BayesSUR: An R Package for High-Dimensional Multivariate Bayesian Variable and Covariance Selection in Linear Regression

Zhi Zhao
University of Oslo

Marco Banterle
London School of Hygiene & Tropical Medicine

Leonardo Bottolo
University of Cambridge
Alan Turing Institute

Sylvia Richardson
University of Cambridge
Alan Turing Institute

Alex Lewin*
London School of Hygiene & Tropical Medicine

Manuela Zucknick*
University of Oslo

Abstract

In molecular biology, advances in high-throughput technologies have made it possible to study complex multivariate phenotypes and their simultaneous associations with high-dimensional genomic and other omics data, a problem that can be studied with high-dimensional multi-response regression, where the response variables are potentially highly correlated. To this purpose, we recently introduced several multivariate Bayesian variable and covariance selection models, e.g., Bayesian estimation methods for sparse seemingly unrelated regression for variable and covariance selection. Several variable selection priors have been implemented in this context, in particular the hotspot detection prior for latent variable inclusion indicators, which results in sparse variable selection for associations between predictors and multiple phenotypes. We also propose an alternative, which uses a Markov random field (MRF) prior for incorporating prior knowledge about the dependence structure of the inclusion indicators. Inference of Bayesian seemingly unrelated regression (SUR) by Markov chain Monte Carlo methods is made computationally feasible by factorization of the covariance matrix amongst the response variables.

In this paper we present **BayesSUR**, an R package, which allows the user to easily specify and run a range of different Bayesian SUR models, which have been implemented in C++ for computational efficiency. The R package allows the specification of the models in a modular way, where the user chooses the priors for variable selection and for covariance selection separately. We demonstrate the performance of sparse SUR models with the hotspot prior and spike-and-slab MRF prior on synthetic and real data sets representing eQTL or mQTL studies and *in vitro* anti-cancer drug screening studies as examples for typical applications.

Keywords: Seemingly unrelated regression, Bayesian multivariate regression, structured covariance matrix, Markov random field prior, multi-omics data.

1. Introduction

With the development of high-throughput technologies in molecular biology, the large-scale molecular characterization of biological samples has become common-place, for example by genome-wide measurement of gene expression, single nucleotide polymorphisms (SNP) or CpG methylation status. Other complex phenotypes, for example, pharmacological profiling from large-scale cancer drug screens, are also measured in order to guide personalized cancer therapies (Garnett, Edelman, Heidorn, Greenman, Dastur, Lau, Greninger, Thompson, Luo, Soares, Liu *et al.* 2012; Barretina, Caponigro, Stransky, Venkatesan, Margolin, Kim, Wilson, Lehar, Kryukov, Sonkin *et al.* 2012; Gray and Mills 2015). The analysis of joint associations between multiple correlated phenotypes and high-dimensional molecular features is challenging.

When multiple phenotypes and high-dimensional genomic information are jointly analyzed, the Bayesian framework allows to specify in a flexible manner the complex relationships between the highly structured data sets. Much work has been done in this area in recent years. Our software package **BayesSUR** (Banterle, Zhao, and Lewin 2021) gathers together several models that we have proposed for high-dimensional regression of multiple responses and also introduces a novel model, allowing for different priors for variable selection in the regression models and for different assumptions about the dependence structure between responses.

Bayesian variable selection uses latent indicator variables to explicitly add or remove predictors in each regression during the model search. Here, as we consider simultaneously many predictors and several responses, we have a matrix of variable selection indicators. Different variable selection priors have been proposed in the literature. For example, Jia and Xu (2007) mapped multiple phenotypes to genetic markers (i.e., expression quantitative trait loci, eQTL) using the spike-and-slab prior and hyper predictor-effect prior. Liquet, Mengersen, Pettitt, and Sutton (2017) incorporated group structures of multiple predictors via a (multivariate) spike-and-slab prior. The corresponding R (R Core Team 2021) package **MBSGS** (Liquet and Sutton 2017) is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=MBSGS`. Bottolo, Petretto, Blankenberg, Cambien, Cook, Tiret, and Richardson (2011) and Lewin, Saadi, Peters, Moreno-Moral, Lee, Smith, Petretto, Bottolo, and Richardson (2015b) further proposed the hotspot prior for variable selection in multivariate regression, in which the probability of association between the predictors and responses is decomposed multiplicatively into predictor and response random effects. This prior is implemented in a multivariate Bayesian hierarchical regression setup in the software **R2HESS** (Lewin, Campanella, Saadi, Liquet, and Chadeau-Hyam 2015a), available from `https://www.mrc-bsu.cam.ac.uk/software/`. Lee, Tadesse, Baccarelli, Schwartz, and Coull (2017) used the Markov random field (MRF) prior to encourage joint selection of the same variable across several correlated response variables. Their C-based R package **mBvs** (Lee, Tadesse, Coull, and Starr 2021) is available from CRAN (`https://CRAN.R-project.org/package=mBvs`).

For high-dimensional predictors and multivariate responses, the space of models is very large. To overcome the infeasibility of the enumerated model space for the MCMC samplers in the high-dimensional situation, Bottolo and Richardson (2010) proposed an evolutionary stochastic search (ESS) algorithm based on evolutionary Monte Carlo. This sampler has been extended in a number of situations and efficient implementation of ESS for multivariate

Bayesian hierarchical regression has been provided by the C++-based R package **R2GUESS** (Liquet, Bottolo, Campanella, Richardson, and Chadeau-Hyam 2016). Richardson, Bottolo, and Rosenthal (2011) proposed a new model and computationally efficient hierarchical evolutionary stochastic search algorithm (HESS) for multi-response (i.e., multivariate) regression which assumes independence between residuals across responses and is implemented in the **R2HESS** package. Petretto, Bottolo, Langley, Heinig, Mcdermott-Roe, Sarwar, Pravenec, Hubner, Aitman, Cook, and Richardson (2010) used the inverse Wishart prior on the covariance matrix of residuals in order to do simultaneous analysis of multiple response variables allowing for correlations in response residuals, for more moderate sized data sets.

In order to analyze larger numbers of response variables, yet retain the ability to estimate dependence structures between them, sparsity can be introduced into the residual covariances, as well as into the regression model selection. Holmes, Denison, and Mallick (2002) adapted seemingly unrelated regression (SUR) to the Bayesian framework and used a Markov chain Monte Carlo (MCMC) algorithm for the analytically intractable posterior inference. The hyper-inverse Wishart prior has been used to learn a sparser graph structure for the covariance matrix of high-dimensional variables (Carvalho, Massam, and West 2007; Wang 2010; Bhadra and Mallick 2013), thus performing covariance selection. However, these approaches are not computationally feasible if the number of input variables is very large. Bottolo, Banterle, Richardson, Ala-Korpela, Järvelin, and Lewin (2021) recently developed a Bayesian variable selection model which employs the hotspot prior for variable selection, learns a structured covariance matrix and implements the ESS algorithm in the SUR framework to further improve computational efficiency.

The **BayesSUR** package implements many of these possible choices for high-dimensional multi-response regressions by allowing the user to choose among three different prior structures for the residual covariance matrix and among three priors for the joint distribution of the variable selection indicators. This includes a novel model setup, where the MRF prior for incorporating prior knowledge about the dependence structure of the inclusion indicators is combined with Bayesian SUR models (Zhao, Banterle, Lewin, and Zucknick 2021). **BayesSUR** employs ESS as a basic variable selection algorithm.

2. Models specification

The **BayesSUR** package fits a Bayesian seemingly unrelated regression model with a number of options for variable selection, and where the covariance matrix structure is allowed to be diagonal, dense or sparse. It encompasses three classes of Bayesian multi-response linear regression models: hierarchical related regressions (HRR, Richardson *et al.* 2011), dense and sparse seemingly unrelated regressions (dSUR and SSUR, Bottolo *et al.* 2021), and the structured seemingly unrelated regression, which makes use of a Markov random field (MRF) prior (Zhao *et al.* 2021).

The regression model is written as

$$\begin{aligned} \mathbf{Y} &= \mathbf{X}\boldsymbol{B} + \mathbf{U}, \\ \text{vec}(\mathbf{U}) &\sim \mathcal{N}(\mathbf{0},\ C \otimes \mathbb{I}_n), \end{aligned} \tag{1}$$

where $\mathbf{Y}$ is a $n \times s$ matrix of outcome variables with $s \times s$ covariance matrix C, $\mathbf{X}$ is a $n \times p$ matrix of predictors for all outcomes, $\boldsymbol{B}$ is a $p \times s$ matrix of regression coefficients, $\mathbf{U}$ is the

	$\gamma_{jk} \sim$ Bernoulli	$\gamma_{jk} \sim$ Hotspot	$\boldsymbol{\gamma} \sim$ MRF
$C \sim$ indep	HRR-B	HRR-H	HRR-M
$C \sim \mathcal{IW}$	dSUR-B	dSUR-H	dSUR-M
$C \sim \mathcal{HIW}_{\mathcal{G}}$	SSUR-B	SSUR-H	SSUR-M

Table 1: Nine models across three priors of C by three priors of $\boldsymbol{\Gamma}$.

matrix of residuals, vec$(\cdot)$ indicates the vectorization of a matrix, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, $\mathbf{0}$ denotes a column vector with all elements zero, $\otimes$ is the Kronecker product and $\mathbb{I}_n$ is the identity matrix of size $n \times n$.

We use a binary latent indicator matrix $\boldsymbol{\Gamma} = \{\gamma_{jk}\}$ to perform variable selection. A spike-and-slab prior is used to find a sparse relevant subset of predictors that explain the variability of $\mathbf{Y}$: conditional on $\gamma_{jk} = 0$ ($j = 1, \cdots, p$ and $k = 1, \cdots, s$) we set $\beta_{jk} = 0$ and conditional on $\gamma_{jk} = 1$ regression coefficients follow a diffuse normal distribution:

$$\boldsymbol{\beta}_{\boldsymbol{\gamma}} \sim \mathcal{N}\left(\mathbf{0}, W_{\boldsymbol{\gamma}}^{-1}\right), \tag{2}$$

where $\boldsymbol{\beta} = \text{vec}(\boldsymbol{B})$, $\boldsymbol{\gamma} = \text{vec}(\boldsymbol{\Gamma})$, $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ consists of the selected regression coefficients only (i.e., where $\gamma_{jk} = 1$), and likewise $W_{\boldsymbol{\gamma}}$ is the sub-matrix of W formed by the corresponding selected coefficients.

The precision matrix W is generally decomposed into a shrinkage coefficient and a matrix that governs the covariance structure of the regression coefficients. Here we use $W = w^{-1}\mathbb{I}_{sp}$, meaning that all the regression coefficients are *a priori* independent, with an inverse gamma hyperprior on the shrinkage coefficient w, i.e., $w \sim \mathcal{IG}amma(a_w, b_w)$. The binary latent indicator matrix $\boldsymbol{\Gamma}$ has three possible options for priors: the independent hierarchical Bernoulli prior, the hotspot prior and the MRF prior. The covariance matrix C also has three possible options for priors: the independent inverse gamma prior, the inverse Wishart prior and hyper-inverse Wishart prior. Thus, we consider nine possible models (Table 1) across all combinations of three priors for C and three priors for $\boldsymbol{\Gamma}$.

2.1. Hierarchical related regression (HRR)

The hierarchical related regression model assumes that C is a diagonal matrix

$$C = \begin{pmatrix} \sigma_1^2 & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & \sigma_s^2 \end{pmatrix}, \tag{3}$$

which translates into conditional independence between the multiple response variables, so the likelihood factorizes across responses. An inverse gamma prior is specified for the residual covariance, i.e., $\sigma_k^2 \sim \mathcal{IG}amma(a_\sigma, b_\sigma)$ which, combined with the priors in (2) is conjugate with the likelihood in the model in (1). We can thus sample the variable selection structure $\boldsymbol{\Gamma}$ marginally with respect to C and $\boldsymbol{B}$. For inference for this model, Richardson *et al.* (2011) implemented the hierarchical evolutionary stochastic search algorithm (HESS).

HRR with independent Bernoulli prior

For a simple independent prior on the regression model selection, the binary latent indicators

follow a Bernoulli prior

$$\gamma_{jk}|\omega_{jk} \sim \mathcal{B}er(\omega_j), \quad j = 1, \cdots, p, \ k = 1, \cdots, s, \tag{4}$$

with a further hierarchical Beta prior on ω_j, i.e., $\omega_j \sim \mathcal{B}eta(a_\omega, b_\omega)$, which quantifies the probability for each predictor to be associated with any response variable.

HRR with hotspot prior

Richardson *et al.* (2011) and Bottolo *et al.* (2011) proposed decomposing the probability of association parameter ω_{jk} in (4) as $\omega_{jk} = o_k \times \pi_j$, where o_k accounts for the sparsity of each response model and π_j controls the propensity of each predictor to be associated with multiple responses simultaneously.

$$\begin{aligned} \gamma_{jk}|\omega_{jk} &\sim \mathcal{B}er(\omega_{jk}), \quad j = 1, \cdots, p, \ k = 1, \cdots, s, \\ \omega_{jk} &= o_k \times \pi_j, \\ o_k &\sim \mathcal{B}eta(a_o, b_o), \\ \pi_j &\sim \mathcal{G}amma(a_\pi, b_\pi). \end{aligned} \tag{5}$$

HRR with MRF prior

To consider the relationship between different predictors and associate highly correlated responses with the same predictors, we set a Markov random field prior on the latent binary vector $\boldsymbol{\gamma}$

$$f(\boldsymbol{\gamma}|d, e, G) \propto \exp\{d\mathbf{1}^\top\boldsymbol{\gamma} + e\boldsymbol{\gamma}^\top G\boldsymbol{\gamma}\}, \tag{6}$$

where G is an adjacency matrix containing prior information about similarities amongst the binary model selection indicators $\boldsymbol{\gamma} = \text{vec}(\boldsymbol{\Gamma})$. The parameters d and e are treated as fixed in the model. Alternative approaches include the use of a hyperprior on e (Stingo, Chen, Tadesse, and Vannucci 2011) or to fit the model repeatedly over a grid of values for these parameters, in order to detect the phase transition boundary for e (Lee *et al.* 2017) and to identify a sensible combination of d and e that corresponds to prior expectations of overall model sparsity and sparsity for the MRF graph.

2.2. Dense seemingly unrelated regression (dSUR)

The HRR models in Section 2.1 assume residual independence between any two response variables because of the diagonal matrix C in (3). It is possible to estimate a full covariance matrix by specifying an inverse Wishart prior, i.e., $C \sim \mathcal{IW}(\nu, \tau\mathbb{I}_s)$. To avoid estimating the dense and large covariance matrix directly, Bottolo *et al.* (2021) exploited a factorization of the dense covariance matrix to transform the parameter space (ν, τ) of the inverse Wishart distribution to space $\{\sigma_k^2, \rho_{kl}|\sigma_k^2 : k = 1, \cdots, s; l < k\}$, with priors

$$\begin{aligned} \sigma_k^2 &\sim \mathcal{IG}amma\left(\frac{\nu - s + 2k - 1}{2}, \frac{\tau}{2}\right), \\ \rho_{kl}|\sigma_k^2 &\sim \mathcal{N}\left(0, \frac{\sigma_k^2}{\tau}\right). \end{aligned} \tag{7}$$

Here, we assume that $\tau \sim \mathcal{G}amma(a_\tau, b_\tau)$. Thus, model (1) is rewritten as

$$\begin{aligned} \boldsymbol{y}_k &= \mathbf{X}\boldsymbol{\beta}_k + \sum_{l<k} \boldsymbol{u}_l \rho_{kl} + \boldsymbol{\epsilon}_k, \quad k = 1, \cdots, s, \\ \boldsymbol{\epsilon}_k &\sim \mathcal{N}(\mathbf{0},\ \sigma_k^2 \mathbb{I}_n), \end{aligned} \tag{8}$$

where $\boldsymbol{u}_l = \boldsymbol{y}_l - \mathbf{X}\boldsymbol{\beta}_l$ and $\boldsymbol{\beta}_l$ is the lth column of $\boldsymbol{B}$, so again the likelihood is factorized across responses.

Similarly to the HRR model, employing either the simple independence prior (4), the hotspot prior (5) or the MRF prior (6) for the indicator matrix $\boldsymbol{\Gamma}$ results in different sparsity specifications for the regression coefficients in the dSUR model. The marginal likelihood integrating out $\boldsymbol{B}$ is no longer available for this model, so joint sampling of $\boldsymbol{B}$, $\boldsymbol{\Gamma}$ and C is required. However, the reparameterization of the model (8) enables fast computation using the MCMC algorithm.

2.3. Sparse seemingly unrelated regression (SSUR)

Another approach to model the covariance matrix C is to specify a hyper-inverse Wishart prior, which means the multiple response variables have an underlying graph $\mathcal{G}$ encoding the conditional dependence structure between responses. In this setup, a sparse graph corresponds to a sparse precision matrix C^{-1}. From a computational point of view, it is infeasible to specify a hyper-inverse Wishart prior directly on C^{-1} in high dimensions (Carvalho *et al.* 2007; Jones, Carvalho, Dobra, Hans, Carter, and West 2005; Uhler, Lenkoski, and Richards 2018; Deshpande, Ročková, and George 2019). However, Bottolo *et al.* (2021) used a transformation of C to factorize the likelihood as in Equation 8. The hyper-inverse Wishart distribution, i.e., $C \sim \mathcal{HIW}_\mathcal{G}(\nu, \tau\mathbb{I}_s)$, becomes in the transformed variables the scalar variance σ_{qt}^2 and the associated correlation vector $\boldsymbol{\rho}_{qt} = (\rho_{1,qt}, \rho_{2,qt}, \cdots, \rho_{t-1,qt})^\top$ with

$$\begin{aligned} \sigma_{qt}^2 &\sim \mathcal{IG}amma\left(\frac{\nu - s + t + |S_q|}{2}, \frac{\tau}{2}\right), \quad q = 1, \cdots, Q, t = 1, \cdots, |R_q|, \\ \boldsymbol{\rho}_{qt}|\sigma_{qt}^2 &\sim \mathcal{N}\left(\mathbf{0}, \frac{\sigma_{qt}^2}{\tau}\mathbb{I}_{t-1}\right), \end{aligned} \tag{10}$$

where Q is the number of prime components in the decomposable graph $\mathcal{G}$, and S_q and R_q are the separators and residual components of $\mathcal{G}$, respectively. $|S_q|$ and $|R_q|$ denote the number of variables in these components. For more technical details, please refer to Bottolo *et al.* (2021).

As prior for the graph we use a Bernoulli prior with probability η on each edge $E_{kk'}$ of the graph as in

$$\begin{aligned} \mathbb{P}(E_{kk'} \in \mathcal{G}) &= \eta, \\ \eta &\sim \mathcal{B}eta(a_\eta, b_\eta). \end{aligned} \tag{11}$$

The three priors on $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$, i.e., independence (4), hotspot (5) and MRF (6) priors can be used in the SSUR model.

2.4. MCMC sampler and posterior inference

To sample from the posterior distribution, we use the evolutionary stochastic search algorithm (Bottolo and Richardson 2010; Bottolo *et al.* 2011; Lewin *et al.* 2015b), which uses a particular form of evolutionary Monte Carlo (EMC) introduced by Liang and Wong (2000). Multiple tempered Markov chains are run in parallel and both exchange and crossover moves are allowed between the chains to improve mixing between potentially different modes in the posterior. Note that we run multiple tempered chains at the same temperature instead of a ladder of different temperatures as was proposed in the original implementations of the (H)ESS sampler in Bottolo and Richardson (2010); Bottolo *et al.* (2011); Lewin *et al.* (2015b). The temperature is adapted during the burn-in phase of the MCMC sampling.

The main chain samples from the un-tempered posterior distribution, which is used for all inference. For each response variable, we use a Gibbs sampler to update the regression coefficients vector, $\boldsymbol{\beta}_k$ $(k = 1, \cdots, s)$, based on the conditional posterior corresponding to the specific model selected among the models presented in Sections 2.2 and 2.3. After L MCMC iterations, we obtain $\boldsymbol{B}^{(1)}, \cdots, \boldsymbol{B}^{(L)}$ and the estimate of the posterior mean is

$$\hat{\boldsymbol{B}} = \frac{1}{L-b} \sum_{t=b+1}^{L} \boldsymbol{B}^{(t)},$$

where b is the number of burn-in iterations. Posterior full conditionals are also available to update σ_k^2 and ρ_{kl} for the dSUR model and σ_{qt}^2 and $\boldsymbol{\rho}_{qt}$ for the SSUR model. In the HRR models in Section 2.1, the regression coefficients and residual covariances have been integrated out and therefore the MCMC output cannot be used directly for posterior inference of these parameters. However, for $\boldsymbol{B}$, the posterior distribution conditional on $\boldsymbol{\Gamma}$ can be derived analytically for the HRR models and this is the output for $\boldsymbol{B}$ that is provided in the **BayesSUR** package for HRR models.

At MCMC iteration t we also update each binary latent vector $\boldsymbol{\gamma}_k$ $(k = 1, \cdots, s)$ via a Metropolis-Hastings sampler, jointly proposing an update for the corresponding $\boldsymbol{\beta}_k$. After L iterations, using the binary matrices $\boldsymbol{\Gamma}^{(1)}, \cdots, \boldsymbol{\Gamma}^{(L)}$, the marginal posterior inclusion probabilities (mPIP) of the predictors are estimated by

$$\hat{\boldsymbol{\Gamma}} = \frac{1}{L-b} \sum_{t=b+1}^{L} \boldsymbol{\Gamma}^{(t)}.$$

In the SSUR models, another important parameter is $\mathcal{G}$ in the hyper-inverse Wishart prior for the covariance matrix C. It is updated by the junction tree sampler (Green and Thomas 2013; Bottolo *et al.* 2021) jointly with the corresponding proposal for σ_{qt}^2 and $\boldsymbol{\rho}_{qt}|\sigma_{qt}^2$ in (10). At each MCMC iteration we then extract the adjacency matrix $\mathcal{G}^{(t)}$ $(t = 1, \cdots, L)$, from which we derive posterior mean estimators of the edge inclusion probabilities as

$$\hat{\mathcal{G}} = \frac{1}{L-b} \sum_{t=b+1}^{L} \mathcal{G}^{(t)}.$$

Note that even though *a priori* the graph $\mathcal{G}$ is decomposable, the posterior mean estimate $\hat{\mathcal{G}}$ can be outside the space of decomposable models (see Bottolo *et al.* 2021).

The hyper-parameter τ in the inverse Wishart prior or hyper-inverse Wishart prior is updated by a random walk Metropolis-Hastings sampler. The hyper-parameter η and the variance w in the spike-and-slab prior are sampled from their posterior conditionals. For details see Bottolo *et al.* (2021).

3. The R package BayesSUR

The package **BayesSUR** is available from the Comprehensive R Archive Network (CRAN) at http://CRAN.R-project.org/package=BayesSUR. This article refers to version 2.0-1.

The main function is `BayesSUR()`, which has various arguments that can be used to specify the models introduced in Section 2, by setting the priors for the covariance matrix C and the binary latent indicator matrix $\mathbf{\Gamma}$. In addition, MCMC parameters (`nIter`, `burnin`, `nChains`) can also be defined. The following syntax example introduces the most important function arguments, which are further explained below. The full list of all arguments in function `BayesSUR()` is given in Table 2.

```
BayesSUR(data, Y, X, X_0, covariancePrior, gammaPrior,
  nIter, burnin, nChains, ...)
```

The data can be provided as a large combined numeric matrix $[\mathbf{Y}, \mathbf{X}, \mathbf{X}_0]$ of dimension $n \times (s + p)$ via the argument `data`; in that case the arguments `Y`, `X` and `X_0` need to contain the dimensions of the individual response variables $\mathbf{Y}$, predictors under selection $\mathbf{X}$ and fixed predictors $\mathbf{X}_0$ (i.e., mandatory predictors that will always be included in the model). Alternatively, it is also possible to supply $\mathbf{X}_0$, $\mathbf{X}$ and $\mathbf{Y}$ directly as numeric matrices via the arguments `X_0`, `X` and `Y`. In that case, argument `data` needs to be `NULL`, which is the default.

The arguments `covariancePrior` and `gammaPrior` specify the different models introduced in Section 2. When using the Markov random field prior (6) for the latent binary vector $\boldsymbol{\gamma}$, an additional argument `mrfG` is needed to assign the edge potentials; this can either be specified as a numeric matrix or as a file directory path leading to a text file with the corresponding information. For example, the HRR model with independent hierarchical prior in Section 2.1 is specified by `(covariancePrior = "IG", gammaPrior = "hierarchical")`, the dSUR model with hotspot prior in Section 2.2 by `(covariancePrior = "IW", gammaPrior = "hotspot")` and the SSUR model with MRF prior in Section 2.3 for example by `(covariancePrior = "HIW", gammaPrior = "MRF", mrfG = "mrfFile.txt")`.

The MCMC parameter arguments `nIter`, `burnin` and `nChains` indicate the total number of MCMC iterations, the number of iterations in the burn-in period and the number of parallel tempered chains in the evolutionary stochastic search MCMC algorithm, respectively. See Section 2.4 and, e.g., Bottolo and Richardson (2010) for more details on the ESS algorithm.

The main function `BayesSUR()` is used to fit the model. It returns an object of S3 class '`BayesSUR`' in a list format, which includes the input parameters and directory paths of output text files, so that other functions can retrieve the MCMC output from the output files, load them into R and further process the output for posterior inference of the model output.

In particular, a `summary()` function has been provided for '`BayesSUR`' class objects, which is used to summarize the output produced by `BayesSUR()`. For this purpose, a number of predictors are selected into the model by thresholding the posterior means of the latent indicator variables. By default, the threshold is 0.5, i.e., variable j is selected into the model for

Argument	Description
`data`	Combined numeric data matrix $[\mathbf{Y}, \mathbf{X}]$ or $[\mathbf{Y}, \mathbf{X}, \mathbf{X}_0]$. Default is `NULL`.
`Y`	Numeric matrix or indices with respect to the argument `data` for the responses.
`X`	Numeric matrix or indices with respect to the argument `data` for the predictors.
`X_0`	Numeric matrix or indices with respect to the argument `data` for predictors forced to be included (i.e., that are not part of the variable selection procedure). Default is `NULL`.
`outFilePath`	Directory path where the output files are saved.
`covariancePrior`	Prior for the covariance matrix; `"IG"`: independent inverse gamma prior, `"IW"`: inverse Wishart prior, `"HIW"`: hyper-inverse Wishart prior (default).
`gammaPrior`	Prior for the binary latent variable $\mathbf{\Gamma}$; `"hierarchical"`: Bernoulli prior, `"hotspot"`: hotspot prior (default), `"MRF"`: Markov random field prior.
`mrfG`	A numeric matrix or a path to the file containing (the edge list of) the G matrix for the MRF prior on $\mathbf{\Gamma}$. Default is `NULL`.
`nIter`	Total number of MCMC iterations.
`burnin`	Number of iterations in the burn-in period.
`nChains`	Number of parallel chains in the evolutionary stochastic search MCMC algorithm.
`gammaSampler`	Local move sampler for the binary latent variable $\mathbf{\Gamma}$, either (default) `"bandit"` for a Thompson-sampling inspired sampler or `"MC3"` for the usual MC^3 sampler.
`gammaInit`	$\mathbf{\Gamma}$ initialization to either all zeros (`"0"`), all ones (`"1"`), MLE-informed (`"MLE"`) or (default) randomly (`"R"`).
`hyperpar`	A list of named prior hyperparameters to use instead of the default values, including `a_w`, `b_w`, `a_sigma`, `b_sigma`, `a_omega`, `b_omega`, `a_o`, `b_o`, `a_pi`, `b_pi`, `nu`, `a_tau`, `b_tau`, `a_eta` and `b_eta`. They correspond to $w \sim \mathcal{IG}amma($`a_w`, `b_w`$)$, $\sigma_k^2 \sim \mathcal{IG}amma($`a_sigma`, `b_sigma`$)$, $\omega_j \sim \mathcal{B}eta($`a_omega`, `b_omega`$)$, $o_k \sim \mathcal{B}eta($`a_o`, `b_o`$)$, $\pi_j \sim \mathcal{G}amma($`a_pi`, `b_pi`$)$, $\nu=$`nu`, $\tau \sim \mathcal{G}amma($`a_tau`, `b_tau`$)$, $\eta \sim \mathcal{B}eta($`a_eta`, `b_eta`$)$. For default values see `help("BayesSUR")`.
`maxThreads`	Maximum threads used for parallelization. Default is 1.
`output_*`	Allow (`TRUE`) or suppress (`FALSE`) the output for *; possible outputs are $\mathbf{\Gamma}$, $\mathcal{G}$, $\boldsymbol{B}$, $\boldsymbol{\sigma}$, $\boldsymbol{\pi}$, tail (hotspot tail probability, see Bottolo and Richardson 2010) or `model_size`. Default is all: `TRUE`.
`tmpFolder`	The path to a temporary folder where intermediate data files are stored (will be erased at the end of the MCMC sampling). Defaults to local `tmpFolder`.

Table 2: Overview of the arguments in the main function `BayesSUR()`.

Function	Description
`BayesSUR()`	Main function of the package. Fits any of the models introduced in Section 2. Returns an object of S3 class '`BayesSUR`', which is a list which includes the input parameters (input) and directory paths of output text files (output), as well as the run status and function call.
`print()`	Print a short summary of the fitted model generated by `BayesSUR()`, which is an object of class '`BayesSUR`'.
`summary()`	Summarize the fitted model generated by `BayesSUR()`, which is an object of class '`BayesSUR`'.
`coef()`	Extract the posterior mean of the coefficients of a '`BayesSUR`' class object.
`fitted()`	Return the fitted response values that correspond to the posterior mean of the coefficients matrix of a '`BayesSUR`' class object.
`predict()`	Predict responses corresponding to the posterior mean of the coefficients, return posterior mean of coefficients or indices of non-zero coefficients of a '`BayesSUR`' class object.
`plot()`	Main plot function to be called by the user. Creates a selection of plots for a '`BayesSUR`' class object by calling one or several of the specific plot functions below as specified by the combination of the two arguments `estimator` and `type`.
`elpd()`	Measure the prediction accuracy by the expected log pointwise predictive density (elpd). The out-of-sample predictive fit can either be estimated by Bayesian leave-one-out cross-validation (LOO) or by widely applicable information criterion (WAIC, Vehtari *et al.* 2017). See Appendix A for details.
`getEstimator()`	Extract the posterior mean of the parameters (i.e., $\boldsymbol{B}$, $\boldsymbol{\Gamma}$ and $\mathcal{G}$) of a '`BayesSUR`' class object. Also, the log-likelihood of $\boldsymbol{\Gamma}$, model size and $\mathcal{G}$ can be extracted for the MCMC diagnostics.
`plotEstimator()`	Plot the estimated relationships between response variables and estimated coefficients of a '`BayesSUR`' class object with argument `estimator = c("beta", "gamma", "Gy")`.
`plotGraph()`	Plot the estimated graph for multiple response variables from a '`BayesSUR`' class object with argument `estimator = "Gy"`.
`plotNetwork()`	Plot the network representation of the associations between responses and predictors, based on the estimated $\hat{\boldsymbol{\Gamma}}$ matrix of a '`BayesSUR`' class object with argument `estimator = c("gamma", "Gy")`.
`plotManhattan()`	Plot Manhattan-like plots for marginal posterior inclusion probabilities (mPIP) and numbers of responses of association for predictors of a '`BayesSUR`' class object with argument `estimator = "gamma"`.
`plotMCMCdiag()`	Show trace plots and diagnostic density plots of a '`BayesSUR`' class object with argument `estimator = "logP"`.
`plotCPO()`	Plot the conditional predictive ordinate (CPO) for each individual of a fitted model generated by `BayesSUR()` with argument `estimator = "CPO"`. CPO is used to identify potential outliers (Gelfand 1996).

Table 3: Overview of the functions in package **BayesSUR**.

response k if $\hat{\gamma}_{jk} > 0.5$. The `summary()` function also outputs the quantiles of the conditional predictive ordinates (CPO, Gelfand 1996), top predictors with respect to average marginal posterior inclusion probabilities (mPIP) across all response variables and top response variables with respect to average mPIP across all predictors, expected log pointwise predictive density (i.e., `elpd.LOO` and `elpd.WAIC`, Vehtari *et al.* 2017), model specification parameters, MCMC running parameters and hyperparameters.

To use a specific estimator, the function `getEstimator()` is convenient to extract point estimates of the coefficients matrix $\hat{\boldsymbol{B}}$, latent indicator variable matrix $\hat{\boldsymbol{\Gamma}}$ or learned structure $\hat{\mathcal{G}}$ from the directory path of the model object. All point estimates are posterior means, thus $\hat{\gamma}_{jk}$ is the marginal posterior inclusion probability for variable j to be selected in the regression for response k, and $\hat{\mathcal{G}}_{kl}$ is the marginal posterior edge inclusion probability between responses k and l, i.e., the marginal posterior probability of conditional dependence between k and l. The regression coefficient estimates $\hat{\boldsymbol{B}}$ can be the marginal posterior means over all models, independently of $\hat{\boldsymbol{\Gamma}}$ (with default argument `beta.type = "marginal"`). Then, $\hat{\beta}_{jk}$ represents the shrunken estimate of the effect of variable j in the regression for response k. Alternatively, $\hat{\beta}_{jk}$ can be the posterior mean conditional on $\gamma_{jk} = 1$ with argument `beta.type = "conditional"`. If `beta.type = "conditional"` and `Pmax = 0.5` are chosen, then these conditional $\hat{\beta}_{jk}$ estimates correspond to the coefficients in a median probability model (Barbieri and Berger 2004).

In addition, the generic S3 methods `coef()`, `predict()`, and `fitted()` can be used to extract regression coefficients, predicted responses, or indices of non-zero coefficients, all corresponding to the posterior mean estimates of an '`BayesSUR`' object.

The main function for creating plots of a fitted BayesSUR model, is the generic S3 method `plot()`. It creates a selection of the above plots, which the user can specify via the `estimator` and `type` arguments. If both arguments are set to `NULL` (default), then all available plots are shown in an interactive manner. The main `plot()` function uses the following specific plot functions internally. These can also be called directly by the user. The function `plotEstimator()` visualizes the three estimators. To show the relationship of multiple response variables with each other, the function `plotGraph()` prints the structure graph based on $\hat{\mathcal{G}}$. Furthermore, the structure relations between multiple response variables and predictors can be shown via function `plotNetwork()`. The marginal posterior probabilities of individual predictors are illustrated via the `plotManhattan()` function, which also shows the number of associated response variables of each predictor.

Model fit can be investigated with `elpd()` and `plotCPO()`. `elpd()` estimates the expected log pointwise predictive density (Vehtari *et al.* 2017) to assess out-of-sample prediction accuracy. `plotCPO()` plots the conditional predictive ordinate for each individual, i.e., the leave-one-out cross-validation predictive density. CPOs are useful for identifying potential outliers (Gelfand 1996). To check convergence of the MCMC sampler, function `plotMCMCdiag()` prints traceplots and density plots for moving windows over the MCMC chains.

Table 3 lists all functions. **BayesSUR** uses the **Rcpp** (Eddelbuettel and François 2011) and **RcppArmadillo** (Eddelbuettel and Sanderson 2014) R packages to integrate C++ code with R. The **igraph** package (Csárdi and Nepusz 2006) is used for constructing the graph plots.

4. Quick start with a simple example

In the following example, we illustrate a simple simulation study where we run two models: the default model choice, which is an SSUR model with the hotspot prior, and in addition an SSUR model with the MRF prior. The purpose of the latter is to illustrate how we can construct an MRF prior graph. We simulate a data set $\mathbf{X}$ with dimensions $n \times p = 10 \times 15$, i.e., 10 observations and 15 input variables, a sparse coefficients matrix $\boldsymbol{B}$ with dimension $p \times s = 15 \times 3$, which creates associations between the input variables and $s = 3$ response variables, and random noise $\mathbf{E}$. The response matrix is generated by the linear model $\mathbf{Y} = \mathbf{X}\boldsymbol{B} + \mathbf{E}$.

```
R> set.seed(82193)
R> n <- 10; s <- 3; p <- 15
R> X <- matrix(rnorm(n * p, 2, 1), nrow = n)
R> B <- matrix(c(0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1,
+    rep(0, s * p - 12)), nrow = p, byrow = TRUE)
R> E <- matrix(rnorm(n * s, 0, 0.2), nrow = n)
R> Y <- X %*% B + E
```

Note that $\boldsymbol{B}$ is sparse and only the first four input variables have non-zero coefficients:

```
R> B
```

```
      [,1] [,2] [,3]
 [1,]    0    0    1
 [2,]    1    1    0
 [3,]    1    1    0
 [4,]    0    0    1
 [5,]    0    0    0
 [6,]    0    0    0
 [7,]    0    0    0
 [8,]    0    0    0
 [9,]    0    0    0
[10,]    0    0    0
[11,]    0    0    0
[12,]    0    0    0
[13,]    0    0    0
[14,]    0    0    0
[15,]    0    0    0
```

First, let us fit the default model. The default is to run two MCMC chains with 10000 iterations each, of which the first 5000 iterations are discarded as the burn-in period. The function `print()` returns a short summary of the results from the fitted model object, including the number of selected predictors by thresholding the marginal posterior inclusion probabilities (mPIP) at 0.5, and two measures of the model's prediction accuracy (i.e., `elpd.LOO` and `elpd.WAIC`).

```
R> library("BayesSUR")
R> library("tictoc")
R> tic("Time of model fitting")
R> set.seed(1294)
R> fit <- BayesSUR(Y = Y, X = X, outFilePath = "results",
+    output_CPO = TRUE)
R> toc()

Time of model fitting: 2.755 sec elapsed

R> fit

Call:
 BayesSUR(Y = Y, X = X, outFilePath = "results", ...)

Number of selected predictors (mPIP > 0.5): 6 of 3x15

Expected log pointwise predictive density (elpd):
 elpd.LOO = -40.84189,  elpd.WAIC = -41.12761
```

The posterior means of the coefficients and latent indicator matrices are printed by the function `plot()` with arguments `estimator = c("beta", "gamma")` and `type = "heatmap"` (Figure 1). Note, that the argument `fig.tex = TRUE` produces PDF figures through LaTeX with the `tools::texi2pdf()` function, which creates authentic math formulas in the figure labels, but requires that the user has LaTeX installed. The argument `output` specifies the name of the PDF file.

```
R> plot(fit, estimator = c("beta", "gamma"), type = "heatmap",
+    fig.tex = TRUE, output = "exampleEst", xlab = "Predictors",
+    ylab = "Responses")
```

Before running the SSUR model with the MRF prior, we need to construct the edge potentials matrix G. If we assume (in accordance with the true matrix $\boldsymbol{B}$ in this simulation scenario) that the second and third predictors are related to the first two response variables, this implies that γ_{21}, γ_{22}, γ_{31} and γ_{32} are expected to be related and therefore we might want to encourage these variables to be selected together. In addition, we assume that we know that the 1st and 4th predictors are associated with the 3rd response variable, and therefore we encourage the selection of γ_{13} as well. Since matrix G represents prior relations of any two predictors corresponding to vec$\{\boldsymbol{\Gamma}\}$, it can be generated by the following code:

```
R> G <- matrix(0, ncol = s * p, nrow = s * p)
R> combn1 <- combn(rep((1:2 - 1) * p, each = length(2:3)) +
+    rep(2:3, times = length(1:2)), 2)
R> combn2 <- combn(rep((3-1) * p, each = length(c(1, 4))) +
+    rep(c(1, 4), times = length(3)), 2)
R> G[c(combn1[1, ], combn2[1]), c(combn1[2, ], combn2[2])] <- 1
```

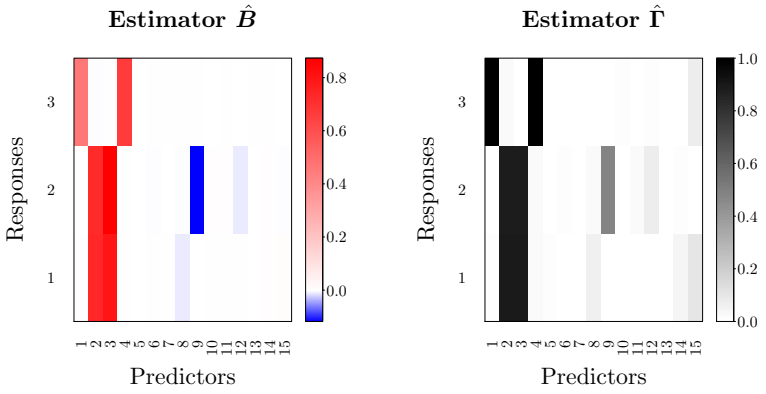

Figure 1: The posterior mean estimates of the coefficients matrix $\hat{\boldsymbol{B}}$ and latent indicator matrix $\hat{\boldsymbol{\Gamma}}$ for the SSUR model with the hotspot prior plotted with `plot(fit, estimator = c("beta", "gamma"), type = "heatmap", ...)`.

Calling `BayesSUR()` with the argument `gammaPrior = "MRF"` will run the SSUR model with the MRF prior, and the argument `mrfG = G` imports the edge potentials for the MRF prior. The two hyper-parameters d and e for the MRF prior (6) can be specified through the argument `hyperpar`; here we use the default values $d = -3$, $e = 0.03$. The posterior mean estimates for the coefficients matrix and latent indicator matrix are shown in Figure 2.

```
R> tic("Time of model fitting")
R> set.seed(5294)
R> fit <- BayesSUR(Y = Y, X = X, outFilePath = "results",
+    gammaPrior = "MRF", mrfG = G)
R> toc()

Time of model fitting: 2.506 sec elapsed

R> plot(fit, estimator = c("beta", "gamma"), type = "heatmap",
+    fig.tex = TRUE, output = "exampleEst2", xlab = "Predictors",
+    ylab = "Responses")
```

5. Two extended examples based on real data

In this section, we use a simulated eQTL data set and real data from a pharmacogenomic database to illustrate the usage of the **BayesSUR** package. The first example is under the known true model and demonstrates the recovery performance of the models introduced in Section 2. It also demonstrates a full data analysis step by step. The second example illustrates how to use potential relationships between multiple response variables and input predictors as the prior information in Bayesian SUR models and showcases how the resulting estimated graph structures can be visualized with functions provided in the package.

5.1. Simulated eQTL data

Similarly to Bottolo *et al.* (2021), we simulate single nucleotide polymorphism (SNP) data $\mathbf{X}$

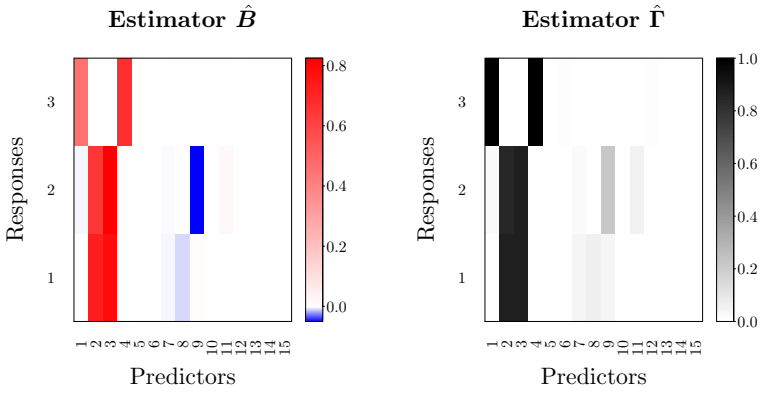

Figure 2: The posterior mean estimates of the coefficients matrix $\hat{\boldsymbol{B}}$ and latent indicator matrix $\hat{\boldsymbol{\Gamma}}$ for the SSUR model with MRF prior plotted with `plot(fit, estimator = c("beta", "gamma"), type = "heatmap", ...)`.

by resampling from the **scrime** package (Schwender 2012), with $p = 150$ SNPs and $n = 100$ subjects. To construct multiple response variables $\mathbf{Y}$ (with $s = 10$) with structured correlation – which we imagine to represent gene expression measurements of genes that are potentially affected by the SNPs – we first fix a sparse latent indicator variable $\boldsymbol{\Gamma}$ and then design a decomposable graph for responses to build association patterns between multi-response variables and predictors. The non-zero coefficients are sampled from the normal distribution independently and the noise term from a multivariate normal distribution with the precision matrix sampled from the $\mathcal{G}$-Wishart distribution $\mathcal{W}_{\mathcal{G}}(2, M)$ (Mohammadi and Wit 2019). Finally, the simulated gene expression data $\mathbf{Y}$ is then generated from the linear model (1). The concrete steps are as follows:

- Simulate SNPs data $\mathbf{X}$ from the **scrime** package, $\dim(\mathbf{X}) = n \times p$.
- Design a decomposable graph $\mathcal{G}$ as in the right panel of Figure 3, $\dim(\mathcal{G}) = s \times s$.
- Design a sparse matrix $\boldsymbol{\Gamma}$ as in the left panel of Figure 3, $\dim(\boldsymbol{\Gamma}) = p \times s$.
- Simulate $\beta_{jk} \sim \mathcal{N}(0, 1)$, $j = 1, \cdots, p$ and $k = 1, \cdots, s$.
- Simulate $\tilde{u}_{ij} \sim \mathcal{N}(0, 0.5^2)$, $i = 1, \cdots, n$ and $j = 1, \cdots, p$.
- Simulate $P \sim \mathcal{W}_{\mathcal{G}}(2, M)$ where diagonals of M are 1 and off-diagonals are 0.9, $\dim(P) = s \times s$.
- Use Cholesky decomposition $\text{chol}(P^{-1})$ to get $\mathbf{U} = \tilde{\mathbf{U}} \cdot \text{chol}(P^{-1})$.
- Generate $\mathbf{Y} = (\mathbf{X}\boldsymbol{B})_{\boldsymbol{\Gamma}} + \mathbf{U}$.

The resulting average signal-to-noise ratio is 25. The R code for the simulation can be found through `help("exampleEQTL")`.

```
R> data("exampleEQTL", package = "BayesSUR")
R> str(exampleEQTL)
```

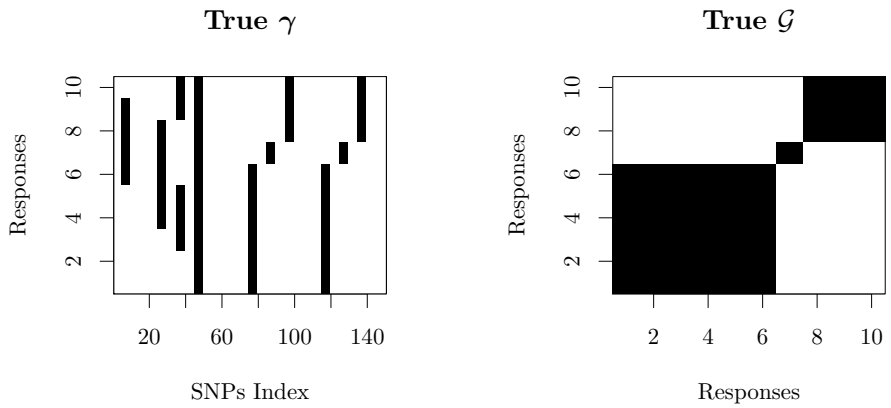

Figure 3: True parameters of the simulated data set `exampleEQTL`. The left panel is the designed sparse matrix $\mathbf{\Gamma}$ and the right panel is the given true structure of responses represented by the decomposible graph $\mathcal{G}$. Black indicates a value of 1 and white indicates 0.

```
List of 4
 $ data     : num [1:100, 1:160] -0.185 -1.01 -2.102 -2.88 1.749 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:100] "1" "2" "3" "4" ...
  .. ..$ : chr [1:160] "GEX1" "GEX2" "GEX3" "GEX4" ...
 $ blockList:List of 2
  ..$ : int [1:10] 1 2 3 4 5 6 7 8 9 10
  ..$ : num [1:150] 11 12 13 14 15 16 17 18 19 20 ...
 $ gamma    : num [1:150, 1:10] 0 0 0 0 0 0 0 0 0 0 ...
 $ Gy       : num [1:10, 1:10] 1 1 1 1 1 1 0 0 0 0 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:10] "GEX1" "GEX2" "GEX3" "GEX4" ...
```

```
R> attach(exampleEQTL)
```

In the **BayesSUR** package, the data $\mathbf{Y}$ and $\mathbf{X}$ are provided as a numeric matrix in the first list component `data` of the example data set `exampleEQTL`. Here the first 10 columns of `data` are the $\mathbf{Y}$ variables, and the last 150 columns are the $\mathbf{X}$ variables. The second component of `exampleEQTL` is `blockList` which specifies the indices of $\mathbf{Y}$ and $\mathbf{X}$ in `data`. The third component is the true latent indicator matrix $\mathbf{\Gamma}$ of regression coefficients. The fourth component is the true graph $\mathcal{G}$ between response variables. Throughout this section we attach the data set for more concise R code.

Figure 3 shows the true $\mathbf{\Gamma}$ and decomposible graph $\mathcal{G}$ used in the eQTL simulation scenario. The following code shows how to fit an SSUR model with hotspot prior for the indicator variables $\mathbf{\Gamma}$ and the sparsity-inducing hyper-inverse Wishart prior for the covariance using the main function `BayesSUR()`.

```
R> set.seed(28173)
R> tic("Time of model fitting")
```

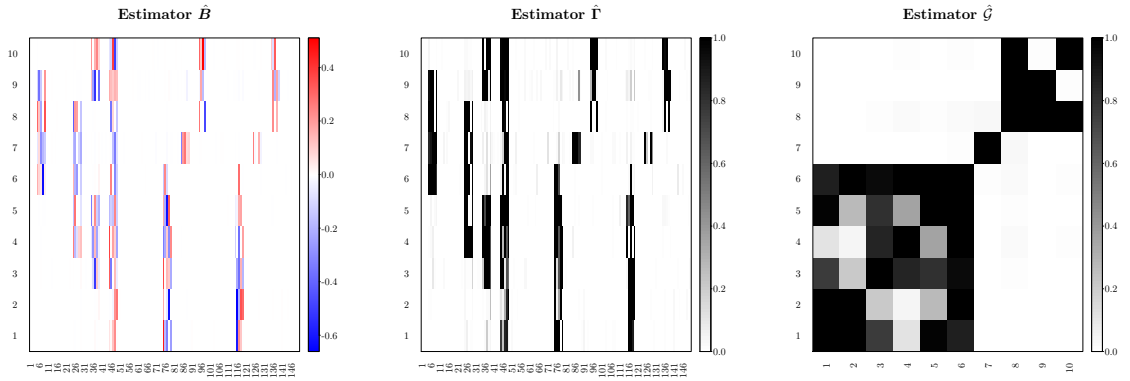

Figure 4: The estimated coefficients matrix $\hat{\boldsymbol{B}}$, latent indicator variable matrix $\hat{\boldsymbol{\Gamma}}$ and learned structure $\hat{\mathcal{G}}$ of the SSUR model with hotspot prior and sparse covariance prior by `plot()`.

```
R> fit <- BayesSUR(data = data, Y = blockList[[1]], X = blockList[[2]],
+    outFilePath = "results", nIter = 200000, nChains = 3,
+    burnin = 100000, covariancePrior = "HIW", gammaPrior = "hotspot")

R> toc()

Time of model fitting: 1159.871 sec elapsed
```

Figure 4 summarizes the posterior inference results by plots for $\hat{\boldsymbol{B}}$, $\hat{\boldsymbol{\Gamma}}$ and $\hat{\mathcal{G}}$ created with the function `plot()` with arguments `estimator = c("beta", "gamma", "Gy")` and `type = "heatmap"`. When comparing with Figure 3, we see that this SSUR model has good recovery of the true latent indicator matrix $\boldsymbol{\Gamma}$ and of the structure of the responses as represented by $\mathcal{G}$. The function `plot()` can also visualize the estimated structure of the ten gene expression variables as shown in the right panel of Figure 5 with arguments `estimator = "Gy"` and `type = "graph"`. For comparison, the true structure is shown in the left panel (created by function `plotGraph()`). When we threshold the posterior selection probability estimates for $\mathcal{G}$ and for $\boldsymbol{\Gamma}$ at 0.5, the resulting full network between the ten gene expression variables and 150 SNPs is displayed in Figure 6. Furthermore, the Manhattan-like plots in Figure 7 show both, the marginal posterior inclusion probabilities (mPIP) of the SNP variables (top panel) and the number of gene expression response variables associated with each SNP (bottom panel).

```
R> plot(fit, estimator = c("beta", "gamma", "Gy"), type = "heatmap",
+    fig.tex = TRUE)
R> layout(matrix(1:2, ncol = 2))
R> plot(fit, estimator = "Gy", type = "graph")
R> plotGraph(Gy)
R> plot(fit, estimator = c("gamma", "Gy"), type = "network",
+    name.predictors = "SNPs", name.responses = "Gene expression")
R> plot(fit, estimator = "gamma", type = "Manhattan")
```

In order to investigate the behavior of the MCMC sampler, the top two panels of Figure 8 show the trace plots of the log-likelihood and model size, i.e., the total number of selected

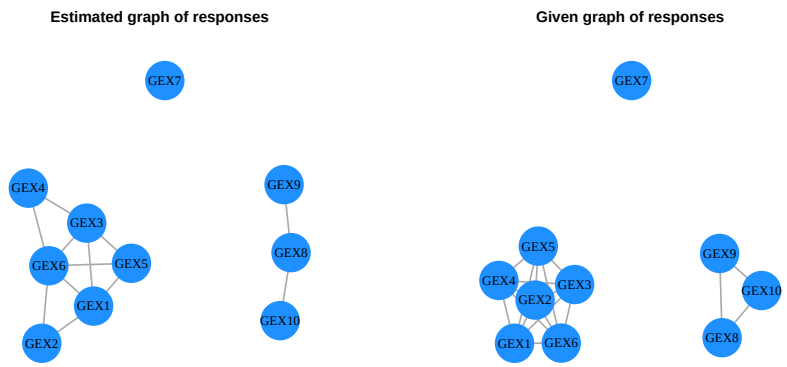

Figure 5: The estimated structure of the ten response variables is visualized by `plot(fit, estimator = "Gy", type = "graph")` with $\hat{\mathcal{G}}$ thresholded at 0.5 (left). The true structure is shown with `plotGraph(Gy)`, where `Gy` is the true adjacency matrix (right).

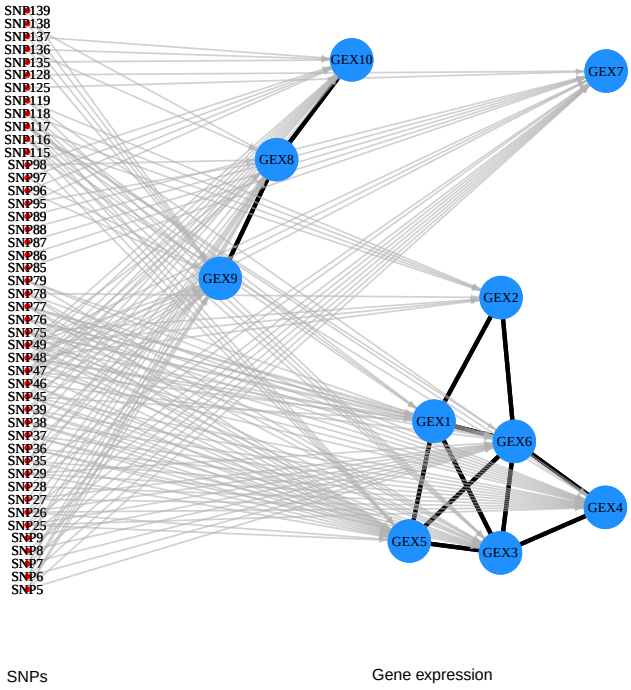

Figure 6: Network representation between the ten gene expression variables and 150 SNP variables by `plot(fit, estimator = c("gamma", "Gy"), type = "network", ...)`. The connections between gene expression variables are based on $\hat{\mathcal{G}}$ thresholded at 0.5, and the connections between the gene expression variables and SNPs are based on $\hat{\mathbf{\Gamma}}$ thresholded at 0.5.

predictors. We observe that the Markov chain seems to start sampling from the correct distribution after ca. 50,000 iterations. The bottom panels of Figure 8 indicate that the log posterior distribution of the latent indicator variable $\mathbf{\Gamma}$ is stable for the last half of the chains after subtracting the burn-in length.

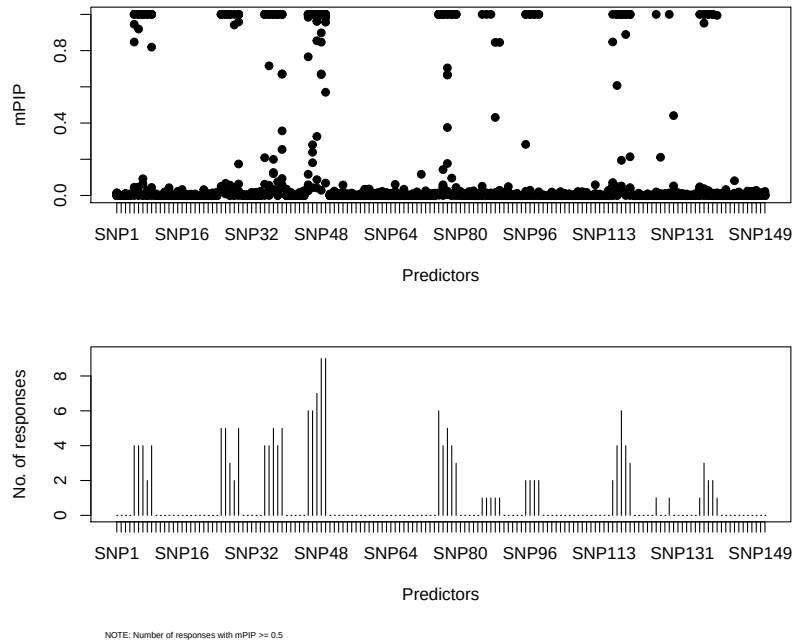

Figure 7: Manhattan-like plots by `plot(fit, estimator = "gamma", type = "Manhattan")`. The top panel shows the marginal posterior inclusion probabilities (mPIP) of each SNP, and the bottom panel shows the number of gene expression response variables associated with each SNP. The number of responses are based on $\hat{\mathbf{\Gamma}}$ thresholded at 0.5.

```
R> plot(fit, estimator = "logP", type = "diagnostics")
```

We finish this example analysis by detaching the eQTL example data set.

```
R> detach(exampleEQTL)
```

5.2. The genomics of drug sensitivity in cancer data

In this section we analyze a subset of the Genomics of Drug Sensitivity in Cancer (GDSC) data set from a large-scale pharmacogenomic study (Yang, Soares, Greninger, Edelman, Lightfoot, Forbes, Bindal, Beare, Smith, Thompson, Ramaswamy, Futreal, Haber, Stratton, Benes, McDermott, and Garnett 2013; Garnett *et al.* 2012). We analyze the pharmacological profiling of $n = 499$ cell lines from $p_0 = 13$ different tissue types for $s = 7$ cancer drugs. The sensitivity of the cell lines to each of the drugs was summarized by the $\log(\text{IC}_{50})$ values estimated from *in vitro* dose response experiments. The cell lines are characterized by $p_1 = 343$ selected gene expression features (GEX), $p_2 = 426$ genes affected by copy number variations (CNV) and $p_3 = 68$ genes with point mutations (MUT). The data sets were downloaded from `ftp://ftp.sanger.ac.uk/pub4/cancerrxgene/releases/release-5.0/` and processed as described in `help("exampleGDSC")`. Gene expression features are log-transformed.

Garnett *et al.* (2012) provide the target genes or pathways for all drugs. The aim of this study was to identify molecular characteristics that help predict the response of a cell line to a particular drug. Because many of the drugs share common targets and mechanisms of

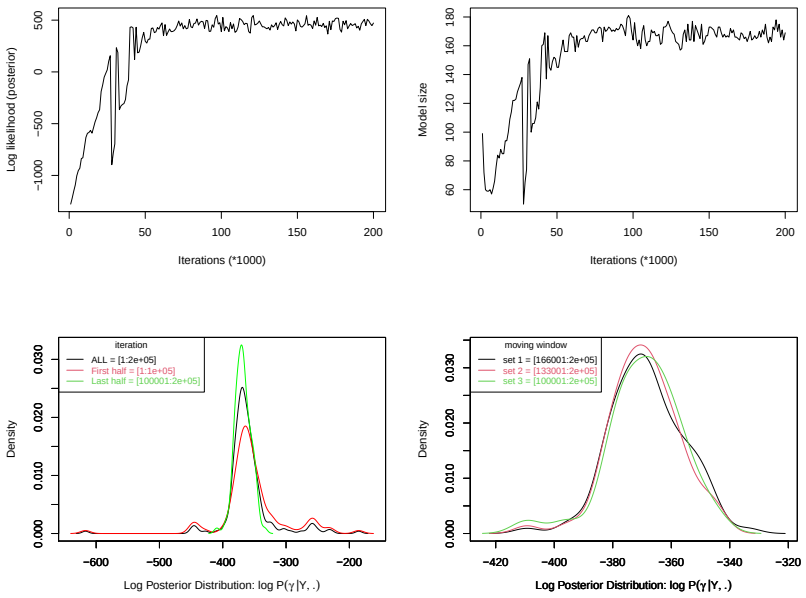

Figure 8: Diagnostic plots of the MCMC sampler by `plot(fit, estimator = "logP", type = "diagnostics")`.

action, the response of cell lines to many of the drugs is expected to be correlated. Therefore, a multivariate model seems appropriate:

$$\mathbf{Y}_{\text{drugs}} = \mathbf{X}_{\text{tissues}}\boldsymbol{B}_0 + \mathbf{X}_{\text{GEX}}\boldsymbol{B}_1 + \mathbf{X}_{\text{CNV}}\boldsymbol{B}_2 + \mathbf{X}_{\text{MUT}}\boldsymbol{B}_3 + \mathbf{U}_{\text{error}},$$

where the elements of $\boldsymbol{B}_0$ and non-zero elements of $\boldsymbol{B}_1$, $\boldsymbol{B}_2$ and $\boldsymbol{B}_3$ are independent and identically distributed with the prior $\mathcal{N}(0, w)$.

We may know the biological relationships within and between drugs and molecular features, so that the MRF prior (6) can be used to learn the above multivariate model well. In our example, we know that the four drugs RDEA119, PD-0325901, CI-1040 and AZD6244 are MEK inhibitors which affect the MAPK/ERK pathway. Drugs Nilotinib and Axitinib are Bcr-Abl tyrosine kinase inhibitors which inhibit the mutated BCR-ABL gene. Finally, the drug Methotrexate is a chemotherapy agent and general immune system suppressant, which is not associated with a particular molecular target gene or pathway. For the target genes (and genes in target pathways) we consider all characteristics (GEX, CNV, MUT) available in our data set as being potentially associated. Based on this information, we construct an edge list of the matrix G for the MRF prior:

- edges between all features representing genes in the MAPK/ERK pathway and the four MEK inhibitors;
- edges between all features representing the Bcr-Abl fusion gene and the two Bcr-Abl inhibitors, see illustration in Figure 9(a);
- edges between all features from different data sources (i.e., GEX, CNV and MUT) representing a gene and all drugs, see illustration in Figure 9(b).

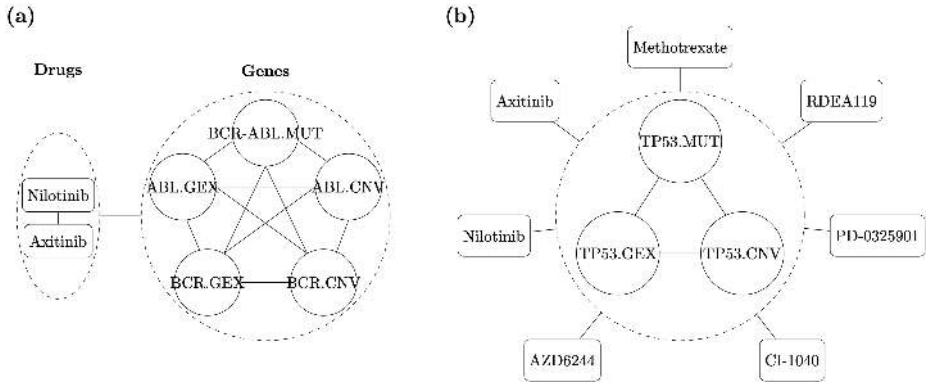

Figure 9: Illustration of the relationship between drugs and a group of related genes. The left panel is for the Bcr-Abl fusion gene and the corresponding related genes. The right panel is for all drugs and gene TP35 as one example with features representing all three data sources. The names with suffix ".GEX", ".CNV" and ".MUT" are features of expression, copy number variation and mutation, respectively.

By matching the selected genes with the gene set of the MAPK/ERK pathway from the KEGG database, 57 features are considered to be connected to the four MEK inhibitors. The two genes (i.e., BCR and ABL) representing the Bcr-Abl fusion are connected with five features in the data set, which are BCR-ABL mutation, BCR gene expression, BCR copy number variation, ABL gene expression and ABL copy number variation (Figure 9(a)). In addition, there are 347 small feature groups representing the different available data sources for each of the genes in the data set, which are potentially connected to all drugs. Figure 9(a) illustrates the edges between drugs Nilotinib, Axitinib and the related genes of the Bcr-Abl fusion gene, and Figure 9(b) uses the TP53 gene as an example for how the different data sources representing a gene are related to each drug, thus linking the data sources together. Based on this information, we construct the matrix G for the MRF prior.

First, we load and attach the data. Note that in this example, we illustrate the use of the specific plot functions `plotEstimator()`, `plotGraph()` and `plotNetwork()`, which are called directly here rather than via the generic `plot()` function as in the examples above.

```
R> data("exampleGDSC", package = "BayesSUR")
R> attach(exampleGDSC)
```

The following code chunk will run the MCMC sampler to fit the model. This represents a full analysis, which might take several hours to run with the chosen MCMC parameter values (`nIter = 200000, nChains = 6, burnin = 100000`) and no parallelization (`maxThreads = 1` by default). Approximate results for an initial assessment of the model can be achieved with much shorter MCMC runs. Note that we use the `X_0` argument for the thirteen cancer tissue types, which are included in the model as mandatory predictors that are always selected.

```
R> hyperpar <- list(mrf_d = -3, mrf_e = 0.2)
R> set.seed(6437)
R> tic("Time of model fitting")
R> fit <- BayesSUR(data = data, Y = blockList[[1]], X_0 = blockList[[2]],
```

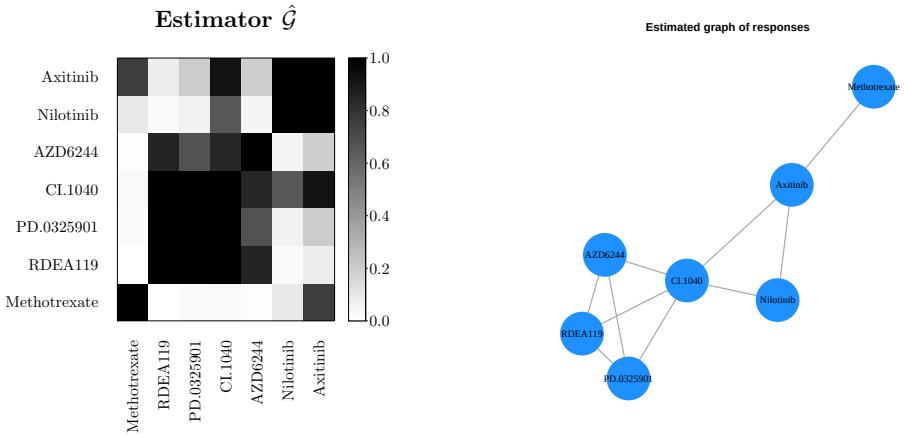

Figure 10: Estimated structure of the seven drugs $\hat{\mathcal{G}}$. Their associations as visualized in the right panel are based on $\hat{\mathcal{G}}$ thresholded at 0.5. Figures created with `plotEstimator()` (left) and `plotGraph()` (right).

```
+    X = blockList[[3]], outFilePath = "results", nIter = 200000,
+    burnin = 100000, nChains = 6, covariancePrior = "HIW",
+    gammaPrior = "MRF", hyperpar = hyperpar, mrfG = mrfG)
R> toc()

Time of model fitting: 7468.874 sec elapsed
```

After fitting an SSUR model with the MRF prior, the structure of the seven drugs, $\mathcal{G}$, has been learned as illustrated in Figure 10, where edges between two drugs k and k' indicate that $\hat{\mathcal{G}}_{kk'} > 0.5$. All expected associations between the drugs within each drug group are found, but some additional connections are also identified: there are edges between Axitinib and Methotrexate and between CI-1040 and both Nilotinib and Axitinib.

```
R> plotEstimator(fit, estimator = "Gy", name.responses = c("Methotrexate",
+    "RDEA119", "PD.0325901",  "CI.1040", "AZD6244", "Nilotinib", "Axitinib"),
+    fig.tex = TRUE, output = "ResponseGraphGDSC1")
R> plotGraph(fit, estimator = "Gy")
R> plotNetwork(fit, estimator = c("gamma", "Gy"), label.predictor = "",
+    name.predictors = "Genes", name.responses = "Drugs",
+    nodesizePredictor = 2)
```

The estimated relationships between the drugs and genes are displayed in Figure 11. There are 259 of all 5859 coefficients selected in total when thresholding $\hat{\mathbf{\Gamma}}$ at 0.5. This results in 82 molecular features being selected for at least one of the drugs, 7 for Methotrexate, 69 for the four MEK inhibitors and 11 for the two Bcr-Abl tyrosine kinase inhibitors.

Network substructures of interest can also be selected and visualized individually, since the user can specify, which response variables (drugs) and which input variables (molecular features) to include in a figure. For example, Figures 12 and 13 show the estimated network representations of the two groups of drugs, respectively.

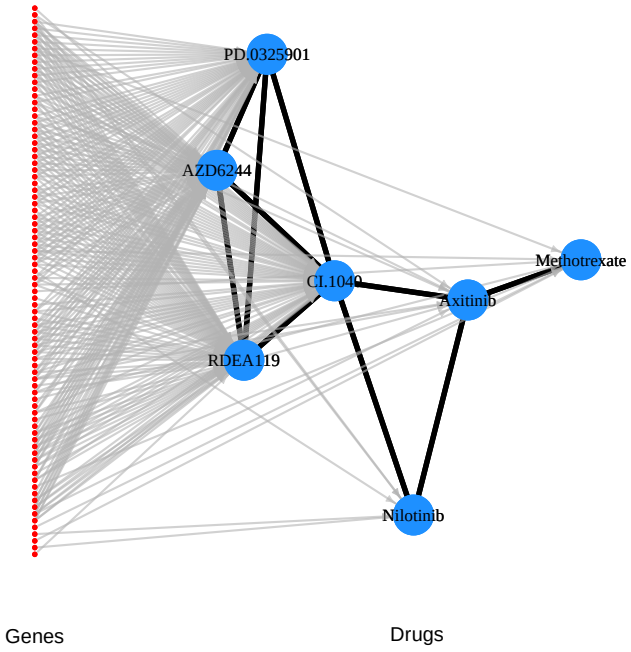

Figure 11: Estimated network between the seven drugs and selected genes based on thresholds 0.5. Figure created with `plotNetwork()`.

```
R> data("targetGene", package = "BayesSUR")
R> plotNetwork(fit, estimator = c("gamma", "Gy"),
+    includeResponse = c("RDEA119", "PD.0325901", "CI.1040", "AZD6244"),
+    includePredictor = names(targetGene$group1))
```

In addition, Figure 13 illustrates, how one can customize the display of the edges between input and response variables to visualize the strength of the association between nodes. In particular, one can either simply use a threshold, e.g., 0.5, to show all edges with marginal posterior inclusion probabilities larger than the threshold equally (left panel), or the width of edges (greater than the specified threshold) can be weighted by the corresponding inclusion probability (right panel).

```
R> layout(matrix(1:2, ncol = 2))
R> plotNetwork(fit, estimator = c("gamma", "Gy"), edge.weight = TRUE,
+    includeResponse = c("Nilotinib", "Axitinib"),
+    includePredictor = names(targetGene$group2))
R> plotNetwork(fit, estimator = c("gamma", "Gy"),
+    edge.weight = TRUE, PmaxPredictor = 0.01,
+    includeResponse = c("Nilotinib", "Axitinib"),
+    includePredictor = names(targetGene$group2))
```

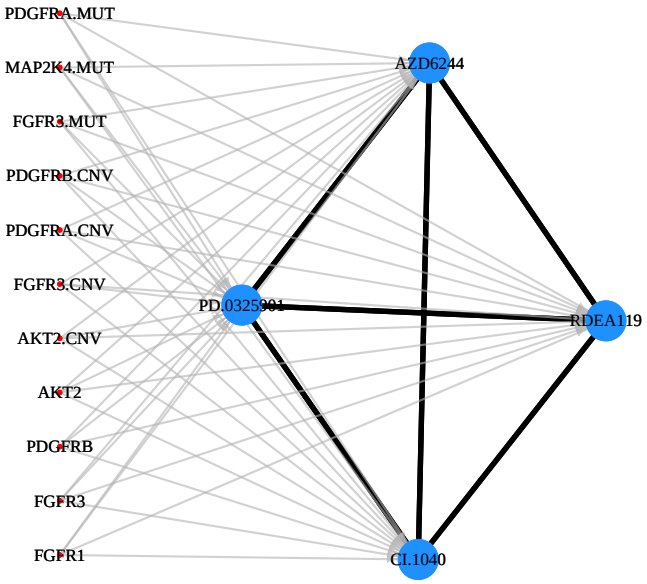

Figure 12: Estimated network between the MEK inhibitors and selected target genes based on thresholds 0.5. Figure created with `plotNetwork()`.

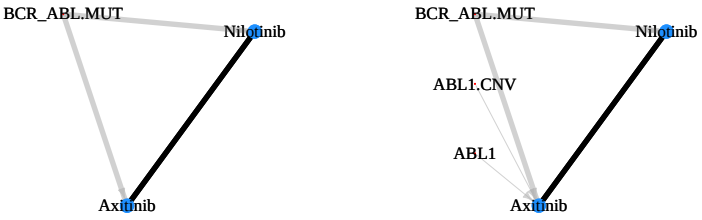

Figure 13: Estimated network between the Bcr-Abl inhibitors and selected target genes. The left panel is based on a threshold on $\hat{\mathbf{\Gamma}}$ of 0.5 while the right panel is based on a threshold of 0.01. Both panels use a threshold on $\hat{\mathcal{G}}$ of 0.5. The edges are weighted by the corresponding inclusion probabilities, if they are greater than the specified thresholds. Figures created with `plotNetwork()`.

6. Conclusion

The **BayesSUR** package presents a series of multivariate Bayesian variable selection models, for which the ESS algorithm is employed for posterior inference over the model space. It provides a unified R package and a consistent interface for the C++ implementations of individual models. The package supports all combinations of the covariance priors and variable selection priors from Section 2 in the Bayesian HRR and SUR model frameworks. This includes the MRF prior on the latent indicator variables to allow the user to make use of prior knowledge of the relationships between both response variables and predictors. To overcome the computational cost for data sets with large numbers of input variables, parallel processing is also implemented with respect to multiple chains, and for calculation of likelihoods of parameters and samples, although the MCMC algorithm itself is still challenging to be parallelized. We demonstrated the modeling aspects of variable selection and structure recovery to identify relationships between multivariate (potentially high-dimensional) responses as well as between responses and high-dimensional predictors, by applying the package to a simulated eQTL data set and to pharmacogenomic data from the GDSC project.

Possible extensions of the R package include the implementation of different priors to introduce even more flexibility in the modeling choices. In particular, the g-prior could be considered for the regression coefficients matrix $\boldsymbol{B}$ (Bottolo and Richardson 2010; Richardson *et al.* 2011; Lewin *et al.* 2015b), whereas currently only the independence prior is available. In addition, the spike-and-slab prior on the covariance matrix C (Wang 2015; Banerjee and Ghosal 2015; Deshpande *et al.* 2019) might be useful, or the horseshoe prior on the latent indicator variable $\boldsymbol{\Gamma}$, which was recently implemented in the multivariate regression setup by Ruffieux, Davison, Hager, Inshaw, Fairfax, Richardson, and Bottolo (2020).

Acknowledgments

A. Lewin and M. Zucknick are joint last authors. The authors thank the editors and the two referees for helpful suggestions. The authors declare no conflicts of interest. This work was made possible through funding from the Faculty of Medicine, University of Oslo (ZZ, MZ), Research Council of Norway project No. 237718 "Big Insight" (ZZ), European Union Horizon 2020 grant agreements No. 847912 "RESCUER" (MZ, SR) and No. 633595 "DynaHealth" (AL), UK Medical Research Council grants MR/M013138/1 (MB, AL, LB, SR) and MC_UU_00002/10 (SR), NIHR Cambridge BRC (SR), BHF-Turing Cardiovascular Data Science Awards 2017 (LB) and The Alan Turing Institute under UK Engineering and Physical Sciences Research Council grant EP/N510129/1 (LB).

References

Banerjee S (2008). "Bayesian Linear Model: Gory Details." URL http://www.biostat.umn.edu/~ph7440/pubh7440/BayesianLinearModelGoryDetails.pdf.

Banerjee S, Ghosal S (2015). "Bayesian Structure Learning in Graphical Models." *Journal of Multivariate Analysis*, **136**, 147–162. doi:10.1016/j.jmva.2015.01.015.

Banterle M, Zhao Z, Lewin A (2021). ***BayesSUR**: Bayesian Seemingly Unrelated Regression*. R package version 2.0-0, URL https://CRAN.R-project.org/package=BayesSUR.

Barbieri MM, Berger JO (2004). "Optimal Predictive Model Selection." *The Annals of Statistics*, **32**(3), 870–897. doi:10.1214/009053604000000238.

Barretina J, Caponigro G, Stransky N, Venkatesan K, Margolin AA, Kim S, Wilson CJ, Lehar J, Kryukov GV, Sonkin D, *et al.* (2012). "The Cancer Cell Line Encyclopedia Enables Predictive Modelling of Anticancer Drug Sensitivity." *Nature*, **483**(7391), 603–607. doi:10.1038/nature11003.

Bhadra A, Mallick BK (2013). "Joint High-Dimensional Bayesian Variable and Covariance Selection with an Application to eQTL Analysis." *Biometrics*, **69**(2), 447–457. doi:10.1111/biom.12021.

Bottolo L, Banterle M, Richardson S, Ala-Korpela M, Järvelin MR, Lewin A (2021). "A computationally efficient Bayesian seemingly unrelated regressions model for high-dimensional quantitative trait loci discovery." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **70**, 886–908. doi:10.1111/rssc.12490.

Bottolo L, Petretto E, Blankenberg S, Cambien F, Cook SA, Tiret L, Richardson S (2011). "Bayesian Detection of Expression Quantitative Trait Loci Hot-Spots." *Genetics*, **189**(4), 1449–1459. doi:10.1534/genetics.111.131425.

Bottolo L, Richardson S (2010). "Evolutionary Stochastic Search for Bayesian Model Exploration." *Bayesian Analysis*, **5**(3), 583–618. doi:10.1214/10-ba523.

Carvalho CM, Massam H, West M (2007). "Simulation of Hyper-Inverse Wishart Distributions in Graphical Models." *Biometrika*, **94**(3), 647–659. doi:10.1093/biomet/asm056.

Csárdi G, Nepusz T (2006). "The **igraph** Software Package for Complex Network Research." *InterJournal*, **Complex Systems**, 1695.

Deshpande SK, Ročková V, George EI (2019). "Simultaneous Variable and Covariance Selection with the Multivariate Spike-and-Slab Lasso." *Journal of Computational and Graphical Statistics*, **28**(4), 921–931. doi:10.1080/10618600.2019.1593179.

Eddelbuettel D, François R (2011). "**Rcpp**: Seamless R and C++ Integration." *Journal of Statistical Software*, **40**(8), 1–18. doi:10.18637/jss.v040.i08.

Eddelbuettel D, Sanderson C (2014). "**RcppArmadillo**: Accelerating R with High-Performance C++ Linear Algebra." *Computational Statistics & Data Analysis*, **71**, 1054–1063. doi:10.1016/j.csda.2013.02.005.

Garnett MJ, Edelman EJ, Heidorn SJ, Greenman CD, Dastur A, Lau KW, Greninger P, Thompson IR, Luo X, Soares J, Liu Q, *et al.* (2012). "Systematic Identification of Genomic Markers of Drug Sensitivity in Cancer Cells." *Nature*, **483**(7391), 570–575. doi:10.1038/nature11005.

Gelfand A (1996). "Model Determination Using Sampling Based Method." In W Gilks, S Richardson, D Spiegelhalter (eds.), *Markov Chain Monte Carlo in Practice*, chapter 9, pp. 145–161. Chapman & Hall, Boca Raton, FL.

Gray JW, Mills GB (2015). "Large-Scale Drug Screens Support Precision Medicine." *Cancer Discovery*, **5**(11), 1130–1132. doi:10.1158/2159-8290.cd-15-0945.

Green PJ, Thomas A (2013). "Sampling Decomposable Graphs Using a Markov Chain on Junction Trees." *Biometrika*, **100**(1), 91–110. doi:10.1093/biomet/ass052.

Holmes CC, Denison DGT, Mallick BK (2002). "Accounting for Model Uncertainty in Seemingly Unrelated Regressions." *Journal of Computational and Graphical Statistics*, **11**(3), 533–551. doi:10.1198/106186002475.

Jia Z, Xu S (2007). "Mapping Quantitative Trait Loci for Expression Abundance." *Genetics*, **176**(1), 611–623. doi:10.1534/genetics.106.065599.

Jones B, Carvalho CM, Dobra A, Hans C, Carter C, West M (2005). "Experiments in Stochastic Computation for High-Dimensional Graphical Models." *Statistical Science*, **20**(4), 388–400. doi:10.1214/088342305000000304.

Lee KH, Tadesse MG, Baccarelli AA, Schwartz J, Coull BA (2017). "Multivariate Bayesian Variable Selection Exploiting Dependence Structure among Outcomes: Application to Air Pollution Effects on DNA Methylation." *Biometrics*, **73**(1), 232–241. doi:10.1111/biom.12557.

Lee KH, Tadesse MG, Coull BA, Starr JR (2021). ***mBvs**: Bayesian Variable Selection Methods for Multivariate Data*. R package version 1.5, URL https://CRAN.R-project.org/package=mBvs.

Lewin A, Campanella G, Saadi H, Liquet B, Chadeau-Hyam M (2015a). ***R2HESS**: Wrapper Functions for Single and Multi-Tissue HESS*. R package version 1.0.1, URL https://www.mrc-bsu.cam.ac.uk/software/.

Lewin A, Saadi H, Peters JE, Moreno-Moral A, Lee JC, Smith KG, Petretto E, Bottolo L, Richardson S (2015b). "MT-HESS: An Efficient Bayesian Approach for Simultaneous Association Detection in OMICS Datasets, with Application to eQTL Mapping in Multiple Tissues." *Bioinformatics*, **32**(4), 523–532. doi:10.1093/bioinformatics/btv568.

Liang F, Wong WH (2000). "Evolutionary Monte Carlo: Applications to C_p Model Sampling and Change Point Problem." *Statistica Sinica*, **10**(2), 317–342.

Liquet B, Bottolo L, Campanella G, Richardson S, Chadeau-Hyam M (2016). "**R2GUESS**: A Graphics Processing Unit-Based R Package for Bayesian Variable Selection Regression of Multivariate Responses." *Journal of Statistical Software*, **69**(2), 1–32. doi:10.18637/jss.v069.i02.

Liquet B, Mengersen K, Pettitt AN, Sutton M (2017). "Bayesian Variable Selection Regression of Multivariate Responses for Group Data." *Bayesian Analysis*, **12**(4), 1039–1067. doi:10.1214/17-ba1081.

Liquet B, Sutton M (2017). ***MBSGS**: Multivariate Bayesian Sparse Group Selection with Spike and Slab*. R package version 1.1.0, URL https://CRAN.R-project.org/package=MBSGS.

Mohammadi R, Wit E (2019). "**BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models." *Journal of Statistical Software*, **89**(3), 1–30. doi:10.18637/jss.v089.i03.

Petretto E, Bottolo L, Langley SR, Heinig M, Mcdermott-Roe C, Sarwar R, Pravenec M, Hubner N, Aitman TJ, Cook SA, Richardson S (2010). "New Insights into the Genetic Control of Gene Expression Using a Bayesian Multi-Tissue Approach." *PLoS Computational Biology*, **6**(4), e1000737. doi:10.1371/journal.pcbi.1000737.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Richardson S, Bottolo L, Rosenthal JS (2011). "Bayesian Models for Sparse Regression Analysis of High Dimensional Data." In *Bayesian Statistics*, volume 9, pp. 539–568. Citeseer.

Ruffieux H, Davison AC, Hager J, Inshaw J, Fairfax BP, Richardson S, Bottolo L (2020). "A Global-Local Approach for Detecting Hotspots in Multiple-Response Regression." *The Annals of Applied Statistics*, **14**(2), 905–928. doi:10.1214/20-aoas1332.

Schwender H (2012). "**scrime**: Analysis of High-Dimensional Categorical Data Such as SNP Data." *R package version 1.3.5*. URL https://CRAN.R-project.org/package=scrime.

Stingo FC, Chen YA, Tadesse MG, Vannucci M (2011). "Incorporating Biological Information into Linear Models: A Bayesian Approach to the Selection of Pathways and Genes." *The Annals of Applied Statistics*, **5**(3), 1978–2002. doi:10.1214/11-aoas463.

Uhler C, Lenkoski A, Richards D (2018). "Exact Formulas for the Normalizing Constants of Wishart Distributions for Graphical Models." *The Annals of Statistics*, **46**(1), 90–118. doi:10.1214/17-aos1543.

Vehtari A, Gelman A, Gabry J (2017). "Practical Bayesian Model Evaluation Using Leave-One-out Cross-Validation and WAIC." *Statistics and Computing*, **27**(5), 1413–1432. doi:10.1007/s11222-016-9696-4.

Wang H (2010). "Sparse Seemingly Unrelated Regression Modelling: Applications in Finance and Econometrics." *Computational Statistics & Data Analysis*, **54**(11), 2866–2877. doi:10.1016/j.csda.2010.03.028.

Wang H (2015). "Scaling It Up: Stochastic Search Structure Learning in Graphical Models." *Bayesian Analysis*, **10**(2), 351–377. doi:10.1214/14-ba916.

Yang W, Soares J, Greninger P, Edelman E, Lightfoot H, Forbes S, Bindal N, Beare D, Smith J, Thompson I, Ramaswamy S, Futreal P, Haber D, Stratton M, Benes C, McDermott U, Garnett M (2013). "Genomics of Drug Sensitivity in Cancer (GDSC): A Resource for Therapeutic Biomarker Discovery in Cancer Cells." *Nucleic Acids Reserch*, **41**(Database issue), D955–61. doi:10.1093/nar/gks1111.

Zhao Z, Banterle M, Lewin A, Zucknick M (2021). "Structured Bayesian Variable Selection for Multiple Related Response Variables and High-Dimensional Predictors." arXiv:2101.05899 [stat.ME], URL https://arxiv.org/abs/2101.05899.

A. The elpd

Without loss of generality, here we only consider each response variable $\boldsymbol{y}$ of the whole response matrix $\mathbf{Y}$, predictor matrix $\mathbf{X}$ and corresponding coefficients p-column vector $\boldsymbol{\beta}$. Then the basic linear model is

$$\begin{aligned} \boldsymbol{y}|\mathbf{X}, \boldsymbol{\beta}, \sigma^2 &\sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbb{I}_m), \\ \boldsymbol{\beta}|\sigma^2 &\sim \mathcal{N}(\boldsymbol{\mu}_\beta, \sigma^2 V_\beta), \\ \sigma^2 &\sim \mathcal{IG}(a, b). \end{aligned} \tag{A.1}$$

In Bottolo *et al.* (2011) and the HRR model of this article, $\boldsymbol{\mu}_\beta = \mathbf{0}$ and $V_\beta = \mathbb{I}_p$ for non-zero coefficients.

A.1. Posterior predictive for the HRR model

From (A.1), the joint distribution of $(\boldsymbol{\beta}, \sigma^2)$ is Normal-Inverse-Gamma, i.e.,

$$f(\boldsymbol{\beta}, \sigma^2) = f(\boldsymbol{\beta}|\sigma^2) f(\sigma^2) = \mathcal{N}(\boldsymbol{\mu}_\beta, \sigma^2 V_\beta) \cdot \mathcal{IG}(a, b) = \mathcal{NIG}(\boldsymbol{\mu}_\beta, V_\beta, a, b).$$

Further we know the posterior distribution of $(\boldsymbol{\beta}, \sigma^2)$ is still Normal-Inverse-Gamma $\mathcal{NIG}(\boldsymbol{\mu}^*_\beta, V^*_\beta, a^*, b^*)$ (Banerjee 2008), where

$$\begin{aligned} \boldsymbol{\mu}^* &= (V_\beta^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} (V_\beta^{-1} \boldsymbol{\mu}_\beta + \mathbf{X}^\top y), \\ V^* &= (V_\beta^{-1} + \mathbf{X}^\top \mathbf{X})^{-1}, \\ a^* &= a + \frac{n}{2}, \\ b^* &= b + \frac{1}{2} (\boldsymbol{\mu}_\beta^\top V_\beta^{-1} \boldsymbol{\mu}_\beta + \boldsymbol{y}^\top \boldsymbol{y} - \boldsymbol{\mu}^{*\top} V^{*-1} \boldsymbol{\mu}^*). \end{aligned}$$

Now we derive the posterior predictive w.r.t. the individual response y_i. Let $f(y_i|\boldsymbol{\beta}, \sigma^2) = \mathcal{N}(X_i \boldsymbol{\beta}, \sigma^2)$, where $X_i = (X_{i1}, \cdots, X_{ip})$ the i-th row of matrix $\mathbf{X}$.

$$\begin{aligned} f(y_i|\boldsymbol{y}) &= \int f(y_i|\boldsymbol{\beta}, \sigma^2) f(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y}) d\boldsymbol{\beta} d\sigma^2 \\ &= \int \mathcal{N}(X_i \boldsymbol{\beta}, \sigma^2) \cdot \mathcal{NIG}(\boldsymbol{\mu}^*, V^*, a^*, b^*) d\boldsymbol{\beta} d\sigma^2 \\ &= \int \frac{b^{*a^*}}{(2\pi)^{\frac{p+1}{2}} \Gamma(a^*) |V^*|^{1/2}} \left(\frac{1}{\sigma^2}\right)^{a^* + \frac{p+1}{2} + 1} \times \\ &\quad \exp\left\{ -\frac{1}{\sigma^2} \left[b^* + \frac{1}{2} \left\{ (\boldsymbol{\beta} - \boldsymbol{\mu}^*)^\top V^{*-1} (\boldsymbol{\beta} - \boldsymbol{\mu}^*) + (y_i - X_i \boldsymbol{\beta})^2 \right\} \right] \right\} d\boldsymbol{\beta} d\sigma^2 \\ &= \int \frac{b^{*a^*}}{(2\pi)^{\frac{p+1}{2}} \Gamma(a^*) |V^*|^{1/2}} \left(\frac{1}{\sigma^2}\right)^{a^* + \frac{p+1}{2} + 1} \times \\ &\quad \exp\left\{ -\frac{1}{\sigma^2} \left[b^{**} + \frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\mu}^{**})^\top V^{**-1} (\boldsymbol{\beta} - \boldsymbol{\mu}^{**}) \right] \right\} d\boldsymbol{\beta} d\sigma^2 \end{aligned}$$

where $|\cdot|$ is determinant and

$$\begin{aligned}\boldsymbol{\mu}^{**} &= (V^{*-1} + X_i^\top X_i)^{-1}(V^{*-1}\boldsymbol{\mu}^* + X_i^\top y_i),\\ V^{**} &= (V^{*-1} + X_i^\top X_i)^{-1},\\ b^{**} &= b^* + \frac{1}{2}(\boldsymbol{\mu}_\beta^{*\top} V^{*-1}\boldsymbol{\mu}^* + y_i^2 - \boldsymbol{\mu}^{**\top} V^{**-1}\boldsymbol{\mu}^{**}).\end{aligned}$$

Let $z \triangleq \frac{c}{\sigma^2}$, $c \triangleq b^{**} + \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\mu}^{**})^\top V^{**-1}(\boldsymbol{\beta} - \boldsymbol{\mu}^{**})$, and then

$$\begin{aligned}f(y_i|\boldsymbol{y}) &= \frac{b^{*a^*}}{(2\pi)^{\frac{p+1}{2}}\Gamma(a^*)|V^*|^{1/2}} \int c^{-(a^*+\frac{p+1}{2}+1)} z^{a^*+\frac{p+1}{2}+1} e^{-z} dc z^{-1} d\boldsymbol{\beta}\\ &= \frac{b^{*a^*}}{(2\pi)^{\frac{p+1}{2}}\Gamma(a^*)|V^*|^{1/2}} \int c^{-(a^*+\frac{p+1}{2})} d\boldsymbol{\beta}\\ &= \frac{b^{*a^*}}{(2\pi)^{\frac{p+1}{2}}\Gamma(a^*)|V^*|^{1/2}} \int \left[b^{**} + \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\mu}^{**})^\top V^{**-1}(\boldsymbol{\beta} - \boldsymbol{\mu}^{**})\right] d\boldsymbol{\beta}\\ &= \frac{b^{*a^*}}{(2\pi)^{\frac{p+1}{2}}\Gamma(a^*)|V^*|^{1/2}} b^{**-\frac{2a^*+p+1}{2}} \int \frac{\Gamma(\frac{2a^*+p+1}{2})}{\Gamma(\frac{2a^*+1}{2})\pi^{p/2}|(2a^*+1)[\frac{2b^{**}}{2a^*+1}V^{**}]|^{\frac{1}{2}}} \times\\ &\quad \left[1 + \frac{(\boldsymbol{\beta} - \boldsymbol{\mu}^{**})^\top [\frac{2b^{**}}{2a^*+1}V^{**}]^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}^{**})}{2a^*+1}\right]^{-\frac{2a^*+p+1}{2}} d\boldsymbol{\beta}.\end{aligned}$$

The integrand above is the density of $\boldsymbol{\beta}$, which is actually a multivariate t-distribution $\text{MVS}t_{2a^*+1}(\boldsymbol{\mu}^{**}, \frac{2b^{**}}{2a^*+1}V^{**})$. Since

$$|2b^{**}V^{**}|^{1/2} = 2^{p/2} b^{**\frac{1}{2}} \frac{|V^*|^{1/2}}{|1 + X_i V^* X_i^\top|^{1/2}},$$

then we have

$$f(y_i|\boldsymbol{y}) = \frac{\Gamma(\frac{2a^*+1}{2})}{\sqrt{2a^*\pi}\Gamma(\frac{2a^*}{2})[\frac{2b^*}{2a^*}(1 + X_i V^* X_i^\top)]^{1/2}} \left[1 + \frac{(y_i - X_i\boldsymbol{\mu}^*)^2\{\frac{2b^*}{2a^*}(1 + X_i V^* X_i^\top)\}}{2a^*}\right]^{-\frac{2a^*+1}{2}}.$$

This is like a univariate t-distribution shifted by $-X_i\boldsymbol{\mu}^*$ and scaled by $\frac{\frac{2b^*}{2a^*}(1+X_i V^* X_i^\top)}{2a^*}$.

Vehtari *et al.* (2017) proposed the expected log pointwise predictive density (elpd) to measure the predictive accuracy for the new data $\tilde{y}_i$ $(i = 1, \cdots, n)$. The elpd is defined as

$$\text{elpd} = \sum_{i=1}^{n} \int f(\tilde{y}_i) \log f(\tilde{y}_i|\boldsymbol{y}) d\tilde{y}_i.$$

Therefore, we use the log pointwise predictive density (lpd) to measure the predictive accuracy, i.e., $\text{lpd} = \sum_{i=1}^{n} \log f(y_i|\boldsymbol{y})$. The widely applicable information criterion (WAIC) is an alternative approach which is

$$\text{lpd} - \sum_{i=1}^{n} \mathbb{V}ar[\log f(y_i|\boldsymbol{y})],$$

where $\mathbb{V}ar[\cdot]$ denotes the variance of logarithm $y_i|\boldsymbol{y}$ that can be estimated from the MCMC iterations.

A.2. Posterior predictive for the dSUR and SSUR models

For the dSUR and SSUR models, the response variables are independent in their parameterized forms. It is feasible to use the out-of-sample predictive to measure the elpd. The Bayesian leave-one-out estimate is

$$\text{elpd}_{\text{loo}} = \sum_{i=1}^{n} \log f(y_i|\boldsymbol{y}_{-i}).$$

As derived via importance sampling, we get

$$f(y_i|\boldsymbol{y}_{-i}) \approx \frac{1}{\frac{1}{T}\sum_{t=1}^{T} \frac{1}{f(y_i|\boldsymbol{\theta}^t)}},$$

where all related parameters $\boldsymbol{\theta}^t$ are drawn from their full posteriors. The WAIC is estimated by

$$\widehat{\text{elpd}}_{\text{waic}} = \widehat{\text{elpd}}_{\text{loo}} - \sum_{i=1}^{n} \mathbb{V}ar_{t=1}^{T}[\log f(y_i|\boldsymbol{\theta}^t)].$$

The posterior predictive $f(y_i|\boldsymbol{y}_{-i})$ can be used to check outliers, which is also named the conditional predictive ordinate (CPO, Gelfand 1996).

Affiliation:

Zhi Zhao, Manuela Zucknick
Oslo Centre for Biostatistics and Epidemiology
Department of Biostatistics
Institute of Basic Medical Sciences
University of Oslo
P.O. Box 1122 Blindern
0317 Oslo, Norway
E-mail: zhi.zhao@medisin.uio.no, manuela.zucknick@medisin.uio.no

Marco Banterle, Alex Lewin
Department of Medical Statistics
Faculty of Epidemiology and Population Health
London School of Hygiene & Tropical Medicine
Keppel St, Bloomsbury
London WC1E 7HT, United Kingdom
E-mail: marco.banterle@gmail.com, alex.lewin@lshtm.ac.uk

Leonardo Bottolo
Department of Medical Genetics
University of Cambridge
J. J. Thomson Avenue
Cambridge CB2 0QQ, United Kingdom
E-mail: lb664@cam.ac.uk

Sylvia Richardson
MRC Biostatistics Unit
University of Cambridge
Robinson Way
Cambridge CB2 0SR, United Kingdom
E-mail: sylvia.richardson@mrc-bsu.cam.ac.uk