

SME0809 - Inferência Bayesiana - Prova 2 - Grupo 13
High-Dimensional Multivariate Bayesian Variable and Covariance Selection in Linear
Regression (Zhao et al. 2021)

Francisco Miranda - 4402962 Heitor Carvalho - 11833351

Dezembro 2021

Contents

1	Introdução	1
2	Metodologia	2
2.1	Regressão Hierárquica Relacionada (HRR)	3
2.2	Regressão não relacionada aparentemente esparsa (SSUR)	4
2.3	Amostragem MCMC e inferência <i>a posteriori</i>	4
3	Conjunto de Dados	5
4	Análise dos Dados	5
5	Conclusão	15
6	Referências	15
7	Apêndice: códigos	15

1 Introdução

Com o desenvolvimento de técnicas de alto processamento na biologia molecular, a caracterização molecular em alta escala tornou-se um lugar comum, com o advento de técnicas como:

- genome-wide measurement of gene expression
- single nucleotide polymorphisms
- CpG methylation status
- pharmacological profiling for large-scale cancer drug screen.

A análise de associações conjuntas entre múltiplos fenótipos correlacionados e atributos moleculares de alta dimensionalidade é desafiadora.

Quando múltiplos fenótipos e informação genômica de alta dimensionalidade são analisados conjuntamente, a abordagem bayesiana permite especificar de maneira flexível as relações complexas entre os conjunto de dados altamente estruturados.

O pacote **BayesSUR** combina diversos modelos que foram propostos para a regressão multidimensional com resposta múltipla e introduz um novo modelo, que permite diferentes *prioris* na seleção de variáveis dos modelos de regressão e para diferentes pressupostos a respeito da estrutura de dependência entre as respostas.

2 Metodologia

- múltiplas opções de seleção de variáveis
- a matriz de covariância pode ser diagonal, densa ou esparsa.
- engloba três classes de modelos de regressão linear de múltipla resposta:
- HRR
- dSUR e SSUR
- MRF

O modelo de regressão é escrito como:

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{U} \quad (1)$$

$$\text{vec}(\mathbf{U}) \sim \mathcal{N}(\mathbf{0}, C \otimes \mathbb{I}_n)$$

onde:

- \mathbf{Y} é uma matriz $s \times s$ das variáveis resposta com matriz de covariância C ;
- \mathbf{X} é uma matriz $n \times p$ de preditores para todas as respostas;
- \mathbf{U} é a matriz dos resíduos;
- $\text{vec}(\cdot)$ denota a vetorização da matriz;
- $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denota uma distribuição normal multivariada com vetor de médias $\boldsymbol{\mu}$ e matriz de covariâncias $\boldsymbol{\Sigma}$;
- $\mathbf{0}$ denota um vetor coluna com todos os elementos nulos,
- \otimes é o produto de Kronecker e \mathbb{I}_n a matriz identidade de ordem n .

A seleção de variáveis é realizada através de uma matriz indicadora binária latente $\boldsymbol{\Gamma} = \{\gamma_{jk}\}$.

Uma *priori* “spike-and-slab” é utilizada para encontrar um subconjunto esparsa relevante de preditores que expliquem a variabilidade de \mathbf{Y} : condicional em $\gamma_{jk} = 0$ ($j = 1, \dots, p$, e $k = 1, \dots, s$)

Definem-se $\beta_{jk} = 0$ condicionado em $\gamma_{jk} = 1$ seguem uma distribuição normal difusa:

$$\beta_{\gamma} \sim \mathcal{N}(\mathbf{0}, W_{\gamma}^{-1}) \quad (2)$$

Onde $\beta = \text{vec}(\mathbf{B})$, $\gamma = \text{vec}(\boldsymbol{\Gamma})$, β_{γ} consiste somente nos coeficientes selecionados (i.e. $\gamma_{jk} = 1$), assim W_{γ} é a sub matriz de W formada pelos coeficientes selecionados correspondentes.

A matriz de precisão, W , é geralmente decomposta em coeficientes de encolhimento e uma matriz que governa a estrutura de covariância dos coeficientes de regressão. É utilizado aqui $W = w^{-1}\mathbb{I}_{sp}$, o que significa que todos os coeficientes de regressão são independentes a priori, com uma *hiperpriori* no coeficiente de encolhimento w , i.e. $w \sim \mathcal{IGamma}(a_w, b_w)$.

A matriz indicadora binária latente $\boldsymbol{\Gamma}$ tem três opções de *priori*:

- Bernoulli independente hierárquica
- hotspot prior
- MRF prior

A matriz de covariância C também possui três *prioris*:

- Gama inversa independente
- Wishart inversa
- hiper-inversa Wishart

São considerados no total nove possíveis modelos dentre as combinações de C e $\mathbf{\Gamma}$

	$\gamma_{jk} \sim \text{Bernoulli}$	$\gamma_{jk} \sim \text{hotspot}$	$\gamma_{jk} \sim \text{MRF}$
$C \sim \text{indep}$	HRR-B	HRR-H	HRR-M
$C \sim \text{IW}$	dSUR-B	dSUR-H	dSUR-M
$C \sim \text{HIW}$	SSUR-B	SSUR-H	SSUR-M

2.1 Regressão Hierárquica Relacionada (HRR)

A Regressão Hierárquica Relacionada assume que C é uma matriz diagonal, o que se traduz em independência condicional entre múltiplas variáveis resposta.

Uma *priori* gama inversa é especificada para a covariância dos resíduos, i.e

$$\sigma_k^2 \sim \mathcal{IGamma}(a_\sigma, b_\sigma)$$

Quando combinada com as *prioris* em (2), é conjugado com a verossimilhança do modelo (1). Podemos então amostrar a estrutura de seleção de variáveis $\mathbf{\Gamma}$ marginalmente com respeito a C e \mathbf{B} .

2.1.1 HRR com uma *priori* Bernouli independente

Para uma *priori* simples de seleção do modelo de regressão, os indicadores binários latentes seguem uma *priori* de Bernoulli:

$$\gamma_{jk} | \omega_{jk} \sim \text{Ber}(\omega_{jk}) \quad (j = 1, \dots, p, \text{ e } k = 1, \dots, s) \quad (3)$$

Com uma *priori* hierárquica Beta em ω_j , i.e. $\omega_j \sim \text{Beta}(a_\omega, b_\omega)$, que quantifica a probabilidade de cada preditor ser associado com qualquer uma das variáveis resposta.

2.1.2 HRR com uma *priori* hotspot

É proposta a decomposição da probabilidade do parâmetro de associação ω_{jk} em (3), onde o_k é responsável pela esparsividade de cada modelo de resposta e π_j controla a propensão de cada preditor a ser associado a múltiplas respostas simultaneamente:

$$\gamma_{jk} | \omega_{jk} \sim \text{Ber}(\omega_{jk}) \quad (j = 1, \dots, p, \text{ e } k = 1, \dots, s) \quad (4)$$

$$\begin{aligned} \omega_{jk} &= o_k \times \pi_j \\ o_k &\sim \text{Beta}(a_o, b_o) \\ \pi_j &\sim \text{Gamma}(a_\pi, b_\pi) \end{aligned}$$

2.2 Regressão não relacionada aparentemente esparsa (SSUR)

Para modelar a matriz de covariância C é especificado uma *priori* hiper-Inversa Wishart, o que significa que as variáveis resposta têm por trás um grafo \mathcal{G} que codifica a dependência condicional entre as respostas.

Um grafo esparsos corresponde à matriz esparsa de precisão C^{-1} . Do ponto de vista computacional, é impraticável especificar uma *priori* hiper-inversa Wishart diretamente em C^{-1} . É realizada uma transformação em C para fatorar a verossimilhança.

A distribuição hiper inversa de Wishart i.e $C \sim \mathcal{HIW}_{\mathcal{G}}(\nu, \tau \mathbb{I}_s)$ transforma-se na variância escalar σ_{qt}^2 e no vetor de correlação associado $\boldsymbol{\rho}_{qt} = (\rho_{1,qt}, \rho_{2,qt}, \dots, \rho_{t-1,qt})^T$, com:

$$\sigma_{qt}^2 \sim \mathcal{IGamma}\left(\frac{\nu - s + t + |S_q|}{2}, \frac{\tau}{2}\right), \quad q = 1, \dots, Q, \quad t = 1, \dots, |R_q|, \quad \boldsymbol{\rho}_{qt} | \sigma_{qt}^2 \sim \mathcal{N}\left(\mathbf{0}, \frac{\sigma_{qt}^2}{\tau} \mathbb{I}_{t-1}\right) \quad (5)$$

onde Q é o número de componentes primos no grafo decomposto \mathcal{G} , S_q e R_q são os separadores e os componentes residuais de \mathcal{G} , respectivamente.

Como *priori* para o grafo é utilizado uma Bernoulli com probabilidade η em cada vértice $E_{kk'}$ de \mathcal{G} como em:

$$\mathbb{P}(E_{kk'} \in \mathcal{G}) = \eta, \quad \eta \sim \mathcal{Beta}(a_\eta, b_\eta). \quad (6)$$

São admitidas três *prioris* em β_γ .

2.3 Amostragem MCMC e inferência *a posteriori*

Para amostrar da distribuição *a posteriori*, os autores utilizam o algoritmo de busca estocástica evolucionária, que utiliza uma forma particular do Monte Carlo evolucionário (EMC).

Múltiplas cadeias de Markov temperadas são processadas paralelamente e movimentos de troca ou mudança são permitidos dentre as cadeias para melhorar a mistura entre modelos potencialmente diferentes da *posteriori*. A temperatura é adaptada durante a fase de burn-in.

A cadeia principal provém amostras da distribuição *a posteriori* não-temperada, que é utilizada para toda a inferência. Para cada variável resposta, os autores utilizam um amostrador de Gibbs para atualizar o vetor dos coeficientes de regressão $\beta_k (k = 1, \dots, s)$, baseado na distribuição *a posteriori* condicional correspondente ao modelo específico, selecionado entre os modelos apresentados anteriormente.

Após L iterações do MCMC, obtêm-se $\mathbf{B}^{(1)}, \dots, \mathbf{B}^{(L)}$ e a estimativa da média *a posteriori* é:

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

onde b é o número de iterações de *burn-in*. As distribuições condicionais completas *a posteriori* também estão disponíveis no modelo SSUR. Já nos modelos HRR, os coeficientes de regressão e as covariâncias residuais foram integrados para fora e ainda assim a saída do MCMC não pode ser utilizada diretamente para inferência posterior desses parâmetros.

Contudo, para \mathbf{B} , a distribuição *posteriori* condicional em $\boldsymbol{\Gamma}$ pode ser obtida analiticamente nos modelos HRR, e é essa a saída oferecida.

Em cada iteração t do MCMC também é atualizado cada vetor binário latente $\gamma_k (k = 1, \dots, s)$ via Metropolis-Hastings, propondo conjuntamente uma atualização para o correspondente β_k . Após L iterações, usando as matrizes binárias $\boldsymbol{\Gamma}^{(1)}, \dots, \boldsymbol{\Gamma}^{(L)}$, as probabilidades de inclusão marginal *a posteriori* são estimadas por:

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

Outro parâmetro importante dos modelos SSUR é \mathcal{G} na *priori* Wishart hiper-inversa para a matriz de covariância C . Ela é atualizada via *junction tree sampler* conjuntamente com a proposta correspondente para σ_{qt}^2 e $\boldsymbol{\rho}_{qt}|\sigma_{qt}^2$ em (5).

A cada iteração do MCMC é extraída a matriz de adjacência $\mathcal{G}^{(t)}(t = 1, \dots, L)$, do qual são derivadas as estimativas da média a *posteriori* das probabilidades de inclusão das areastas como:

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

Mesmo que a *priori* o grafo \mathcal{G} seja decomposto, a média estimada posteriormente $\hat{\mathcal{G}}$ pode estar no espaço de modelos decompostos.

O hiperparâmetro τ da Wishart hiper-inversa é atualizado através de um passeio aleatório do amostrador Metropolis-Hastings. Já η e a variância w na *priori* pico-e-tapa são amostrados das condicionais posteriores.

3 Conjunto de Dados

Os autores simularam dados de polimorfismo de nucleotídeo único (SNP) dentro de um modelo verdadeiro conhecido para demonstrar a performance de recuperação dos modelos introduzidos anteriormente. O algoritmo completo pode ser encontrado em (Zhao et al. 2021).

Para construir variáveis resposta múltiplas \mathbf{Y} (com $s = 10$) com uma relação estruturada, os autores fixam uma variável indicadora esparsa $\mathbf{\Gamma}$ e desenham um grafo decomposto para as respostas, para construir padrões de associação dentre os múltiplos regressores e variáveis resposta.

Aqui, temos 10 colunas, que são as respostas em \mathbf{Y} , e 150 linhas, que são os regressores em \mathbf{X} . O segundo componente do exemplo é uma `blockList` que especifica os índices de \mathbf{Y} e \mathbf{X} em `data`.

O terceiro componente é a verdadeira matriz indicadora $\mathbf{\Gamma}$ dos coeficientes de regressão. O quarto componente é o grafo verdadeiro \mathcal{G} entre as variáveis resposta. A Figura 1 mostra os verdadeiros $\mathbf{\Gamma}$ e \mathcal{G} utilizados para simular o exemplo. As matrizes de associação são exibidas na forma de mapas de calor.

4 Análise dos Dados

Os autores utilizam o exemplo para ajustar um modelo SSUR com uma *priori hotspot* para as variáveis indicadoras $\mathbf{\Gamma}$ e a *priori* indutora de esparsidade Wishart hiper-inversa utilizando o pacote `BayesSUR`. No exemplo são 200000 iterações do MCMC, com burn-in de 100000, em três cadeias paralelamente. Para manter a reprodutibilidade, o código foi executado em um único núcleo.

```
## BayesSUR -- Bayesian Seemingly Unrelated Regression Modelling
## Using OpenMP
## Reading input files ... .. successfull!
## Clearing and initialising output files
## Initialising the (SUR) MCMC Chain ... .. DONE!
## Drafting the output files with the start of the chain ... DONE!
##
## Starting 3 (parallel) chain(s) for 5000 iterations:
```

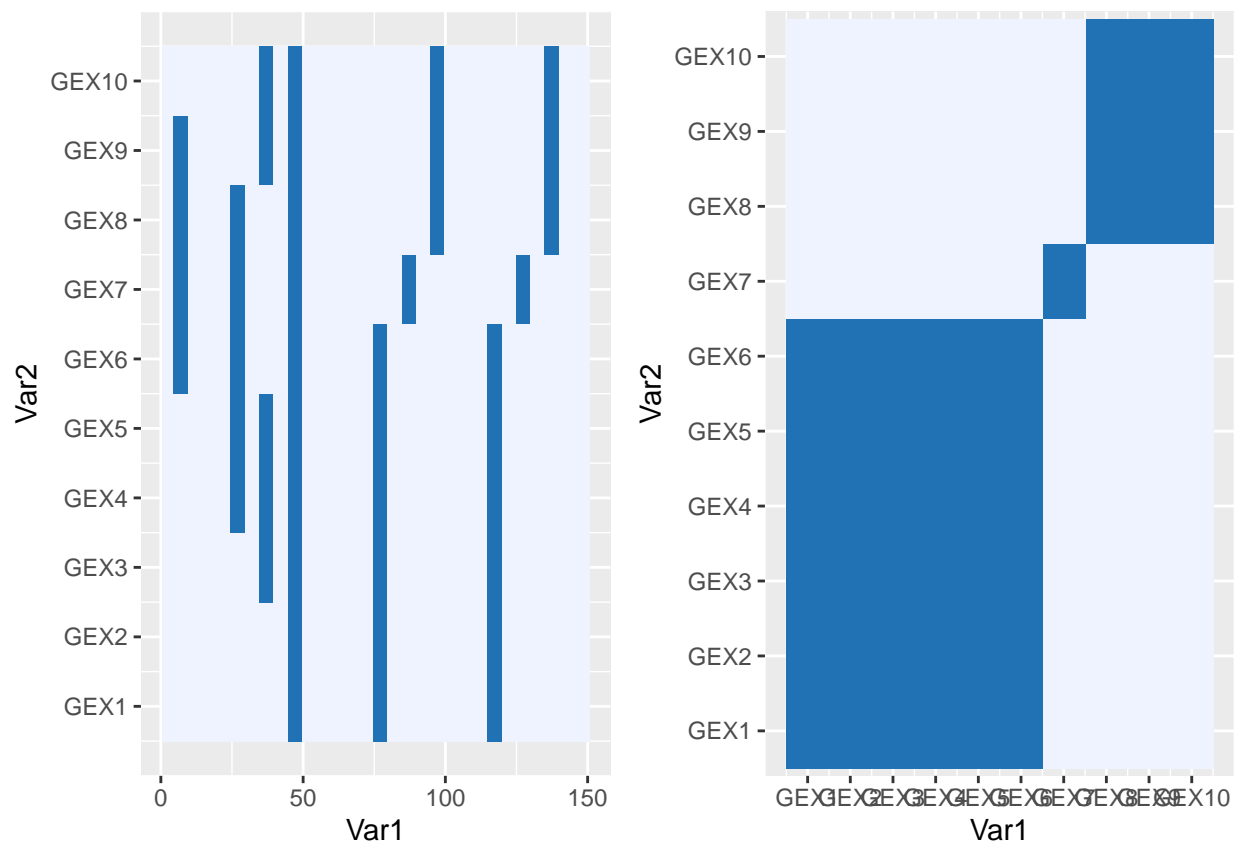


Figure 1: Parâmetros verdadeiros dos dados gerados no conjunto de dados de exemplo. Valores em branco representam 0 e, em azul, 1

```

## Temperature ladder updated, new temperature ratio : 1.1
## Temperature ladder updated, new temperature ratio : 1.21
## Temperature ladder updated, new temperature ratio : 1.331
## Temperature ladder updated, new temperature ratio : 1.4641
## Temperature ladder updated, new temperature ratio : 1.61051
## Temperature ladder updated, new temperature ratio : 1.77156
## Running iteration 1000 ... local Acc Rate: ~ gamma: 0.065 -- JT: 0.057 -- Global: 0.242
## Running iteration 2000 ... local Acc Rate: ~ gamma: 0.06 -- JT: 0.083 -- Global: 0.24
## Running iteration 3000 ... local Acc Rate: ~ gamma: 0.053 -- JT: 0.099 -- Global: 0.22
## Running iteration 4000 ... local Acc Rate: ~ gamma: 0.051 -- JT: 0.11 -- Global: 0.194
## Running iteration 5000 ... local Acc Rate: ~ gamma: 0.05 -- JT: 0.108 -- Global: 0.178
## MCMC ends. --- Saving results and exiting
## Saved to : results/data_SSUR_****_out.txt
## Final w : 0.261593
## Final tau : 0.967061 w/ proposal variance: 0.496255
## Final eta : 0.047631
## -- Average Omega : 0.0225257
## Final temperature ratio : 1.77156
##
## DONE, exiting!

## Tempo de ajuste do modelo: 64.715 sec elapsed

```

As Figuras 2, 3 e 4 resumem os resultados da inferência a posteriori dos estimadores \hat{B} , $\hat{\Gamma}$, e $\hat{\mathcal{G}}$.

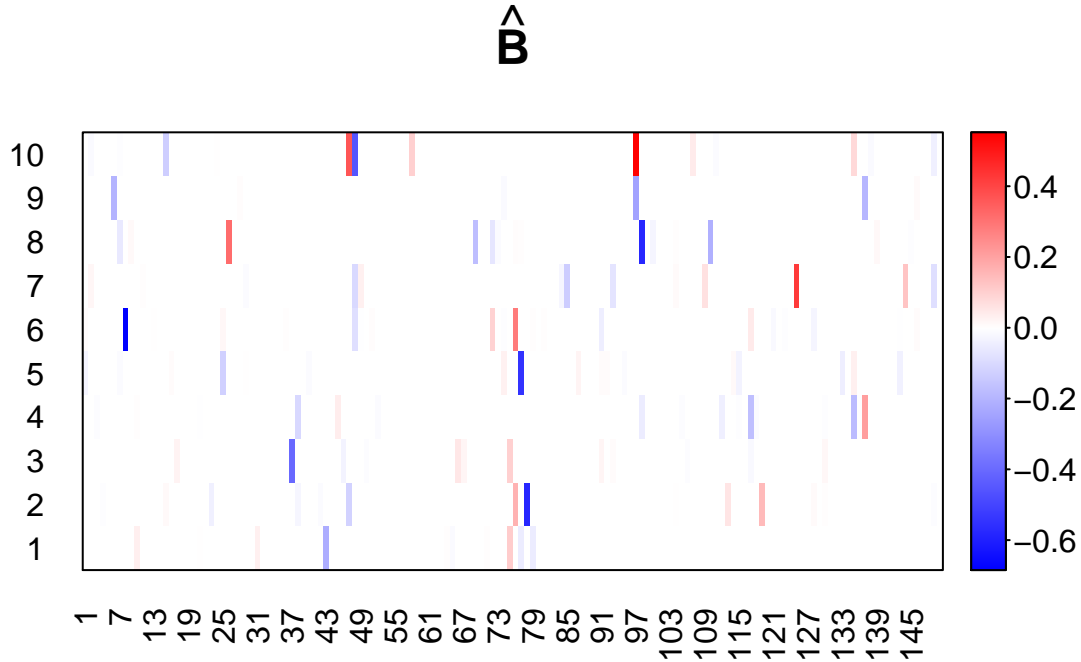


Figure 2: Coeficientes estimados da matriz \hat{B}

Vemos que o modelo SSUR possui uma boa recuperação da verdadeira matriz indicadora latente Γ e da estrutura das respostas representada por \mathcal{G} . Podemos comparar a verdadeira estrutura com a estimada quando limitamos a probabilidade de seleção a *posteriori* para Γ e \mathcal{G} a 0.5, criando o grafo exibido à direita na 1.

Utilizando o mesmo limiar, de $\hat{\mathcal{G}}$ limitado a 0.5, e $\hat{\Gamma}$ limitado a 0.5, podemos visualizar a rede completa com as dez expressões gênicas e os 150 SNPs na 6.

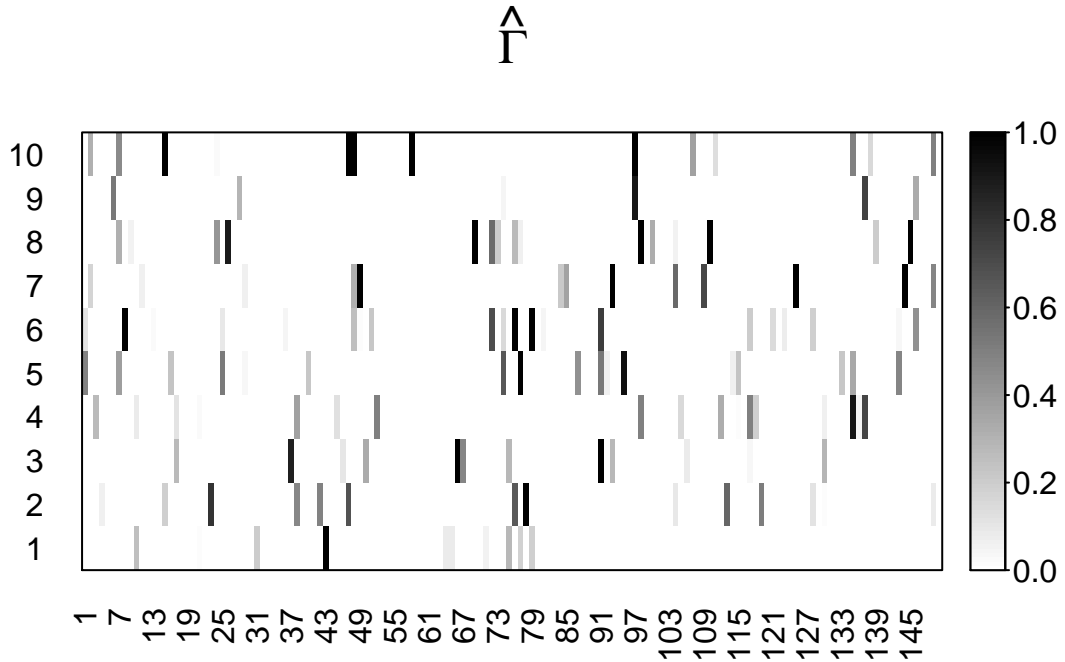


Figure 3: Coeficientes estimados da matriz indicadora latente $\hat{\Gamma}$

Estimated graph of responses

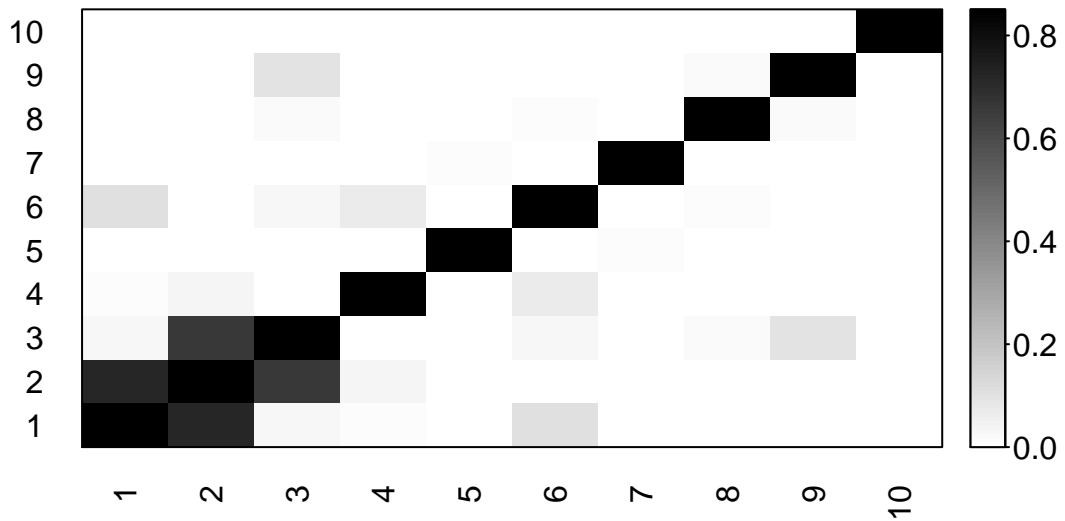


Figure 4: Estrutura aprendida de $\hat{\mathcal{G}}$ pelo modelo SSUR com priori hotspot e covariância esparsa

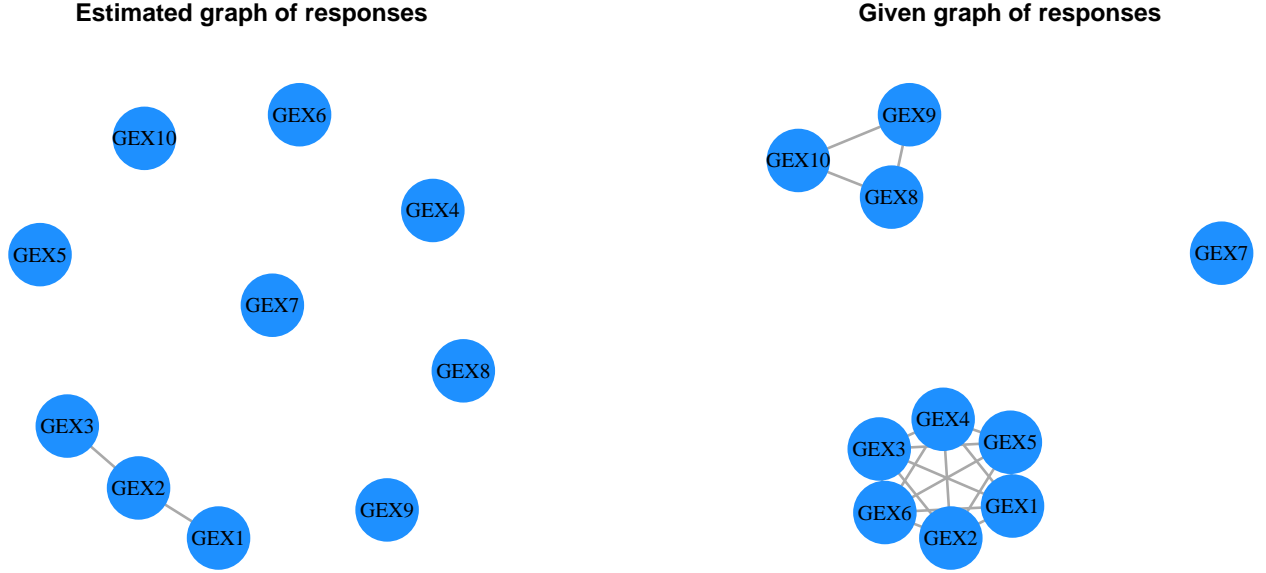
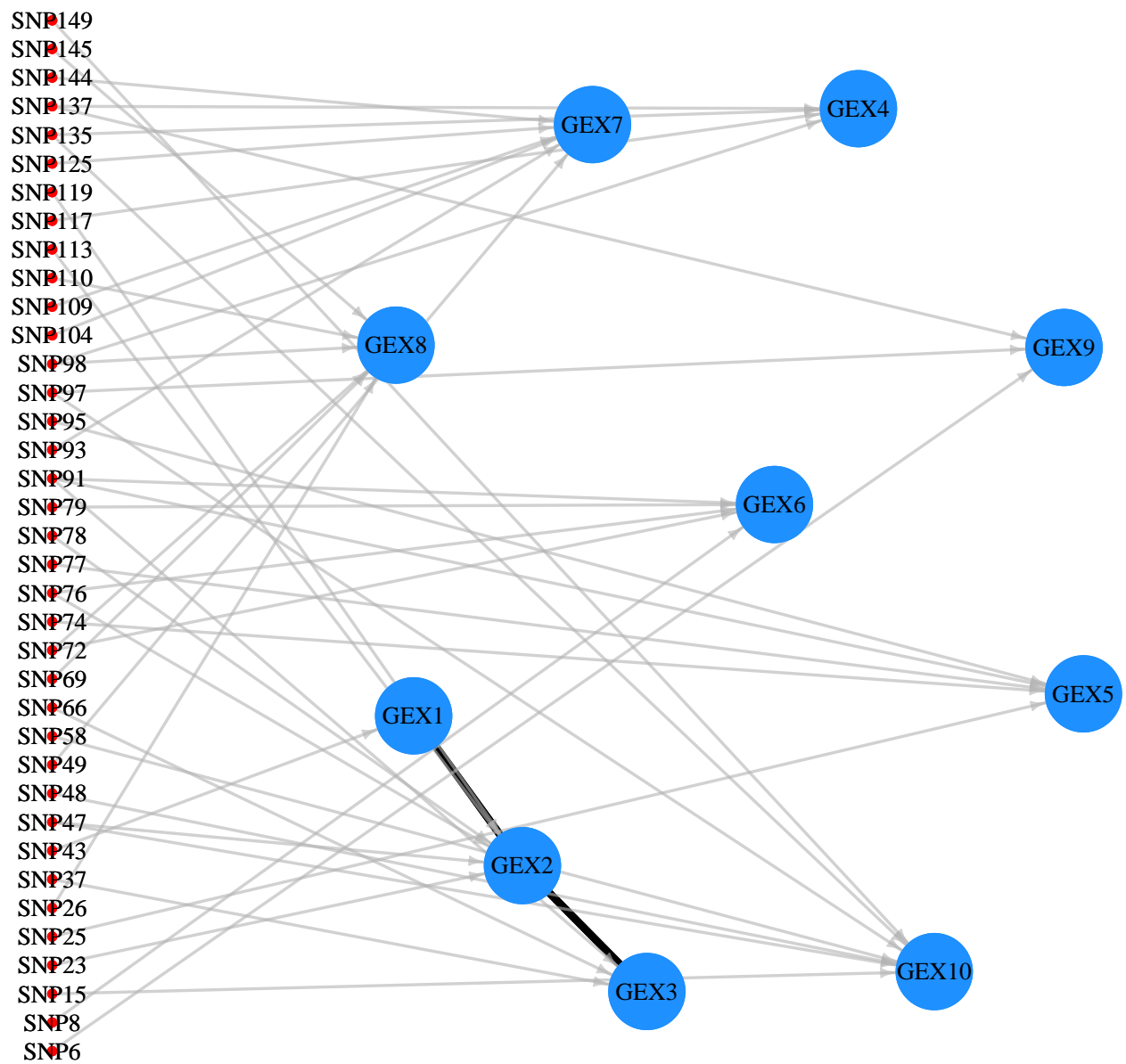


Figure 5: Estrutura estimada das dez variáveis resposta com $\hat{\mathcal{G}}$ limitado a 0.5 (esquerda). Estrutura verdadeira de \mathcal{G} representada pela matriz de adjacência \mathbf{G}_y (direita).

Além disso, os plots estilo Manhattan na Figura 7 mostram as probabilidades de inclusão marginal (mPIP) dos SNPs (painel superior) e o número de expressões gênicas da resposta associado com cada SNP (painel inferior). O número de respostas é baseado em $\hat{\mathbf{\Gamma}}$ limitado a 0.5.

Para investigarmos o comportamento do nosso amostrador MCMC, podemos utilizar os plots de diagnóstico apresentados na ???. Observa-se que as cadeias de Markov utilizadas aparentemente começam a amostrar da distribuição correta após aproximadamente 50.000 iterações.

Os painéis inferiores da Figura ??? indicam que o logaritmo da distribuição a posteriori da variável indicadora latente $\mathbf{\Gamma}$ é estável para a última metade das cadeias, após subtraído o burn-in.



SNPs

Gene expression

Figure 6: Representação da rede entre as expressões gênicas (\mathbf{Y}), com $\hat{\mathcal{G}}$ limitado a 0.5, e os SNPs (\mathbf{X}), com $\hat{\mathbf{\Gamma}}$ limitado a 0.5.

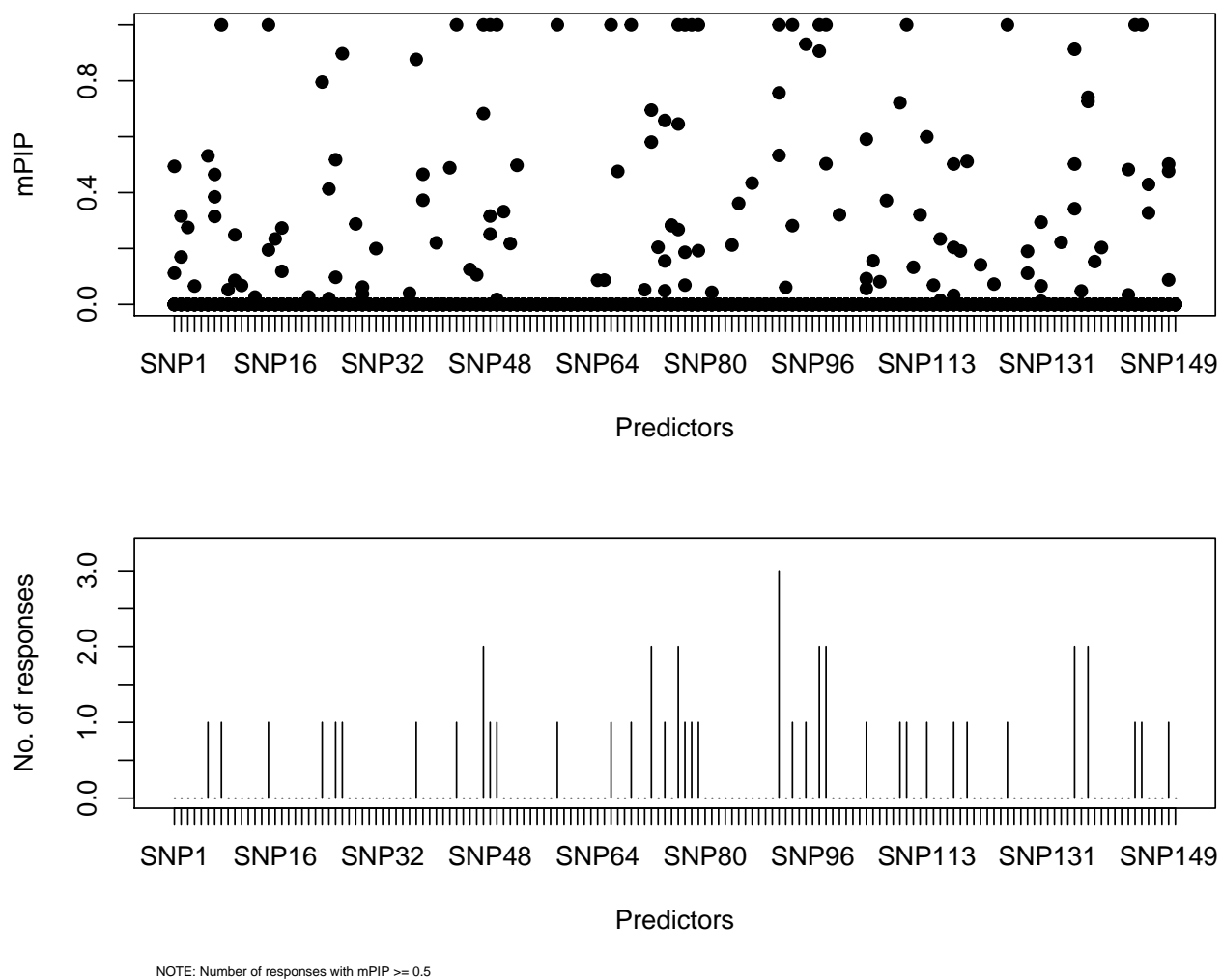
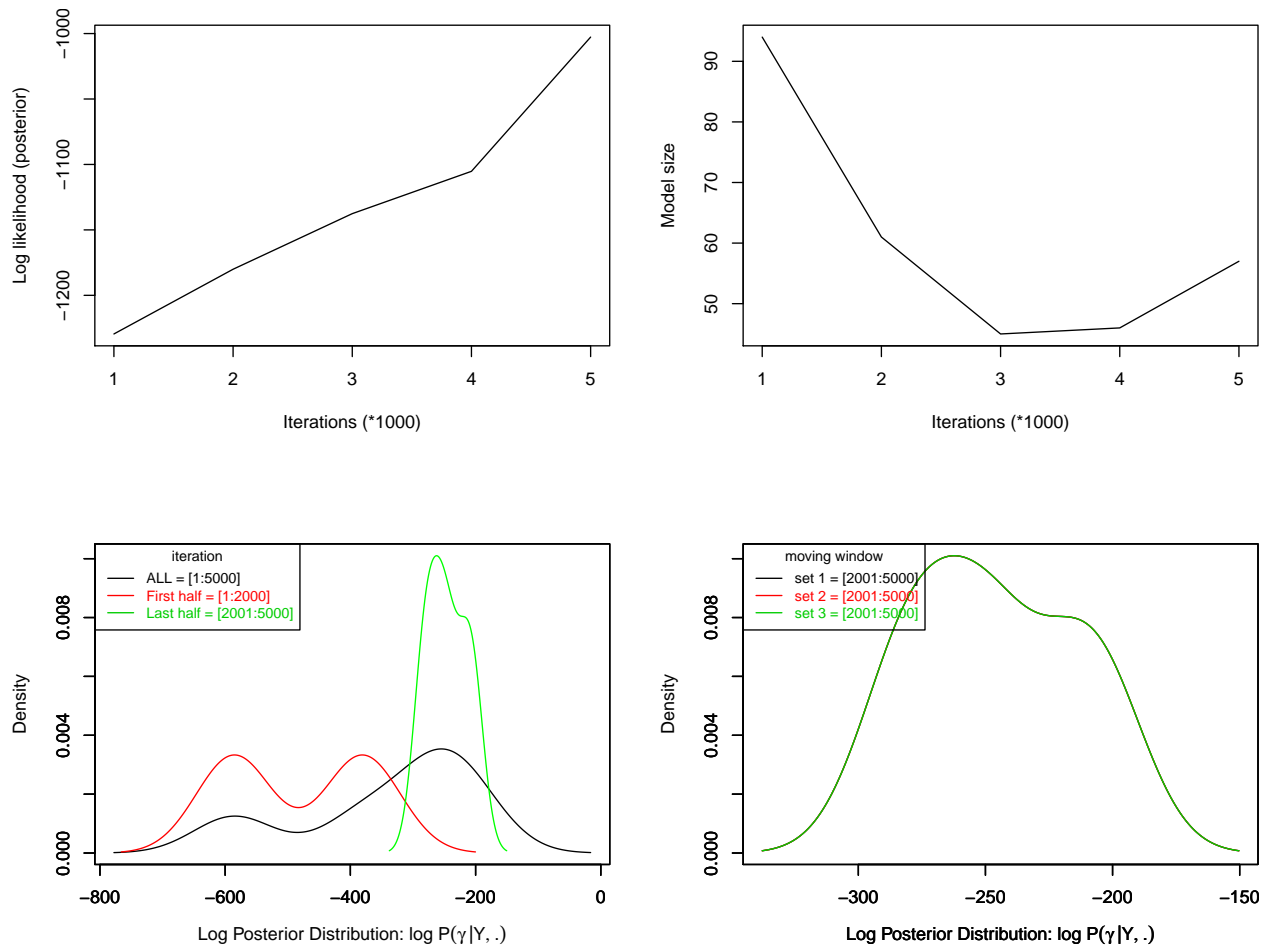


Figure 7: Plots estilo Manhattan, mostrando as probabilidades de inclusão marginal posterior (em cima) e o número de expressões gênicas da resposta associada a cada SNP (embaixo).



Embora não tenhamos acesso direto às cadeias simuladas, podemos também visualizar a verossimilhança das outras matrizes estimadas pelo modelo conforme mostra a Figura 8.

É também possível utilizar o CPO para encontrar observações destoantes, como uma forma de validação cruzada do modelo com a amostra obtida, conforme mostra a Figura 9. Temos algumas poucas observações abaixo do limiar.

Finalizamos nossa análise com um sumário do modelo obtido. Após isso, desatachamos o conjunto de dados utilizado.

Call: BayesSUR(data = data, Y = blockList[[1]], X = blockList[[2]], ...)

CPOs: Min. 1st Qu. Median 3rd Qu. Max. 0.0009178457 0.1934408094 0.3167474795 0.3850252944 0.4727190007

Number of selected predictors (mPIP > 0.5): 46 of 10x150

Top 10 predictors on average mPIP across all responses: SNP91 SNP76 SNP97 SNP135 SNP47 SNP48 SNP98 SNP137 SNP93 SNP72 0.22897 0.19130 0.19060 0.17571 0.16828 0.15678 0.15033 0.14669 0.12817 0.12759

Top 10 responses on average mPIP across all predictors: GEX10 GEX8 GEX7 GEX5 GEX6 GEX2 GEX3 0.04976267 0.04920667 0.04652733 0.04570800 0.04323267 0.03834600 0.03357267 GEX4 GEX9 GEX1 0.03296133 0.01895867 0.01567200

Expected log pointwise predictive density (elpd) estimates: elpd.LOO = -1422.081, elpd.WAIC = -1422.061

MCMC specification: iterations = 5000, burn-in = 1000, chains = 3 gamma local move sampler: bandit gamma initialisation: R

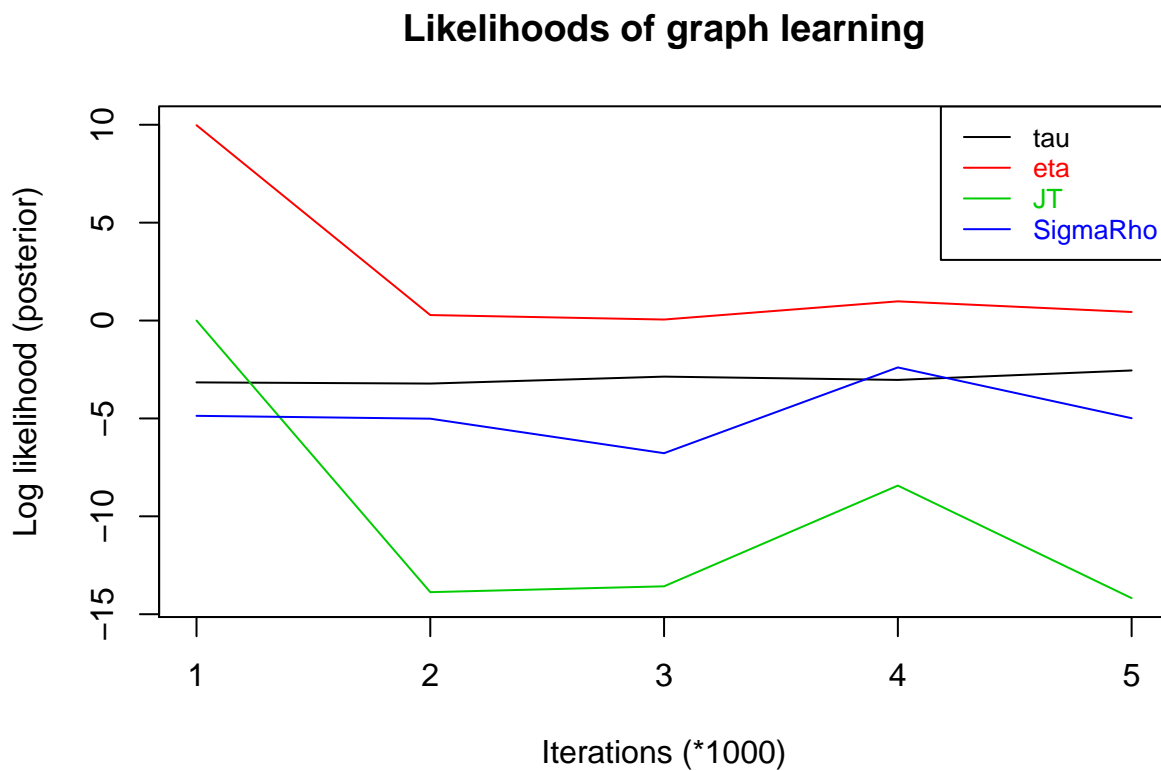


Figure 8: Verossimilhança dos estimadores pelo número de iterações do MCMC.

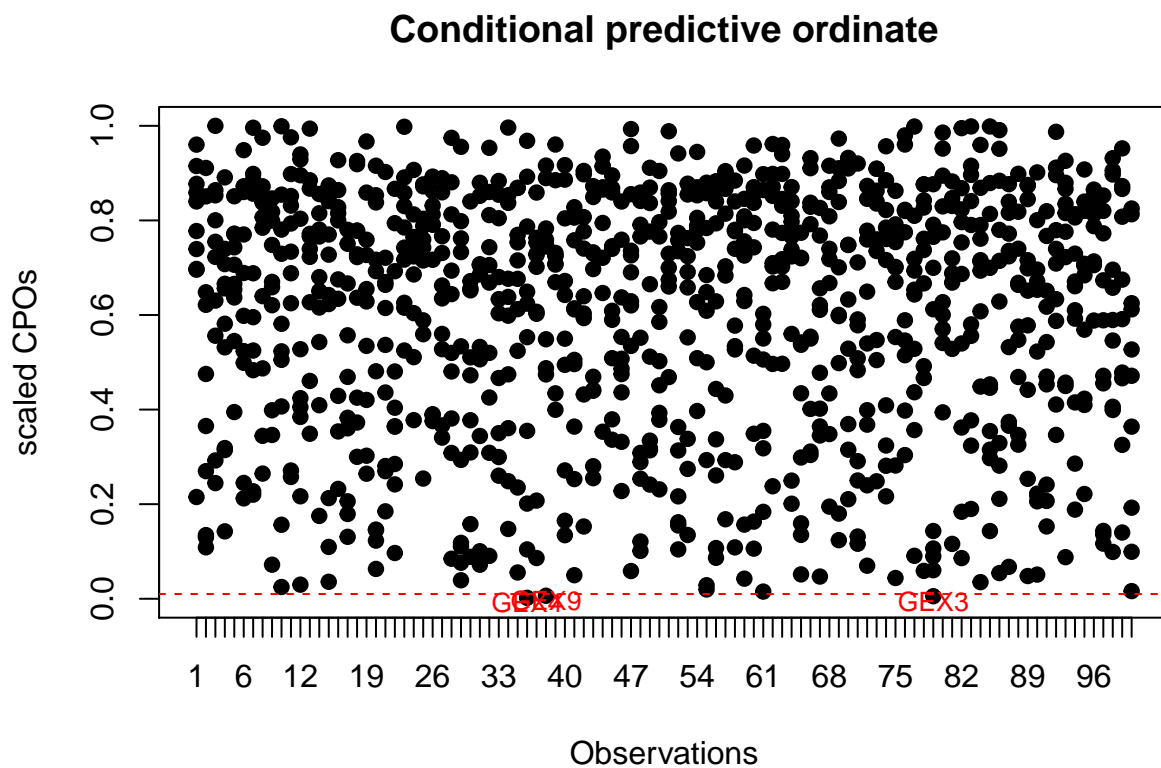


Figure 9: CPOs escalonados para o modelo de regressão ajustado

Model specification: covariance prior: HIW gamma prior: hotspot

Hyper-parameters: a_w b_w a_o b_o a_pi b_pi nu a_tau b_tau a_eta b_eta 2.0 5.0 2.0 148.0 2.0 1.0 12.0 0.1 10.0 0.1 1.0

- **status:** 0

- **elpd:**

elpd.loo	elpd.waic
-1422	-1422

- **CPO:**

Min.	1st Qu.	Median	3rd Qu.	Max.
0.0009178	0.1934	0.3167	0.385	0.4727

- **df:** 46

- **chainParameters:**

- **nIter:** 5000
- **burnin:** 1000
- **nChains:** 3

- **modelParameters:**

- **covariancePrior:** HIW
- **gammaPrior:** hotspot
- **gammaSampler:** bandit
- **gammaInit:** R
- **mrfg:** results/mrfg.txt
- **hyperParameters:**
 - * **a_w:** 2
 - * **b_w:** 5
 - * **a_o:** 2
 - * **b_o:** 148
 - * **a_pi:** 2
 - * **b_pi:** 1
 - * **nu:** 12
 - * **a_tau:** 0.1
 - * **b_tau:** 10
 - * **a_eta:** 0.1
 - * **b_eta:** 1

- **outputFiles:**

- **logP:** results/data_SSUR_logP_out.txt
- **gamma:** results/data_SSUR_gamma_out.txt
- **pi:** results/data_SSUR_pi_out.txt
- **tail:** results/data_SSUR_hotspot_tail_p_out.txt
- **beta:** results/data_SSUR_beta_out.txt

- **Gy**: results/data_SSUR_Gy_out.txt
- **Gvisit**: results/data_SSUR_Gy_visit.txt
- **sigmaRho**: results/data_SSUR_sigmaRho_out.txt
- **model_size**: results/data_SSUR_model_size_out.txt
- **CPO**: results/data_SSUR_CPO_out.txt
- **CPOsumy**: results/data_SSUR_CPOsumy_out.txt
- **WAIC**: results/data_SSUR_WAIC_out.txt
- **Y**: results/data_Y.txt
- **X**: results/data_X.txt

5 Conclusão

6 Referências

Zhao, Zhi, Marco Banterle, Leonardo Bottolo, Sylvia Richardson, Alex Lewin, and Manuela Zucknick. 2021. “BayesSUR: An R Package for High-Dimensional Multivariate Bayesian Variable and Covariance Selection in Linear Regression.” *Journal of Statistical Software* 100 (11): 1–32. <https://doi.org/10.18637/jss.v100.i11>.

7 Apêndice: códigos

```
knitr::opts_chunk$set(echo = TRUE)
# pacote principal
library(BayesSUR)
data("exampleEQTL", package = "BayesSUR")
# attach nos dados para um código mais compacto
attach(exampleEQTL)
# bibliotecas para gráficos
library(tidyverse)
library(gridExtra)
library(ggpubr)
library(tictoc)
# funcao que recebe uma matriz e plota um mapa de calor
plot_heatmap<- function(df){
  reshape2::melt(df) %>% ggplot(aes(x=Var1, y=Var2, fill=-value)) +
    geom_raster() + guides(fill="none") +
    scale_fill_fermenter()
}
# rotulos da escala
labs <- as_labeller(
  c("1" = "GEX1", "2" = "GEX2", "3" = "GEX3", "4" = "GEX4", "5" = "GEX5",
    "6" = "GEX6", "7" = "GEX7", "8" = "GEX8", "9" = "GEX9", "10" = "GEX10"))
# mapa de calor Y versus X
p <- plot_heatmap(gamma) + scale_y_continuous(breaks= 1:10, labels = labs)
# mapa de calor Y versus Y
q <- plot_heatmap(Gy) + scale_x_continuous(breaks= 1:10, labels = labs)

grid.arrange(p,q, ncol = 2)
set.seed(28173)
tic("Tempo de ajuste do modelo")
```

```

# ajuste do modelo
fit <- BayesSUR(data = data, Y = blockList[[1]], X = blockList[[2]],
  outFilePath = "results", nIter = 5000, nChains = 3,
  burnin = 1000, covariancePrior = "HIW",
  gammaPrior = "hotspot",
  output_CPO = TRUE,
)

toc()
# plota estimador beta
plotEstimator(fit, "beta")
# plota estimador gamma
plotEstimator(fit, "gamma")
# plota o grafo das respostas
plotEstimator(fit, "Gy")
# estrutura estimada vs estrutura verdadeira
layout(matrix(1:2, ncol = 2))
plot(fit, estimator = "Gy", type = "graph")
plotGraph(Gy)
# estrutura entre X e Y
plot(fit, estimator = c("gamma", "Gy"), type = "network",
  name.predictors = "SNPs", name.responses = "Gene expression")
plot(fit, estimator = "gamma", type = "Manhattan")
plot(fit, estimator = "logP", type = "diagnostics")
plotMCMCdiag(fit, HIWg = "lik")
# CPO
plotCPO(fit)
# sumario do modelo
pander::pander(summary(fit))
detach(exampleEQTL)

```