \) based on the product of the marginal densities of the vectors \( y_{1}, y_{2}, \ldots, y_{q} \) as follows.

\[
\tilde{L}_{g, \text{individual}}(Y|X, B, (\sigma_{j}^{2})_{j=1}^{q}) = \prod_{j=1}^{q} \left( \frac{1}{2\pi \sigma_{j}^{2}} \right)^{n/2} \exp \left\{ -\frac{1}{2\sigma_{j}^{2}} (y_{j} - XB_{j})^{T} (y_{j} - XB_{j}) \right\}.
\]

We use spike-and-slab priors (mixture of point mass at zero and a normal density) for the entries of \( B = (b_{rs}) \), and inverse-gamma priors for \( (\sigma_{r}^{2})_{s=1}^{q} \). In particular for \( 1 \leq r \leq p, 1 \leq s \leq q \),

\[
b_{rs} \sim (1 - q_{1}) \delta_{0} + q_{1} N(0, \tau_{1}^{2} \sigma_{s}^{2}),
b_{rs}^{2} \sim \text{Inv-Gamma}(\alpha, \beta),
\]

where \( b_{rs}^{2} \)’s and \( \sigma_{s}^{2} \)’s are independently distributed and \( \delta_{0} \) denotes the distribution with a point mass at 0. Again, \( q_{1} \in (0, 1) \) is a hyperparameter denoting the mixing probability for the spike-and-slab priors. The resulting generalized posterior distribution (denoted by \( \pi_{g, \text{individual}} \)) is again intractable in the sense that closed-form computation or direct sampling is not feasible. However, straightforward calculations show that:

- the full conditional posterior distribution of each entry of \( B \) (given all the other parameters and the data) is a mixture of a point mass at zero and an appropriate normal density:

\[
(b_{rs}|Y, B_{-(rs)}, (\sigma_{r}^{2}), \ldots, \sigma_{q}^{2}) \sim (1 - q_{1}^{*}) \delta_{0} + q_{1}^{*} N \left( \frac{C_{2}}{C_{1}}, \frac{\sigma_{s}^{2}}{C_{1}} \right)
\]
where
\[ 1 - q^*_s = C_0(1 - q_1), \]
\[ C_0 = \left[ (1 - q_1) + \frac{q_1}{\tau_1 \sqrt{C_1}} \exp \left( \frac{C_2^2}{2\sigma_s^2 C_1} \right) \right]^{-1}, \]
\[ C_1 = \sum_{i=1}^{n} x_{ir}^2 + \frac{1}{\tau_1}, \]
\[ C_2 = \sum_{i=1}^{n} x_{ir} \left( y_i - \sum_{j \neq r} x_{ij} b_{js} \right). \]

- The full conditional posterior distribution of \( \sigma_s^2 \) (given all the other parameters and the data) is again inverse-gamma:
\[ (\sigma_s^2 | y_s, b_{1s}, \ldots, b_{ps}) \sim \text{Inv-Gamma} \left( \alpha^*, \beta^* \right), \quad (13) \]

where
\[ \alpha^* = \alpha + \frac{n + |B_s|}{2}, \]
\[ \beta^* = \beta + \frac{||y_s - XB_s||^2}{2} + \frac{B_s^T B_s}{2\tau_1}, \]
\[ |B_s| := \text{number of nonzero entries in } B_s. \]

These properties allow us to construct a Gibbs sampler to generate approximate samples from the generalized posterior distribution of \( (B, (\sigma_s^2)_{s=1}^q) \). We can construct an estimate \( \hat{y}_{\text{stepwise}} \) of the sparsity pattern in \( B \) using the majority voting approach similar to the one mentioned in Sect. 2.1. An estimate \( \hat{B} \) of \( B \) can also be obtained as follows. Let \( B^* \) be a \( p \times q \) matrix whose \( k \)-th column, \( B^*_k \), is given by the posterior mean
\[ E(B_k | y_k, Y) \]
which has a closed-form expression given in Sect. A.3 of the supplementary document. Our estimate \( \hat{B}_{\text{stepwise}} \) of \( B \) is obtained from \( B^* \), replacing \( y \) by its estimate \( \hat{y}_{\text{stepwise}} \). Alternatively, an estimate of \( B \) can also be obtained using the Gibbs output in a similar manner as done for the JRNS approach toward the end of Sect. 2.1. For notational simplicity, in the rest of the paper, we will simply write \( \hat{B} \) in place of \( \hat{B}_{\text{stepwise}} \).

Note that using the generalized likelihood denoted by \( L_{g,\text{individual}}(\hat{y}|\Omega) \) amounts to simultaneously and independently estimating \( q \) individual regressions with Gaussian errors. The Gibbs sampling approach in (Narisetty and He 2014, Sect. 7) for univariate regressions with spike-and-slab priors can potentially be used for each of the \( q \) regressions. However, this approach again relies on first making appropriate moves in the space of sparsity patterns and then drawing the regression coefficient vector from the relevant multivariate normal distribution. With settings where \( p \) and \( q \) both are large in mind, we prefer to avoid the multivariate normal draws and instead use univariate mixture normal updates for each entry of \( B \) as previously specified.

**Step 2: Estimating the sparsity pattern in \( \Omega \) using error estimates from Step 1.** Using the working estimate \( \hat{B} \) from Step 1, we construct error estimates
\[ \hat{e}_i = y_i - \hat{B}^T x_i \quad \text{for } i = 1, 2, \ldots, n. \]

Let \( \hat{e} = Y - X \hat{B} \) denote the \( n \times q \) matrix with \( i \)-th row given by \( \hat{e}_i \) for \( 1 \leq i \leq n \). We know that the true errors \( \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n \) are i.i.d. \( N(0, \Omega^{-1}) \). Using the estimated error \( \hat{e}_i \) as an approximation for the true error \( \varepsilon_i \), our task of estimating \( \Omega \) is now reduced to a sparse precision matrix estimation problem. For this purpose, we use the generalized regression-based likelihood for \( \Omega \) by replacing \( \varepsilon = Y - XB \) in (5) by \( \hat{e} \) as follows.
\[ L_{g,\text{omega}}(\hat{e}|\Omega) \]
\[ = \left( \prod_{j=1}^{q} (\omega_{jj})^{n_j} \right) \exp \left\{ -\sum_{j=1}^{q} \frac{1}{2} ||\hat{e}_{\Omega,j}||^2 \right\}. \quad (14) \]

We use spike-and-slab prior distributions (mixture of point mass at zero and a normal density) for the off-diagonal entries for \( \Omega \), and exponential priors for the diagonal entries of \( \Omega \). In particular for \( 1 \leq s < t \leq q \),
\[ \omega_{st} \sim (1 - q_2) \alpha_0 + q_2 N(0, \tau_2^2), \]
\[ \omega_{ss} \sim \lambda \exp (-\lambda \omega_{ss}), \quad \omega_{ss} > 0. \]

The resulting generalized posterior distribution is intractable in the sense that closed-form computation or direct sampling is not feasible. However, straightforward calculations show that the full conditional posterior distributions of the off-diagonal and diagonal elements of \( \Omega \) are exactly as in (6) and (7) with \( B \) replaced by \( \hat{B} \) as needed. These properties allow us to construct a Gibbs sampler to generate approximate samples from the generalized posterior distribution of \( \Omega \), which can further be used to construct an estimator \( \hat{\Omega}_{\text{stepwise}} \) of the sparsity pattern of \( \Omega \) using the majority voting approach in a similar manner as was done for \( B \) in Step 1 of Method 2.

The issue of hyperparameter selection is also important in this approach. In the stepwise approach, we have the inverse-gamma parameters \( \alpha \) and \( \beta \) from the prior on the diagonals of \( \Omega^{-1} \) in Step 1 along with the other hyperparameters considered for the JRNS approach, namely the prior mixture probabilities \( q_1, q_2 \), the prior slab variances \( \tau_1^2, \tau_2^2 \), \( q_2 \), and \( \lambda \). For the hyperparameters \( q_1, q_2, \tau_1^2, \tau_2^2 \), and \( \lambda \), similar choices
can be taken as in the JRNS algorithm. As for the prior distributions on the diagonals of $\Omega^{-1}$, one might consider the objective inverse-gamma priors with shape $= 10^{-4}$ and rate $= 10^{-8}$ as considered for $\tau_1^2$ and $\tau_2^2$.

### 3.1 High-dimensional selection consistency for the stepwise approach

We establish high-dimensional consistency of the stepwise procedure for the estimation of the sparsity patterns of $B$ and $\Omega$ described in Sect. 3. We will consider a high-dimensional setting, where the number of responses $q$ and the number of predictors $p$ vary with $n$. Under the true model, the response matrix $Y$ is obtained as

$$Y = XB_0 + \varepsilon,$$

or, equivalently,

$$y_i = B_0^T x_i + \varepsilon_i \quad \text{for } i = 1, 2, \ldots, n.$$

The predictor vectors $x_1, x_2, \ldots, x_n$ and the error vectors $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ are assumed to be i.i.d. $N_p(0, R_0)$ and i.i.d. $N_0(0, \Omega_0^{-1})$, respectively. Since both $p$ and $q$ grow with $n$, the true parameters $B_0, \Omega_0$ also change with $n$, but we suppress this dependence for ease of exposition and remind the reader of this dependence as needed. Let the indicator matrices $\iota$ and $\iota_x$ respectively, denote the sparsity patterns in $B_0$ and $\Omega_0$ respectively, and $P_0$ denote the probability measure underlying the true model. We define $v_{k}$ as the number of nonzero entries in $(\gamma_1)_k$, the $k^{th}$ column of $\gamma_1$, $k_n = \max_{1\leq k \leq q} v_k$, and $\delta_n = \sum_{k=1}^{q} \nu_k$. Under standard and mild regularity conditions on the eigenvalues of $R_0$ and the hyperparameters $q_1, q_2, \tau_1^2$ and $\tau_2^2$ (see Supplementary Sects. 1 and 2 for details), the following consistency result can be established in a regime where $pq$ essentially can grow sub-exponentially with $n$.

**Theorem 1** (Selection and Estimation Consistency of the generalized posterior) Suppose $\frac{\log(pq)}{n} \rightarrow 0$. Then,

(a) (Selection Consistency for $B$) Under Assumptions A1–A4 stated in Supplementary Sect. 1, the (sequence of) sparsity pattern estimates $\hat{y}_{\text{stepwise}}$ for $B$ obtained from the stepwise approach satisfy

$$P_0 \left( \hat{y}_{\text{stepwise}} = \iota \right) \rightarrow 1 \quad \text{as } n \rightarrow \infty.$$

(b) (Estimation Consistency for $B$) Under Assumptions A1–A4 stated in Supplementary Sect. 1, the pseudo-posterior distribution on $B$ concentrates around the truth at a rate of $\sqrt{\frac{\log(pq)}{n}}$ (in Frobenius norm). In particular,

$$E_0 \left[ \prod_{i} \left( \|B - B_0\|_F > K \sqrt{\frac{\log(pq)}{n}} \mid Y \right) \right]$$

converges to 0 as $n \rightarrow \infty$ for a large enough constant $K$.

(c) (Selection Consistency for $\Omega$) Under Assumptions A1–A4 and B1–B4 stated in Supplementary Sects. 1 and 2, the (sequence of) sparsity pattern estimates $\hat{y}_{\text{stepwise}}$ for $\Omega$ obtained from the stepwise approach satisfy

$$P_0 \left( \hat{y}_{\text{stepwise}} = \iota \right) \rightarrow 1 \quad \text{as } n \rightarrow \infty.$$

The proof of the above results leverages arguments in Narisetty and He (2014) and Khare et al. (2015) for univariate spike-and-slab regression and standard graphical models with no covariates. However, some careful modifications and additional arguments are needed for the multivariate setting and the fact that pseudo-errors with an estimate $\hat{B}$ are being used in Step 2 of the stepwise approach. The proof is provided in Supplementary Sects. 1 and 2.

### 4 Performance evaluation

We evaluate the performance of the joint JRNS approach presented in Sect. 2 and the stepwise approach in Sect. 3 under diverse simulation settings. The data-generating model is $Y = XB_0 + \varepsilon$, where the $n$ rows of the error matrix $\varepsilon$ are i.i.d. multivariate normal with mean vector $0$ and precision matrix $\Omega_0$. We consider six different combinations of the triplet $(n, p, q)$ along with the number of nonzero entries in $B_0$ and the off-diagonal part of $\Omega_0$ provided in Table 1. For each combination, the rows of $X$ are independently generated according to $\mathcal{N}_p(0, R_0)$, where $R_0 = (0.7)^{1-k}I_{k=1}^{1}$. The nonzero entries of $B_0$ are drawn independently from a $U(1, 2)$ distribution. Further, the nonzero entries of the off-diagonals of $\Omega_0$ are drawn independently from $U((-1, -0.5) \cup (0.5, 1))$, while diagonal entries are drawn independently from a $U(1, 2)$ distribution.

For each simulation setting in Table 1, we generate 200 replicated data sets to evaluate the computational performance and selection accuracy with respect to $B$ and $\Omega$ of the proposed methods along with state-of-the-art Bayesian methods. Specifically, we compare the following methods: joint (JRNS algorithm in Sect. 2), stepwise (stepwise algorithm in Sect. 3), BANS (Bayesian node-wise selection algorithm from Ha et al. (2020b)), DPE (spike-and-slab lasso with dynamic posterior exploration from Deshpande et al. (2019)), DCPE (spike-and-slab lasso with dynamic conditional posterior exploration from Deshpande et al. (2019)) and HS-GHS (horseshoe-graphical horseshoe) from Li et al.
Table 1 Six different simulation settings with different \((n, p, q)\) combinations and number of true nonzero entries

| Combination | \((n, p, q)\) | Nonzeros in \(B_0\) | Nonzeros in \(\Omega_0\) (off-diagonals) |
|-------------|-------------|-----------------|------------------|
| 1           | (100, 30, 60) | \(p/5\)         | \(q/5\)          |
| 2           | (100, 60, 30) | \(p/5\)         | \(q/5\)          |
| 3           | (150, 200, 200) | \(p/5\)     | \(q/5\)          |
| 4           | (150, 300, 300) | \(p/5\)    | \(q/5\)          |
| 5           | (100, 200, 200) | \(p/30\)        | \(q/5\)          |
| 6           | (200, 200, 200) | \(p/30\)        | \(q/5\)          |

(2021). Note that any estimator obtained by maximizing a penalized likelihood can be interpreted as the posterior mode of an appropriate Bayesian model. The DPE and DCPE estimators are essentially penalized likelihood estimators obtained by using spike and (Laplace) slab penalties for individual entries of \(B\) and \(\Omega\). In detailed simulations in Deshpande et al. (2019), these methods are shown to provide significantly superior selection performance than the other penalized likelihood approaches such as MRCE Rothman et al. (2010) and CAPME Cai et al. (2013), and we use them here as benchmarks for the selection performance of the proposed methods. Of course, these optimization-based approaches do not generate samples from the posterior distribution and cannot provide uncertainty quantification in the form of posterior credible intervals/inclusion probabilities. The HS-GHS method of Li et al. (2021) is a fully Bayesian approach based on the Gaussian likelihood.

The joint and stepwise methods were both run for 1000 burn-in iterations and then 2000 more follow-up iterations. The hyperparameters were chosen as described toward the end of Sect. 2.1 (theoretically motivated choices for \(q_1\) and \(q_2\) and objective inverse-gamma priors for \(\tau_1^2\) and \(\tau_2^2\)). We also consider learning \(q_1\) and \(q_2\) adaptively by using Beta hyperpriors on \(q_1\) and \(q_2\), and the results are presented in Tables 8 and 9 of Supplementary Sect. 4.2. We use traceplots and cumulative average plots to monitor and ensure the convergence of the MCMC. Some of these plots are provided in Figs. 1 and 2. The BANS algorithm was run using the default
hyperparameter settings in Ha et al. (2020b) again with 1000 burn-in and 2000 more follow-up iterations. DPE and DCPE are optimization algorithms for identifying the relevant posterior mode, and they were run with default settings provided in Deshpande et al. (2019).

**Computational performance:** We start by evaluating the computational performance of each of the methods. For each of the six combinations from Table 1, we run each of these methods for the 200 replicated data sets and report the average wall clock time. All methods were run on HiPerGator, a high-performance computing cluster at the University of Florida with each node running an Intel Haswell E5-2698 processor. The average wall clock-times are reported in Table 2. The results demonstrate the challenges with scalability for BANS due to the matrix inversion issues discussed in Sect. 2.2. For the four combinations of \((n, p, q)\) with at least 200 predictors and responses, the required 3000 iterations of the MCMC algorithm for a single replication could not be completed in 4 days (mostly, less than 100 iterations were completed). The HS-GHS method also encountered similar issues. For Settings 3, 4, 5 and 6 (where \(p \) and \(q \) are both at least 200), the HS-GHS could not complete the required number of iterations in 4 days. In fact, for all these settings less than 150 iterations were completed in 4 days. For all 200 replications in Setting 1 we get an error involving positive definiteness of an intermediate matrix calculation. Hence, results are only provided for Setting 2. On the other hand, we see that both JRNS and stepwise approaches scale well and can easily handle settings with large \(p \) and \(q \) values. As expected, the stepwise approach takes less computing time than the joint approach. While the DPE algorithm also has scalability issues with increasing \(p \) and \(q \), the DCPE algorithm scales very well and is the fastest among all the five algorithms in most settings. In the \(p = q = 300 \) setting, the stepwise algorithm is faster, and the joint (JRNS) algorithm also roughly takes the same time as DCPE. However, as noted in Deshpande et al. (2019), the faster speed of DCPE can come at the cost of sub-optimal performance (see also Table 3). More importantly, the DCPE algorithm focuses on optimization of the posterior mode and does not provide samples from the posterior distribution for uncertainty quantification. On the other hand, output from the joint (JRNS) and stepwise methods can be used to construct posterior marginal inclusion probabilities and credible intervals. This is demonstrated in Tables 7, 8 and 9 for the JRNS method.

**Sparsity selection performance:** To assess the sparsity selection performance of the methods developed, the following measures were evaluated after running each method.

---

Fig. 2 Cumulative average plots for a few randomly selected entries in \(B \) when \((n, p, q) = (150, 300, 300)\) over the total 3000 Gibbs sampling iterations of the JRNS algorithm. The coordinates selected are \((162,37), (14,295), (231,151), (299,102), (162,277), (98,102)\). The black bold line represents the corresponding true value in \(B_0\). These plots illustrate that the MCMC cumulative averages are converging to the respective posterior means, and these posterior means are very close to the corresponding true values in the data-generating model.
on each of the 200 replicates, and comparing the estimated sparsity patterns with the true sparsity pattern:

\[
\text{Sensitivity} = \frac{TP}{TP + FN}, \quad \text{Specificity} = \frac{TN}{TN + FP}
\]

\[
\text{Matthews Correlation Coefficient (MCC)} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}
\]

where TP, TN, FP, FN are the total number of true positive, true negative, false positive and false negative identifications made. The average MCC values for sparsity estimation in Ω and B for all methods across all combinations are provided in Tables 3 and 4, respectively.

It can be seen that for sparsity selection in Ω, the JRNS and stepwise approaches significantly outperform DPE and DCPE in most settings. On closer examination of the outputs, one of the reasons appears to be that in some cases the DPE and DCPE estimate Ω by a diagonal matrix, thus failing to identify all nonzero off-diagonal elements. The performance of BANS in the settings where results are available is quite sub-optimal compared to other approaches. For sparsity selection in B, the JRNS method gives the best performance. Here the performance of BANS improves compared to Ω sparsity selection, but remains sub-optimal compared to competing approaches. The computationally faster approximations DCPE and stepwise are in general less accurate than DPE and JRNS, respectively. We have results for HS-GHS only in Setting 2 (due to timeout issues discussed before), and its performance with respect to sparsity selection in that setting is comparable to other methods.

### Table 2: Average wall-clock time (in s) over 200 replications for different methods

| Density Setting | Cases | Joint | Stepwise | DPE | DCPE | BANS | HS-GHS |
|-----------------|-------|-------|----------|-----|------|------|--------|
| (p/5, q/5)      | 100   | 3.79  | 4.48     | 0.67| 0.05 | 5922.96 | 5811.68 |
| (p/30, q/5)     | 100   | 833.80| 280.29   | TO | TO  | TO   | TO     |
|                 | 200   | 294.20| 138.50   | 4956.46| 78.47 | TO | TO     |

‘TO’ is short for ‘Timeout’ which implies that the method could not complete the required number of iterations in 4 days. ‘PDE’ refers to an error caused by intermediate matrices not being positive definite (PD).

### Table 3: MCC values for sparsity selection in Ω averaged over 200 replicates for different methods

| Sparsity Setting | Cases | Joint | Stepwise | DPE | DCPE | BANS | HS-GHS |
|------------------|-------|-------|----------|-----|------|------|--------|
| (p/5, q/5)       | 100   | 0.783 | 0.778    | 0.593| 0.576| 0.374| PDE    |
| (p/30, q/5)      | 100   | 0.867 | 0.846    | 0.533| 0.571| TO   | TO     |
|                  | 200   | 0.969 | 0.968    | 0.959| 0.964| TO   | TO     |

‘TO’ is short for ‘Timeout’ which implies that the method could not complete the required number of iterations in 4 days. ‘PDE’ refers to an error caused by intermediate matrices not being positive definite (PD).

### Table 4: MCC values for sparsity selection in B averaged over 200 replicates for different methods

| Sparsity Setting | Cases | Joint | Stepwise | DPE | DCPE | BANS | HS-GHS |
|------------------|-------|-------|----------|-----|------|------|--------|
| (p/5, q/5)       | 100   | 1.000 | 1.000    | 1.000| 0.613| PDE  |        |
| (p/30, q/5)      | 100   | 0.991 | 0.961    | 0.950| 0.943| TO   | TO     |
|                  | 200   | 1.000 | 0.956    | 0.997| 0.924| TO   | TO     |

‘TO’ is short for ‘Timeout’ which implies that the method could not complete the required number of iterations in 4 days. ‘PDE’ refers to an error caused by intermediate matrices not being positive definite (PD).
Table 5  Relative estimation error for $B$ averaged over 200 replicates for different methods

| Sparsity   | Cases $(p, q)$ | Joint $\bar{E}$ | Joint Refitted $\bar{E}$ | Stepwise $\bar{E}$ | DPE $\bar{E}$ | DCPE $\bar{E}$ | BANS $\bar{E}$ | HS-GHS $\bar{E}$ |
|------------|----------------|----------------|---------------------------|--------------------|-------------|--------------|---------------|----------------|
| $(p/5,q/5)$| 100 $(30,60)$  | 0.0167         | 0.0169                    | 0.0169             | 0.0169      | 0.9323       | PDE           |               |
|            | 100 $(60,30)$  | 0.0269         | 0.0276                    | 0.0275             | 0.0277      | 0.9317       | 0.0308        |               |
|            | 150 $(200,200)$| 0.0154         | 0.0172                    | 0.0152             | 0.0153      | TO           | TO            |               |
|            | 150 $(300,300)$| 0.0038         | 0.0434                    | TO                 | 0.0140      | TO           | TO            |               |
| $(p/30,q/5)$| 100 $(200,200)$| 0.0043         | 0.0141                    | 0.0063             | 0.0089      | TO           | TO            |               |
|            | 200 $(200,200)$| 0.00350        | 0.0116                    | 0.0033             | 0.0109      | TO           | TO            |               |

‘TO’ is short for ‘Timeout’ which implies that the method could not complete the required number of iterations in 4 days. ‘PDE’ refers to an error caused by intermediate matrices not being positive definite (PD)

Table 6  Relative estimation error for $\Omega$ averaged over 200 replicates for different methods

| Sparsity   | Cases $(p, q)$ | Joint $\bar{E}$ | Joint Refitted $\bar{E}$ | Stepwise $\bar{E}$ | DPE $\bar{E}$ | DCPE $\bar{E}$ | BANS $\bar{E}$ | HS-GHS $\bar{E}$ |
|------------|----------------|----------------|---------------------------|--------------------|-------------|--------------|---------------|----------------|
| $(p/5,q/5)$| 100 $(30,60)$  | 0.2444         | 0.1794                    | 0.2361             | 0.2300      | 0.2271       | 1.0247        | PDE           |
|            | 100 $(60,30)$  | 0.2475         | 0.1874                    | 0.2389             | 0.2424      | 0.2538       | 1.0125        | 0.2180        |
|            | 150 $(200,200)$| 0.2197         | 0.1442                    | 0.2092             | 0.1429      | 0.1444       | TO            | TO            |
|            | 150 $(300,300)$| 0.2181         | 0.1424                    | 0.2306             | TO          | 0.1679       | TO            | TO            |
| $(p/30,q/5)$| 100 $(200,200)$| 0.2352         | 0.1854                    | 0.2262             | 0.2423      | 0.2320       | TO            | TO            |
|            | 200 $(200,200)$| 0.2209         | 0.1159                    | 0.2029             | 0.1129      | 0.1100       | TO            | TO            |

‘TO’ is short for ‘Timeout’ which implies that the method could not complete the required number of iterations in 4 days. ‘PDE’ refers to an error caused by intermediate matrices not being positive definite (PD)

Estimation performance: To assess the estimation performance of the proposed methods, we compute the relative estimation error of the final estimates of $B$ and $\Omega$ which have been constructed using the majority voting approach described in Sect. 2.1. The relative estimation errors for $B$ are presented in Table 5 and those for $\Omega$ are presented in Table 6. As is seen from Table 5 the JRNS method performs very well in all the simulation settings, in fact it is the best-performing method in terms of relative estimation error of $B$ in most of the settings. The performance of the stepwise method is also quite competitive here. For estimation of $\Omega$, the transformation $h(\cdot)$ described at the end of Sect. 2.1 was used to ensure positive definiteness of the iterates and the resulting estimate. We have included an additional column in Table 6 for the refitted estimates of $\Omega$ as described in Sect. 2.1. It is evident from Table 6 that the performance of the JRNS method (without refitting) is competitive with the other methods. In Setting 2 where we were able to get HS-GHS output in a reasonable time, its performance is slightly better than JRNS, Stepwise, DPE and DCPE. The refitting-based $\Omega$ estimates for JRNS exhibit better performance than any other method in most of the settings including Setting 2. Note that for refitting-based estimates, the connection to the magnitudes of the entries in the $\Omega$ iterates of the JRNS MCMC output, and hence the corresponding credible intervals, is lost (for uncertainty quantification).

Uncertainty quantification based on the generalized posterior distribution: Next, we illustrate uncertainty quantification for JRNS using inclusion probabilities (see Sect. 2.1) and credible intervals obtained from the generalized posterior distribution. Note that the DPE and DCPE algorithms do not provide posterior samples for this purpose. We first consider the simulation setting where $n=(100, 200, 200)$ and randomly choose one out of the 200 replicated data sets. Table 7 shows the estimated marginal inclusion probabilities for selected entries in $B$ and $\Omega$ using the JRNS algorithm. For the matrix, $B$, entries $(47,4), (30,14), (181,43)$ are true positives: they are estimated as nonzero, since all have estimated inclusion probability 1 (the corresponding values were chosen as nonzero for all 2000 post burn-in iterations), and their true values in $B_0$ are nonzero. Entry $(78,84)$ is a false positive: it is estimated as nonzero since the estimated inclusion probability is $0.632 > 0.5$ (the corresponding values were chosen as nonzero for 1262 out of 2000 post burn-in iterations), but its true value in $B_0$ is zero. Hence, the inclusion probabilities indicate that the decision to classify $(78,84)$ as nonzero is not supported with the same certainty by the posterior distribution as the decision to classify $(47,4), (30,14), (181,43)$. Finally, entries $(67,5), (12,72)$ are true negatives: they are estimated as zero since the inclusion probabilities 0.005 and 0.0915 are less than 0.5, and their true values in $B_0$ are zero. For the $\Omega$ matrix, entries $(109,136), (30,32), (9,200)$ are true positives with esti-
Table 7 Illustration of classification based on marginal posterior inclusion probabilities using the joint (JRNS) method for selected entries of $B$ and $\Omega$ for the randomly chosen replication for $(n, p, q) = (100, 200, 200)$.

| Matrix | Entry | JRNS classification | Inclusion probability | True classification |
|--------|-------|----------------------|-----------------------|---------------------|
| $B$    | (47, 4) | Nonzero              | 1                     | Nonzero             |
| $B$    | (30, 14) | Nonzero             | 1                     | Nonzero             |
| $B$    | (181, 43) | Nonzero            | 1                     | Nonzero             |
| $B$    | (78, 84) | Nonzero              | 0.632                 | Zero                |
| $B$    | (67, 5) | Zero                | 0.005                 | Zero                |
| $B$    | (12, 72) | Zero               | 0.0915                | Zero                |
| $\Omega$ | (109, 136) | Nonzero         | 1                     | Nonzero             |
| $\Omega$ | (30, 32) | Nonzero            | 1                     | Nonzero             |
| $\Omega$ | (9, 200) | Nonzero            | 1                     | Nonzero             |
| $\Omega$ | (101, 122) | Nonzero         | 0.568                 | Zero                |
| $\Omega$ | (180, 2) | Zero              | 0                     | Zero                |
| $\Omega$ | (103, 13) | Zero              | 0                     | Zero                |

Fig. 3 Left: Network graph indicating associations between the predictors and the response variables for a randomly chosen replication when $(n, p, q) = (100, 200, 200)$. Right: Network graph indicating associations among the response variables for the same replication. The red circles represent the different predictor variables, and the blue squares represent the different response variables. The inclusion probabilities are mentioned on each edge. All the edge widths are proportional to the corresponding inclusion probabilities.

Estimated marginal inclusion probabilities 1, entry $(101, 122)$ is false positive with estimated marginal inclusion probability 0.568, and entries $(180, 2), (103, 13)$ are true negatives both with estimated marginal inclusion probabilities 0. The network plots indicating the associations between the predictors and the response variables and also among the response variables for this replication are presented in Fig. 3. Note that the BANS algorithm cannot provide a full set of iterations for the $(n, p, q) = (100, 200, 200)$ setting due to computational scalability issues. Hence, inclusion probabilities for the BANS algorithm are not included in Table 7.

For a comparative illustration with both JRNS and BANS, we consider the $(n, p, q) = (100, 30, 60)$ setting. Marginal inclusion probability estimates for selected entries of $B$ and $\Omega$ for both the joint (JRNS) method and the BANS approach (based on 2000 post burn-in iterations) are provided in Table 8. Entries $(30, 5), (25, 6)$ in $B$, and $(9, 47)$ in $\Omega$ are true positives for both methods (correctly identified as nonzero), but the inclusion probabilities for BANS are smaller than those of JRNS for all three entries. Entries $(21, 48)$ in $B$ and $(8, 31)$ in $\Omega$ are falsely identified as nonzero by BANS based on inclusion probabilities greater than 0.5, but correctly identified as zero by JRNS. Entry $(10, 40)$ in $\Omega$ is correctly identified as nonzero by JRNS with an inclusion probability of 1 while BANS incorrectly identifies it as zero with a low inclusion probability. Other entries in the table are true negatives for both methods (correctly identified as zero), but JRNS has a lower inclusion probability for all as...
compared to BANS. While the entries reported in Table 8 are just a small subset, we found that the pattern of JRNS having a higher/lower inclusion probability than BANS when the true value is nonzero/zero is repeated for most entries of $B$ and $\Omega$. This is not surprising given the significantly better selection performance of JRNS in this setting (see Tables 3 and 4).

Next, we consider the second simulation setting, where $(n, p, q) = (100, 60, 30)$, for a comparison of the empirical coverage probabilities of the 95% posterior credible intervals by JRNS and HS-GHS. For each of 12 true nonzero entries of $B$, and each of the 200 replications, we compute the 95% posterior credible interval obtained by using the relevant sample quantiles of the nonzero values in the 2000 post burn-in iterations (for both the methods). The proportion of credible intervals (out of 200) which contain the true value gives us an estimate of the coverage probability for each method. Table 9 presents the average coverage over the 200 replicated datasets of true value in the 95% credible intervals for these 12 entries of $B_0$. For $B$ entries, both methods perform very well with respect to including the true value in their corresponding credible intervals. The average coverage probability for JRNS is 0.948, while that for HS-GHS is 0.945. We also provide Fig. 4 for a visual comparison of the posterior credible intervals for the nonzero entries of the true $B$ in one of the replicates. The plot shows the credible intervals by both methods for all 12 nonzero values in the true $B$ for a single data set. In this particular data set, most of the credible intervals by JRNS are in general narrower than the corresponding credible intervals by HS-GHS, though the difference is relatively small. However, for co-ordinate (56,8) the HS-GHS credible interval fails to capture the true value, while the JRNS credible interval contains the true one. Sim-

![Fig. 4](image-url)
ilar patterns were observed in the credible intervals for other replicates.

We also obtain credible intervals for the three true nonzero entries of $\Omega$ in this setting and compute the coverage probability in a similar process as mentioned above for $B$. For a randomly selected replicate, we plot the credible intervals of the true nonzero entries of $\Omega$ by JRNS and HS-GHS in Fig. 5, and the average coverage probabilities are listed in Table 9. The credible intervals by JRNS are narrower; however, the comparison of coverage performance is mixed. Due to the narrower credible intervals of JRNS, the true value can sometimes lie just outside the credible interval and hence, the coverage probability gets negatively impacted by this.

We also obtain credible intervals for the three true nonzero entries of $\Omega$ in this setting and computed the coverage probability in a similar process as mentioned above for $B$. For a randomly selected replicate, we plot the credible intervals of true nonzero entries of $\Omega$ by JRNS and HS-GHS in Fig. 5, and the average coverage probabilities are listed in Table 9. The credible intervals by JRNS are narrower; however, the comparison of coverage performance is mixed. Due to the narrower credible intervals of JRNS, the true value can sometimes lie just outside the credible interval, and hence, the coverage probability gets negatively impacted by this.

We also consider a simulation setting with $(n, p, q) = (150, 300, 300)$ and select a group of entries in $B_0$ which are nonzero. The coverage probabilities for the 95% credible intervals, as described before, are estimated by the proportion of credible intervals (out of 200) containing the true value. The average coverage probability over all true nonzero entries in $B$ and over all 200 replications is 0.9422. Recall that the values for BANS and HS-GHS in this setting are not available due to computational scalability issues. Next, we present a comparison between the credible intervals obtained from JRNS and the frequentist confidence intervals obtained from the debiased lasso approach Van de Geer et al. (2014) in Fig. 6 for 7 randomly selected coordinates of $B$. The plot indicates that for all of these coordinates the JRNS approach provides narrower and more precise intervals while containing the corresponding true values for most of these coordinates. The codes implementing the two proposed methods, namely the JRNS and the stepwise methods, are available at https://github.com/srijata06/JRNS_Stepwise.

### 5 Analysis of TCGA cancer data

To further illustrate the performance of the proposed methods, we present results from the analysis of cancer data from TCGA (The Cancer Genome Atlas). We consider data for 7 different TCGA tumor types: colon adenocarcinoma (COAD), lung adenocarcinoma (LUAD), lung squamous cell carcinoma (LUSC), ovarian serous cystadenocarcinoma (OV), rectum adenocarcinoma (READ) skin cutaneous melanoma (SKCM) and uterine corpus endometrial carcinoma (UCEC). For each of these cancer types, we have mRNA expression data and RPPA-based proteomic data. As mentioned in the introduction, since mRNA is translated to protein, it is natural to consider protein expression data to be the response variable and the mRNA expression data to be the predictors. The sample size $(n)$, number of predictors $(p)$ and the number of response variables $(q)$ for the 7 data sets corresponding to each cancer type are given in Table 10.

| Cancer type | $n$  | $p$  | $q$  |
|-------------|-----|-----|-----|
| READ        | 121 | 73  | 76  |
| LUAD        | 356 | 73  | 76  |
| COAD        | 338 | 73  | 76  |
| LUSC        | 309 | 73  | 86  |
| OV          | 227 | 73  | 77  |
| SKCM        | 333 | 73  | 76  |
| UCEC        | 393 | 73  | 77  |
Fig. 7 LUAD networks with 0.5 as the inclusion probability cutoff. The circles represent genes, and the squares represent proteins. The different colors represent the different pathways listed in Table 14. Left: Network graph indicating associations between mRNA and protein. The inclusion probabilities are listed in Table 11. Right: Network graph indicating associations among proteins. The inclusion probabilities are listed in Table 10 in Supplementary document.

Fig. 8 A comparison of the coverage of credible interval by JRNS and confidence interval by debiased lasso for LUAD lung cancer for a few coordinates of $B$. The x-labels are in the form “gene-protein”.

We carry out a separate data analysis for each of the seven cancer types. For JRNS and the stepwise estimation methods, the Gibbs samplers were run for 1000 iterations for burn-in followed by additional 2000 iterations for calculating the regression coefficients and the precision matrices. As noted earlier, DPE and DCPE do not provide uncertainty quantification. While BANS does provide uncertainty quantification, computationally it takes a prohibitively long time with the above \((n, p, q)\) values. In Ha et al. (2020b), this dataset was analyzed, but the dataset for each cancer type was further broken based on pathway information, which significantly reduces the dimensionality of the problem.

We present the estimated network plots obtained using JRNS depicting (1) the associations between mRNA and pro-
teins and (2) that among the proteins for the LUAD cancer type in Fig. 7. The indices/serial numbers for genes and proteins for the LUAD dataset are given in Table 13 in Appendix. Figure 7 depicts the sparsity estimates of $B$ and $\Omega$ for the LUAD type cancer based on a 0.5 cutoff for the inclusion probabilities. The genes and proteins are mapped to their respective functional pathways to aid interpretation. The list of pathways and the corresponding genes for each of the pathways are listed in Table 14 in Appendix. For the associations encoded in matrix $B$ (see the left panel of Fig. 7), we see genes and proteins from the following pathways to be involved: RTK, EMT, Cell Cycle and Apoptosis. The results are broadly consistent with known functional mechanisms for the disease including stimulation of RTK to activate downstream signaling that encodes EMT’s inducing transcription factors Gonzalez and Medici (2014). The epithelial–mesenchymal transition (EMT) is an essential mechanism that contributes to the progression in cancer and involves apoptotic responses and the cell cycle, all elements captured in some of the connections depicted in the figure. Further, we see similar connections at the protein expression network in the right panel of Fig. 7. One can also see that there are strong connections within members of the same pathway, as well as cross-talk with members of other pathways. We particularly focus on the LUAD network plots here as it shows some very interesting biological connections. The network plots for the other cancer types are included in Supplementary file.

Next, we present Fig. 8 that depicts the coverage of the credible intervals by JRNS and the confidence intervals by debiased lasso for six randomly selected entries of $B$ for the lung adenocarcinoma (LUAD) cancer data. We randomly selected 6 gene-protein coordinates in $B$. Here the credible intervals are not only much shorter than the corresponding confidence intervals, but in most cases are subsets of their corresponding confidence intervals.

We also compare the prediction accuracy of the proposed methods with DPE and DCPE. Default settings were chosen for these methods as mentioned in Sect. 4. Results for HSGHS could not be obtained since we get the same error involving positive definiteness of an intermediate matrix calculation here as well. For prediction evaluation purposes, we perform a fivefold cross-validation in which we randomly divide the data set for each cancer type into 5 parts. The model for each of the listed approaches in Table 12 is built 5 times, each time using one of the parts as the test set and the rest as the training set. The average prediction error is then normalized with respect to that corresponding to the vanilla regression method ($q$ separate response-specific linear models). A relative prediction error less than 1 implies that the corresponding method has better prediction performance than the vanilla regression approach. All the relative prediction errors are listed in Table 12. The results show that the proposed methods have a very similar and compet-

### Table 11 Inclusion probability of each edge for the LUAD network graph indicating associations between mRNA and protein presented in Fig. 7

| Gene | Protein | Inclusion Probability |
|------|---------|-----------------------|
| 1    | X1      | 1                     | 0.98 |
| 2    | X2      | 2                     | 1.00 |
| 3    | X64     | 2                     | 0.88 |
| 4    | X60     | 3                     | 0.78 |
| 5    | X4      | 4                     | 1.00 |
| 6    | X32     | 4                     | 0.53 |
| 7    | X5      | 5                     | 1.00 |
| 8    | X16     | 5                     | 0.56 |
| 9    | X6      | 6                     | 1.00 |
| 10   | X7      | 7                     | 1.00 |
| 11   | X20     | 7                     | 0.98 |
| 12   | X8      | 8                     | 1.00 |
| 13   | X56     | 9                     | 0.96 |
| 14   | X60     | 9                     | 1.00 |
| 15   | X11     | 10                    | 1.00 |
| 16   | X11     | 11                    | 1.00 |
| 17   | X20     | 11                    | 0.76 |
| 18   | X12     | 12                    | 1.00 |
| 19   | X13     | 13                    | 1.00 |
| 20   | X14     | 13                    | 0.91 |
| 21   | X16     | 13                    | 0.63 |
| 22   | X70     | 14                    | 0.79 |
| 23   | X15     | 16                    | 1.00 |
| 24   | X40     | 16                    | 0.72 |
| 25   | X10     | 17                    | 0.94 |
| 26   | X16     | 17                    | 1.00 |
| 27   | X17     | 18                    | 1.00 |
| 28   | X68     | 18                    | 0.52 |
| 29   | X18     | 19                    | 1.00 |
| 30   | X11     | 20                    | 0.69 |
| 31   | X17     | 21                    | 0.72 |
| 32   | X21     | 21                    | 1.00 |
| 33   | X6      | 22                    | 0.86 |
| 34   | X21     | 24                    | 0.75 |
| 35   | X24     | 24                    | 1.00 |
| 36   | X23     | 25                    | 1.00 |
| 37   | X28     | 25                    | 0.81 |
| 38   | X63     | 25                    | 0.66 |
| 39   | X70     | 25                    | 0.97 |
| 40   | X10     | 26                    | 0.54 |
| 41   | X8      | 27                    | 0.60 |
| 42   | X11     | 27                    | 0.89 |
| 43   | X27     | 27                    | 1.00 |
| 44   | X28     | 28                    | 1.00 |
| 45   | X58     | 28                    | 0.98 |
Table 11 continued

| Gene | Protein | Inclusion Probability |
|------|---------|-----------------------|
| 46   | X29     | 29                    | 1.00 |
| 47   | X64     | 30                    | 0.89 |
| 48   | X17     | 31                    | 1.00 |
| 49   | X31     | 31                    | 1.00 |
| 50   | X32     | 31                    | 0.81 |
| 51   | X42     | 32                    | 0.99 |
| 52   | X34     | 34                    | 1.00 |
| 53   | X70     | 34                    | 0.83 |
| 54   | X33     | 35                    | 0.99 |
| 55   | X35     | 35                    | 1.00 |
| 56   | X36     | 37                    | 0.51 |
| 57   | X31     | 38                    | 0.58 |
| 58   | X37     | 38                    | 1.00 |
| 59   | X38     | 39                    | 1.00 |
| 60   | X25     | 40                    | 0.65 |
| 61   | X39     | 40                    | 1.00 |
| 62   | X8      | 41                    | 0.53 |
| 63   | X47     | 41                    | 1.00 |
| 64   | X24     | 43                    | 0.53 |
| 65   | X43     | 43                    | 1.00 |
| 66   | X44     | 44                    | 0.92 |
| 67   | X46     | 44                    | 0.94 |
| 68   | X11     | 47                    | 0.96 |
| 69   | X47     | 47                    | 1.00 |
| 70   | X36     | 49                    | 0.91 |
| 71   | X50     | 50                    | 0.71 |
| 72   | X6      | 52                    | 0.81 |
| 73   | X61     | 53                    | 1.00 |
| 74   | X68     | 53                    | 0.88 |
| 75   | X72     | 56                    | 0.96 |
| 76   | X58     | 57                    | 1.00 |
| 77   | X60     | 57                    | 1.00 |
| 78   | X15     | 58                    | 0.81 |
| 79   | X58     | 58                    | 1.00 |
| 80   | X28     | 59                    | 0.78 |
| 81   | X58     | 59                    | 1.00 |
| 82   | X59     | 59                    | 0.92 |
| 83   | X61     | 61                    | 0.51 |
| 84   | X6      | 62                    | 0.64 |
| 85   | X29     | 63                    | 0.98 |
| 86   | X62     | 63                    | 1.00 |
| 87   | X70     | 63                    | 1.00 |
| 88   | X63     | 64                    | 1.00 |
| 89   | X33     | 65                    | 0.90 |
| 90   | X63     | 65                    | 1.00 |
| 91   | X63     | 66                    | 1.00 |

Table 12 Relative prediction errors using fivefold cross-validation and normalized with respect to the vanilla linear regression approach for JRNS, stepwise method, DPE and DCPE

|                | JRNS | Stepwise | DPE  | DCPE |
|----------------|------|----------|------|------|
| READ           | 0.509| 0.513    | 0.516| 0.513|
| LUAD           | 0.878| 0.869    | 0.874| 0.876|
| LUSC           | 0.838| 0.839    | 0.834| 0.836|
| COAD           | 0.887| 0.881    | 0.886| 0.884|
| OV             | 0.778| 0.779    | 0.774| 0.779|
| SKCM           | 0.859| 0.860    | 0.859| 0.860|
| UCEC           | 0.919| 0.912    | 0.917| 0.919|

ative predictive performance compared to DPE and DCPE, while additionally providing uncertainty quantification by sampling from the posterior distribution.

6 Discussion

In this paper, we use a biconvex generalized likelihood function along with sparsity-inducing spike-and-slab prior distributions for joint sparsity selection and estimation of the regression coefficient and error covariance matrices in multivariate linear regression models. The proposed JRNS and stepwise algorithms are significantly faster than related (generalized) Bayesian methods both due to the simpler algebraic structure of the generalized likelihood used and also due to more efficient MCMC implementation (as discussed in Sect. 2.2), provide samples from the generalized posterior distribution for uncertainty quantification, and perform competitively in terms of selection/estimation performance.
in simulated data settings and in the TCGA cancer data application.

Intuitively, the joint JRNS approach should provide better accuracy than the stepwise approach as it utilizes the cross-correlations among the errors in estimation of $B$ (while the stepwise approach ignores them). This is borne out in the simulations, especially for Setting 4 with $p = q = 300$. However, theoretical analysis of the joint generalized posterior of $(B, \Omega)$ for the JRNS approach is much more complicated than the corresponding analysis for the stepwise approach. One possible direction of future enquiry is to establish high-dimensional posterior consistency results for the joint JRNS approach (analogous to those in Theorem 1 for the stepwise approach). Another possible future direction would be to explore the use of the biconvex generalized likelihood functions along with continuous shrinkage prior distributions, such as the Horseshoe one and study the computational and theoretical properties of such an approach.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11222-022-10102-5.

Acknowledgements The work of GM was supported in part by NIH grants 1U01CA235489-01 and 1R01GM114029-01A1.

Appendix: Pathways for TCGA cancer data

Table 13 lists the indices of all the genes and proteins in the LUAD cancer data, and Table 14 lists all the pathways that have been considered in the analysis of the TCGA cancer data in Sect. 5 and their gene members.

| Gene  | Protein   |
|-------|-----------|
| BAK1  | BAK       |
| BAX   | BAX       |
| BID   | BID       |
| BCL2L1| BIM       |
| CASP7 | CASPASE7CLEAVEDD198 |
| BAD   | BADPS112  |
| BCL2  | BCL2      |
| BCL2L1| BCLXL     |
| BIRC2 | CIAP      |
| CDK1  | CDK1      |
| CCNB1 | CYCLINB1  |
| CCNE1 | CYCLINE1  |
| CCNE2 | CYCLINE2  |
| CDKN1B| P27PT157  |
| PCNA  | P27PT198  |
| FOXM1 | FOXM1     |
| TP53BP1| 53BP1    |
| ATM   | ATM       |
| BRCA2 | ATM       |
| CHEK1 | CHK1PS345 |
| CHEK2 | CHK2PT68  |
| XRCC5 | KU80      |
| MRE11A| MRE11     |
| TP53  | P53       |
| RAD50 | RAD50     |
| RAD51 | RAD51     |
| XRCC1 | XRCC1     |
| FN1   | FIBRONECTIN |
| CDH2  | NCADHERIN |
| COL6A1| COLLAGENVI |
| CLDN7 | CLAUDIN7  |
| CDH1  | ECADHERIN |
| CTNNB1| BETACATENIN |
| SERPINE1| PAI1   |
| ESR1  | ERALPHA   |
| PGR   | ERALPHAPS118 |
| AR    | PR        |
| INPP4B| AR        |
| GATA3 | INPP4B    |
| AKT1  | GATA3     |
| AKT2  | AKTPS473  |
| AKT3  | AKTPS308  |
| GSK3A | GSK3ALPHABETAPS21S9 |
| GSK3B | GSK3PS9   |
| AKT1S1| PRAS40PT246 |
Table 13  continued

| Gene  | Protein          |
|-------|------------------|
| 46    | TSC2             | TUBERINPT1462 |
| 47    | PTEN             | PTEN          |
| 48    | ARAF             | ARAFPS299     |
| 49    | JUN              | CIUNPS73      |
| 50    | RAF1             | CRAFPS338     |
| 51    | MAPK8            | JNKPT183Y185  |
| 52    | MAPK1            | MAPKPT202Y204 |
| 53    | MAPK3            | MEK1PS217S221 |
| 54    | MAP2K1           | P38PT180Y182  |
| 55    | MAPK14           | P90RSKPT359S363 |
| 56    | RPS6KA1          | YB1PS102      |
| 57    | YBX1             | EGFRPY1068    |
| 58    | EGFR             | EGFRPY1173    |
| 59    | ERBB2            | HER2PY1248    |
| 60    | ERBB3            | HER3PY1298    |
| 61    | SHC1             | SHCPY317      |
| 62    | SRC              | SRCPY416      |
| 63    | EIF4EBP1         | SRCPY527      |
| 64    | RPS6KB1          | 4EBP1PS65     |
| 65    | MTOR             | 4EBP1PT37T46  |
| 66    | RPS6             | 4EBP1PT70     |
| 67    | RB1              | P70S6KPT389   |
| 68    | CAV1             | MTORPS2448    |
| 69    | MYH11            | S6PS235S236   |
| 70    | RAB11A           | S6PS240S244   |
| 71    | RAB11B           | RBPS807S811   |
| 72    | GAPDH            | CAVEOLIN1     |
| 73    | RBM15            | MYH11         |
| 74    | RAB11            | RAB11         |
| 75    | GAPDH            | GAPDH         |
| 76    | RBM15            | RBM15         |

Table 14  Pathways and gene membership

| Pathway          | Genes                                      |
|------------------|--------------------------------------------|
| 1 AKT/PI3K       | AKT1, AKT2, AKT3, GSK3A, GSK3B, CDKN1B, AKT1S1, TSC2, INPP4B, PTEN |
| 2 Apoptosis      | BAK1, BAX, BID, BCL2L11, CASP7, BAD, BCL2, BCL2L1, BIRC2 |
| 3 Breast Reactive| CAV1, MYH11, RAB11A, RAB11B, CTNNB1, GAPDH, RBM15 |
| 4 Cell Cycle     | CDK1, CCNB1, CCNE1, CCNE2, CDKN1B, PCNA, FOXM1 |
| 5 DNA damage response | TP53BP1, ATM, BRCA2, CHEK1, CHEK2, XRCC5, MRE11A, TP53, RAD50, RAD51, XRCC1 |
| 6 EMT            | FN1, CDH2, COL6A1, CLDN7, CDH1, CTNNB1, SERPINE1 |
| 7 Hormone Receptor | ES1, EGR, PR |
| 8 Hormone Signaling (Breast) | INPP4B, GATA3, BCL2 |
| 9 RAS            | ARAF, JUN, RAF1, MAPK8, MAPK1, MAPK3, MAPK14, RPS6KA1, YBX1 |
| 10 RTK           | EGFR, ERBB2, ERBB3, SHC1, SRC |
| 11 TSC           | EIF4EBP1, RPS6KB1, MTOR, RPS6, RB1 |
References

Alquier, P.: Approximate bayesian inference. Entropy (Basel) 22, 1272 (2020)
Barbieri, M.M., Berger, J.O.: Optimal predictive model selection. Ann. Stat. 32(3), 870–897 (2004)
Besag, J.: Statistical analysis of non-lattice data. J. R. Stat. Soc.: Ser. D (The Statistician) 24(3), 179–195 (1975)
Bhadra, A., Mallick, B.K.: Joint high-dimensional Bayesian variable and covariance selection with an application to eQTL analysis. Biometrics 69(2), 447–457 (2013)
Bissiri, P., Holmes, C., Walker, S.: A general framework for updating belief distributions. J. R. Stat. Soc. Ser. B 78, 1103–1130 (2016)
Brown, P.J., Vannucci, M., Fearn, T.: Multivariate Bayesian variable selection and prediction. J. R. Stat. Soc. Ser. B 60, 627–641 (1998)
Cai, T.T., Li, H., Liu, W., Xie, J.: Covariate-adjusted precision matrix estimation with an application in genetical genomics. Biometrika 100(1), 139–156 (2013)
Cao, X., Khare, K., Ghosh, M.: Posterior graph selection and estimation consistency for high-dimensional bayesian dag models. Ann. Stat. 47(1), 319–348 (2019)
Consonni, G., La Rocca, L., Peluso, S.: Objective Bayes covariate-adjusted sparse graphical model selection. Scand. J. Stat. 44, 741–764 (2017)
Deshpande, S.K., Roˇcková, V., George, E.I.: Simultaneous variable and covariance selection with the multivariate spike-and-slab lasso. J. Comput. Gr. Stat. 28(4), 921–931 (2019)
Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. Biostatistics 9(3), 432–441 (2008)
Van de Geer, S., Bühlmann, P., Ritov, Y., Dezeure, R.: On asymptotically optimal confidence regions and tests for high-dimensional models. Ann. Stat. 42(3), 1166–1202 (2014)
Gonzalez, D.M., Medici, D.: Signaling mechanisms of the epithelial-mesenchymal transition. Sci. Signal. 7(344), 8 (2014)
Ha, M.J., Stingo, F., Baladandayuthapani, V.: Supplemental material for ‘Bayesian Structure Learning in Multi-layered Genomic Networks’. Github (2020a)
Ha, M.J., Stingo, F.C., Baladandayuthapani, V.: Bayesian structure learning in multi-layered genomic networks. J. Am. Stat. Assoc. 1, 1–33 (2020b)
Khare, K., Oh, S., Rajaratnam, B.: A convex pseudolikelihood framework for high dimensional partial correlation estimation with convergence guarantees. J. R. Stat. Soc. B 77, 803–825 (2015)
Lee, K., Lee, K., Lee, J.: Post-processed posteriors for banded covariances. arXiv preprint arXiv:2011.12627 (2020)
Lee, W., Liu, Y.: Simultaneous multiple response regression and inverse covariance matrix estimation via penalized gaussian maximum likelihood. J. Multivar. Anal. 111, 241–255 (2012)
Li, Y., Datta, J., Craig, B.A., Bhadra, A.: Joint mean-covariance estimation via the horseshoe. J. Multivar. Anal. 183, 104716 (2021)
Lin, J., Basu, S., Banerjee, M., Michailidis, G.: Penalized maximum likelihood estimation of multi-layered gaussian graphical models. J. Mach. Learn. Res. 17, 1–51 (2016)
Lin, L., Drton, M., Shojaie, A.: High-dimensional inference of graphical models using regularized score matching. Electron. J. Stat. 10(1), 394–422 (2016)
Ma, J., Michailidis, G.: Joint structural estimation of multiple graphical models. J. Mach. Learn. Res. 17(166), 1–48 (2016)
McCarter, C., Kim, S.: (2014) On sparse gaussian chain graph models. In: Advances in Neural Information Processing Systems, pp. 3212–3220
Meinshausen, N., Bühlmann, P.: High-dimensional graphs and variable selection with the lasso. Ann. Stat. 34(1), 1436–1462 (2006)
Narisetty, N., He, X.: Bayesian variable selection with shrinking and diffusing priors. Ann. Stat. 42, 789–817 (2014)
Peng, J., Wang, P., Zhou, N., Zhu, J.: Partial correlation estimation by joint sparse regression models. J. Am. Stat. Assoc. 104, 735–746 (2009)
Richardson, S., Bottolo, L., Rosenthal, J.S.: Bayesian models for sparse regression analysis of high dimensional data. In: Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A.F.M., West, M. (Eds.) Bayesian Statistics 9 (2010)
Rothman, A.J., Levina, E., Zhu, J.: Sparse multivariate regression with covariance estimation. J. Comput. Gr. Stat. 19(4), 947–962 (2010)
Sohn, K.A., Kim, S.: (2012) Joint estimation of structured sparsity and output structure in multiple-output regression via inverse-covariance regularization. In: International Conference on Artificial Intelligence and Statistics, pp 1081–1089
Wang, H.: Bayesian graphical lasso models and efficient posterior computation. Bayesian Anal. 7(4), 867–886 (2012)
Yuan, X.T., Zhang, T.: Partial gaussian graphical model estimation. IEEE Trans. Inf. Theory 60(3), 1673–1687 (2014)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.