



**POLITECNICO**  
MILANO 1863

MSC. IN MATHEMATICAL ENGINEERING A.Y. 2022/2023  
PROJECT REPORT OF BAYESIAN STATISTICS (052499) – PROF. A. GUGLIELMI  
SUPERVISOR: ALESSANDRO COLOMBI

---

# Stochastic Block Model Prior with Ordering Constraints for Gaussian Graphical Models

---


Teo Bucci<sup>1</sup>, Filippo Cipriani<sup>2</sup>, Filippo Pagella<sup>3</sup>, Flavia Petruso<sup>4</sup>, Andrea Puricelli<sup>5</sup>, and  
Giulio Venturini<sup>6</sup>

<sup>1</sup>10621873, <sup>2</sup>10596877, <sup>3</sup>10616351, <sup>4</sup>10544566, <sup>5</sup>10632135, <sup>6</sup>10624098

14th February 2023

## Abstract

Gaussian graphical models are used to study the conditional dependence structure among variables through the presence or absence of edges in the underlying undirected graph. In many applications, the variables can be grouped so that the graph to be learnt from the data has a block structure. Stochastic block models offer a powerful tool to detect such structure in a network. The goal of this project is to propose a new flexible prior that accounts for a random partition of the nodes, respects their ordering constraints and allows to learn a block-structured graph.

The source code of the entire project,  
including this report and the presentations, is available at  
 <https://github.com/teobucci/bayesian-statistics-project>

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Proposed Model</b>	<b>3</b>
<b>3</b>	<b>Sampling strategy</b>	<b>5</b>
3.1	Graph sampling . . . . .	5
3.2	Random partition sampling . . . . .	5
3.2.1	Split and merge move . . . . .	5
3.2.2	Adaptive step . . . . .	6
3.2.3	Shuffle move . . . . .	7
3.3	Updating the hyperparameters . . . . .	7
<b>4</b>	<b>Posterior analysis</b>	<b>7</b>
4.1	Posterior graph . . . . .	7
4.2	Posterior partition . . . . .	8
4.3	Performance indexes: Kullback-Leibler . . . . .	8
4.4	Performance indexes: Rand index . . . . .	8
<b>5</b>	<b>Simulation study</b>	<b>9</b>
5.1	Group 1: changing seed . . . . .	10
5.2	Group 2: changing the variance of the Beta prior . . . . .	11
5.3	Group 3: varying initial partition . . . . .	11
5.4	Group 4: varying generating partition's cluster numerosities . . . . .	12
5.5	Group 5: varying number of observations . . . . .	13
5.6	Group 6: varying number of nodes . . . . .	13
5.7	Group 7: using noised block structure as data generator . . . . .	14
<b>6</b>	<b>Conclusions and further developments</b>	<b>14</b>
<b>A</b>	<b>Mathematical details</b>	<b>15</b>
A.1	Graph ratio split and merge . . . . .	15
A.2	Graph ratio shuffle . . . . .	15
<b>B</b>	<b>Computational complexity</b>	<b>16</b>
<b>C</b>	<b>Glossary</b>	<b>16</b>
	<b>References</b>	<b>17</b>

## 1 Introduction

Gaussian Graphical Models (GGM) are probabilistic models where undirected graphs are used to express the conditional dependence structure among variables with a joint Gaussian distribution. The nodes of the graph represent the variables of the model, whereas the edges model the dependency between them. A crucial concept in GGM is the one of conditional independence: given a  $p$ -dimensional random vector  $\mathbf{Y}$  distributed as a multivariate normal with zero mean and precision matrix  $\mathbf{K}$ , the random variables  $Y_i$  and  $Y_j$  are independent, conditionally on all the others, if and only if the corresponding entry in precision matrix  $\mathbf{K}$  is null. This translates into the absence of an undirected edge linking nodes  $i$  and  $j$  in the graph.

$$Y_i \perp\!\!\!\perp Y_j \mid \mathbf{Y}_{-(ij)} \iff (i, j) \notin E \iff K_{ij} = 0$$

where  $\mathbf{Y}_{-(ij)}$  is the random vector containing all elements in  $\mathbf{Y}$  except the  $i$ -th and the  $j$ -th.

The graph is usually unknown and must be learnt from the data. Following a Bayesian approach, the graph itself is considered a random variable, which can assume values in the space of all possible undirected graphs with  $p$  nodes. A common choice is to place a uniform prior distribution over such space. Furthermore, a prior on the precision matrix conditionally on the graph must also be introduced. Due to its conjugacy property, a G-Wishart distribution is often selected as a prior for the precision matrix.

In many real-life applications, the variables of interest can be grouped into clusters. This is a well-known and studied problem in network analysis, where the network of dependencies is observed and one only aims to study the properties. In such a context, the most widely used model is the Stochastic Block Model (SBM). More in detail, SBM is a generative model for random networks with wide applications in the context of community detection, allowing to cluster the nodes of a graph into mutually exclusive groups, not known a priori, that share similar connectivity patterns. Most importantly, in an SBM, the probability of having an edge between two nodes only depends on the group membership of the nodes. Recently, there is a growing literature which aims to extend this reasoning also to problems where the network is latent, not observed. If so, the idea is to exploit SBM as a prior for the graph to simultaneously perform structural learning and the clustering of the nodes.

One assumption for clustering often made in the literature is that the nodes' names can be relabeled. This is equivalent to assuming that there is a lack of ordering among the variables. However, we argue that this assumption may fail in some contexts, especially those related to spatial statistics, where the localization of the variables imposes a **natural ordering constraint**. For instance, several biological problems require the identification of groups of genes which regulate specific cellular functions; in this case, respecting the order of alignment of genomic loci along the chromosome might yield additional important information. Another situation where the variables' order may play a significant role is imaging, where any reasonable grouping of variables should consider the underlying 3-D structure of data.

In this project, we propose a flexible prior that allows us to infer a block-structured graph while respecting an ordering constraint on the nodes. To do so, we build upon the theory of changepoint models from the works of Benson and Friel (2018) and Martínez and Mena (2014), borrowing ideas concerning the prior on the partition and relying on an adaptive approach.

In the next two sections, we introduce the proposed model and describe the sampling strategy, which relies on a two-block Gibbs sampling to update the graph and the partition. Then, we present the main results of our simulations. Finally, we perform a critical analysis and discuss the limitations of the current work while also identifying potential directions for future investigation.

## 2 Proposed Model

We consider an independent and identically distributed sample of size  $n$  with  $p$  variables from a multivariate normal distribution. Without any loss of generality, we assume that the distribution has zero-mean and precision matrix  $\mathbf{K}$ . As anticipated, each variable corresponds to a node on the underlying graph.

As concerns the group memberships, we adopt three equivalent ways to express the partition, depending on the aspect that is most convenient and we wish to highlight. In the following, we will always denote by  $M$  the number of groups the nodes have been partitioned into, which is not known a priori, with  $C_1, \dots, C_M$  the groups, with  $n_j = |C_j|$  and with  $p$  the number of nodes (i.e. number of variables).

- $\boldsymbol{\rho} = (|C_1|, \dots, |C_M|) = (n_1, \dots, n_M)$  is the vector of groups cardinalities. Since we have a constraint on the ordering of the nodes, it's enough to specify the cardinalities of each group.
- $\mathbf{z} = (z_1, \dots, z_p)$  is a  $p$ -dimensional vector of memberships,  $z_j = m$  if node  $j$  belongs to group  $C_m$
- $\mathbf{r} = (r_1, \dots, r_{p-1})$  is a  $(p-1)$ -dimensional vector in which  $r_j = 1$  if node  $j$  is the right end of a group, and 0 otherwise. The last node is, by definition, always the end of a group, thus by convention can be thought of as 1.

For example, when  $p = 8$  and  $M = 4$ , let us consider the partition

$$(C_1, C_2, C_3, C_4) = (\{1\}, \{2, 3, 4\}, \{5, 6\}, \{7, 8\})$$

then we have

$$\boldsymbol{\rho} = (1, 3, 2, 2) \iff \mathbf{z} = (1, 2, 2, 2, 3, 3, 4, 4) \iff \mathbf{r} = (1, 0, 0, 1, 0, 1, 0)$$

Our goal is to simultaneously infer the conditional dependence structure of such variables and their clustering, keeping in mind that the partition on the nodes must respect the original order of the variables. We place a G-Wishart prior distribution for the precision matrix  $\mathbf{K}$ , and we rely on SBM for the prior on  $\mathbf{G}$ , which is identified by its adjacency matrix.

In SBM, the probability of having an edge from node  $i$  to node  $j$  only depends on their group membership. The probability that edge  $(i, j)$  belongs to the set of edges  $E$  is

$$P((i, j) \in E \mid \mathbf{z}, \mathbf{Q}) = Q_{z_i, z_j} \quad i, j = 1, \dots, p \quad \text{independent}$$

where  $\mathbf{Q}$  is a symmetric probability matrix and  $Q_{uv}$  is the probability of having an edge between any node in cluster  $u$  and any other node in cluster  $v$ . To exploit conjugacy, we place a Beta prior on  $\mathbf{Q} \mid \mathbf{z}$

$$Q_{uv} \mid \mathbf{z} \stackrel{\text{iid}}{\sim} \text{Beta}(\alpha, \beta), \quad 1 \leq u \leq v \leq M$$

The Beta distribution is conjugate to the prior for  $\mathbf{G}$ , and can be integrated out. After this step, the prior for the graph conditionally on the vector of group membership reduces to

$$P(\mathbf{G} \mid \mathbf{z}) = \prod_{u=1}^M \prod_{v=u}^M \frac{B(\alpha + S_{uv}, \beta + S_{uv}^*)}{B(\alpha, \beta)} \quad (1)$$

$S_{uv}$  is the number of existing edges between the nodes from cluster  $u$  and the nodes from cluster  $v$  and  $S_{uv}^*$  is the number of all possible edges between the clusters  $u$  and  $v$ , minus the already existing ones.

Given this model, our goal is to propose a prior distribution for  $\mathbf{z}$  which accounts for the ordering constraint on the nodes, assigning a probability law over the space of admissible partitions. Choosing a law for  $\mathbf{z}$  is therefore equivalent to specifying a law for  $\boldsymbol{\rho}$ , which will be the focus of the next paragraphs. From this Section onwards, we will switch from one representation to the other accordingly to the context. To introduce an ordering constraint on the law of  $\boldsymbol{\rho}$ , we need to restrict to the space of partition to the admissible ones. We define  $\boldsymbol{\rho}$  as admissible if, for each  $i < j$ , each element of  $C_i$  is strictly smaller than each element of  $C_j$ .

As a prior for our model, we use the Exchangeable Partition Probability Function (EPPF) induced by the two-parameter Poisson-Dirichlet process (Pitman-Yor process) from Martínez and Mena (2014, p. 830):

$$P(\boldsymbol{\rho} = (n_1, \dots, n_M)) = \begin{cases} \frac{p!}{M!} \frac{\prod_{i=1}^{M-1} (\theta + i\sigma)}{(\theta+1)_{(p-1)\uparrow}} \prod_{j=1}^M \frac{(1-\sigma)_{(n_j-1)\uparrow}}{n_{j\uparrow}}, & \boldsymbol{\rho} \text{ admissible} \\ 0, & \boldsymbol{\rho} \text{ not admissible.} \end{cases} \quad (2)$$

where  $x_{n\uparrow}$  is the rising factorial (or Pochhammer function), namely

$$x_{n\uparrow} = \overbrace{x(x+1)(x+2) \cdots (x+n-1)}^{n \text{ factors}} \quad x_{0\uparrow} = 1.$$

$\theta$  and  $\sigma$  are hyperparameters such that  $\sigma \in [0, 1)$  with  $\theta > -\sigma$  or  $\sigma < 0$  with  $\theta = m|\sigma|$  for some positive integer  $m$ . We will work with the case  $\sigma \in [0, 1)$ .

We can rewrite the model as

$$\begin{aligned} \mathbf{Y}_1, \dots, \mathbf{Y}_n \mid \mathbf{K} &\stackrel{\text{iid}}{\sim} \mathcal{N}_p(0, \mathbf{K}^{-1}) \\ \mathbf{K} \mid \mathbf{G} &\sim \text{G-Wishart}(b, D) \\ \mathbf{G} \mid \mathbf{z} &\sim P(\mathbf{G} \mid \mathbf{z}) \\ \boldsymbol{\rho} &\sim P(\boldsymbol{\rho}) \end{aligned}$$

where  $P(\mathbf{G} \mid \mathbf{z})$  is given by (1) and  $P(\boldsymbol{\rho})$  is given by (2).

### 3 Sampling strategy

We propose a Block Gibbs sampling strategy divided into two steps:

1. sampling of the graph  $\mathbf{G}$  and the precision matrix  $\mathbf{K}$ , conditionally to the partition  $\mathbf{z}$  and the data  $\mathbf{Y}$ ,
2. sampling of the partition  $\mathbf{z}$ , conditionally to the graph  $\mathbf{G}$ , the precision matrix  $\mathbf{K}$  and the data  $\mathbf{Y}$ .

#### 3.1 Graph sampling

The conditional distribution used to sample the graph is:

$$P(\mathbf{K}, \mathbf{G} \mid \mathbf{Y}, \mathbf{z}) \propto P(\mathbf{Y} \mid \mathbf{K})P(\mathbf{K} \mid \mathbf{G})P(\mathbf{G} \mid \mathbf{z})$$

Given the membership vector  $\mathbf{z}$ , a Birth-and-Death approach is used to sample the graph, as suggested in the work of Mohammadi and Wit (2015). The Birth-and-Death algorithm decides at every iteration of the Gibbs sampling whether to add a new edge to the graph (birth) or delete an already existing one (death). For this purpose, we modified the R and C++ package **BDgraph** to take into account the dependency from membership vector  $\mathbf{z}$ , updating the target distribution and the birth and death rates as follows:

$$\text{Birth rate} \propto \frac{P(\mathbf{G}^{+e} \mid \mathbf{z})}{P(\mathbf{G} \mid \mathbf{z})} = \frac{S_{uv} + \alpha}{S_{uv}^* + \beta} \quad \text{Death rate} \propto \frac{P(\mathbf{G}^{-e} \mid \mathbf{z})}{P(\mathbf{G} \mid \mathbf{z})} = \frac{S_{uv}^* + \beta}{S_{uv} + \alpha}$$

where  $\alpha$  and  $\beta$  are the parameters of the Beta distribution of  $Q_{uv} \mid \mathbf{z}$  and where  $\mathbf{G}^{\pm e}$  denotes graph  $\mathbf{G}$  with the added/removed edge  $e$ .

#### 3.2 Random partition sampling

After the graph update, the random partition is sampled conditionally on the graph  $\mathbf{G}$  using the conditional distribution:

$$P(\mathbf{z} \mid \mathbf{Y}, \mathbf{K}, \mathbf{G}) \propto P(\mathbf{Y} \mid \mathbf{K})P(\mathbf{K} \mid \mathbf{G})P(\mathbf{G} \mid \mathbf{z})P(\mathbf{z}) \propto P(\mathbf{G} \mid \mathbf{z})P(\mathbf{z})$$

To sample from  $\mathbf{z}$ , an adaptive split and merge was built from scratch using R.

##### 3.2.1 Split and merge move

Suppose the current partition at iteration  $t$  is  $\mathbf{z}$ . We propose a new candidate partition  $\mathbf{z}'$  and we either accept it or reject it using Metropolis-Hastings and update the partition accordingly for the next iteration  $t + 1$ .

The proposal distribution  $Q(\mathbf{z}, \mathbf{z}')$  exploits the property that the nodes are ordered. Depending on the move, we choose with some probability a group from the current partition to be split into two groups or two groups to be merged into one. To this extent, we consider the  $\mathbf{r}$  representation of the partition.

Unless we are forced by extreme cases to choose a split move (*i.e.*, all the nodes belong to the same group) or a merge move (*i.e.*, all the nodes are in their own single-node group), the algorithm works as follows.

1. Perform either a split or a merge move, with probability  $\alpha_{\text{split}}$  and  $1 - \alpha_{\text{split}}$ , respectively, usually set to 0.5.
2. For a split (merge) move, consider all the 0 (1) only in the partition  $\mathbf{r}$ , one of which will be drawn as a candidate to become a 1 (0), thus splitting a group into two (merging two groups into one). The draw of such a candidate is made according to two weights vectors as explained in 3.2.1.1.
3. Accept or reject using Metropolis-Hastings. The acceptance probability is the minimum between 1 and the product of the graph ratio, the partition ratio, and the proposal ratio. Namely

$$\alpha_{\text{accept}} = \min \left\{ 1, \underbrace{\frac{P(\mathbf{G} \mid \mathbf{z}')}{P(\mathbf{G} \mid \mathbf{z})}}_{\text{graph ratio}} \underbrace{\frac{P(\mathbf{z}')}{P(\mathbf{z})}}_{\text{partition ratio}} \underbrace{\frac{Q(\mathbf{z}', \mathbf{z})}{Q(\mathbf{z}, \mathbf{z}')}}_{\text{proposal ratio}} \right\} \quad (3)$$

**3.2.1.1 Proposal distribution** Using Benson and Friel (2018) approach we introduce two  $(p-1)$ -dimensional vectors iteration-dependent

$$\mathbf{a}^{(t)} = (a_1^{(t)}, \dots, a_{p-1}^{(t)}) \quad \mathbf{d}^{(t)} = (d_1^{(t)}, \dots, d_{p-1}^{(t)})$$

where

- $a_j^{(t)}$  is the probability that node  $j$  is chosen as a candidate for splitting a group at iteration  $t$
- $d_j^{(t)}$  is the probability that node  $j$  is chosen as a candidate for merging a group at iteration  $t$

They are unnormalized discrete densities that are used to choose the node to perform the split or the merge, namely where to add or remove a 1 from the  $\mathbf{r}$  partition representation.

For example, supposing a splitting move, the probability of drawing node  $i$  for the split is proportional to its weight  $a_i^{(t)}$ .

We then introduce:

$$a^* = \sum_{j:r_j=0} a_j^{(t)} \quad d^* = \sum_{j:r_j=1} d_j^{(t)}$$

The proposal ratio in (3) becomes

$$\frac{Q(\mathbf{z}', \mathbf{z})}{Q(\mathbf{z}, \mathbf{z}')} = \frac{P(\text{choose merge})}{P(\text{choose split})} \cdot \frac{P(\text{merge at node } i)}{P(\text{split at node } i)} = \frac{1 - \alpha_{\text{split}}}{\alpha_{\text{split}}} \cdot \frac{\frac{d_i^{(t)}}{d^* + d_i^{(t)}}}{\frac{a_i^{(t)}}{a^*}}$$

The first term is the ratio of the probabilities of choosing one move over the other. In the second ratio, we have at the denominator the probability of choosing exactly the node  $i$  for the split, and at the numerator the probability of going back after the split by choosing the same node  $i$  for a merge.

**3.2.1.2 Target ratio** In the split case, after simplifying common factors, the partition ratio in (3) is suitably expressed in the  $\boldsymbol{\rho}$  representation:

$$\frac{P(\mathbf{z}')}{P(\mathbf{z})} = \frac{1}{M} (\theta + M\sigma) \frac{(1-\sigma)_{(n'_s-1)\uparrow} (1-\sigma)_{(n'_s+1)\uparrow}}{(1-\sigma)_{(n_s-1)\uparrow}} \frac{n_s!}{n'_s! n'_{s+1}!}$$

As for the graph ratio in (3), the likelihood is given by

$$P(\mathbf{G} | \mathbf{z}) = \prod_{l=1}^M \prod_{l=m}^M \frac{B(\alpha + S_{uv}, \beta + S_{uv}^*)}{B(\alpha, \beta)} = \left( \frac{1}{B(\alpha, \beta)} \right)^{\frac{M(M+1)}{2}} \prod_{l=1}^M \prod_{l=m}^M B(\alpha + S_{uv}, \beta + S_{uv}^*)$$

It's enough to compute it both with the current partition  $\mathbf{z}$  and with the proposed one  $\mathbf{z}'$  and compute the ratio simplifying common factors. For further details about the resulting expression see A.1.

### 3.2.2 Adaptive step

The adaptive step consists of updating the two weights vectors  $\mathbf{a}^{(t)}$  (in case of a split move) and  $\mathbf{d}^{(t)}$  (in case of a merge move) at each iteration  $t$  as in Benson and Friel (2018) using the following scheme:

- If a split move at node  $i$  has been accepted, then update:

$$\log(a_i^{(t+1)}) = \log(a_i^{(t)}) + \frac{h}{t/p} (\alpha_{\text{split}} - \alpha_{\text{target}}).$$

- If a merge move at node  $i$  has been accepted, then update:

$$\log(d_i^{(t+1)}) = \log(d_i^{(t)}) + \frac{h}{t/p} (\alpha_{\text{merge}} - \alpha_{\text{target}}).$$

Where  $h > 0$  is the initial adaptation,  $t/p$  are the iterations ( $t$ ) per number of nodes ( $p$ ),  $\alpha_{\text{target}}$  is the target Metropolis-Hastings acceptance rate, and  $\alpha_{\text{merge}} = 1 - \alpha_{\text{split}}$ .

### 3.2.3 Shuffle move

After the split and merge step we perform a shuffle move to improve the mixing of the chain. The shuffle proposes a new partition by moving a certain number of nodes from a group to an adjacent one. Specifically, if  $M > 1$ :

1. choose  $j$  uniformly from  $\{1, \dots, M-1\}$ , the group to be shuffled with the  $(j+1)$ -th;
2. choose  $\ell$  uniformly from  $\{1, \dots, n_j + n_{j+1} - 1\}$  the number of nodes to keep in the  $j$ -th group and set the proposed random partition as

$$\boldsymbol{\rho}' = (n_1, \dots, n_{j-1}, \ell, n_j + n_{j+1} - \ell, \dots, n_M)$$

3. accept or reject using Metropolis-Hastings. Since the proposal is a Uniform, the proposal ratio is 1, thus in the acceptance probability we only have the target ratio

$$\alpha_{\text{shuffle}} = \min \left\{ 1, \frac{P(\mathbf{z}' | \mathbf{G})}{P(\mathbf{z} | \mathbf{G})} \right\} = \min \left\{ 1, \frac{P(\mathbf{G} | \mathbf{z}')}{P(\mathbf{G} | \mathbf{z})} \frac{P(\mathbf{z}')}{P(\mathbf{z})} \right\} \quad (4)$$

In this case, the number of groups  $M$  doesn't change. The first ratio in (4) can be found in A.2. The second ratio in (4), after simplifying common factors, is

$$\frac{P(\mathbf{z}')}{P(\mathbf{z})} = \frac{(1-\sigma)_{(\ell)\uparrow} (1-\sigma)_{(n_s+n_{s+1}-\ell)\uparrow}}{(1-\sigma)_{(n_s-1)\uparrow} (1-\sigma)_{(n_{s+1}-1)\uparrow}} \cdot \frac{n_s! n_{s+1}!}{\ell! (n_s + n_{s+1} - \ell)!}$$

### 3.3 Updating the hyperparameters

Update  $\theta$  and  $\sigma$  in (2) according to Martínez and Mena (2014, pp. 835–836).

## 4 Posterior analysis

### 4.1 Posterior graph

To select the posterior graph, a common approach is to choose the graph with the highest posterior probability. However, this method can be unreliable due to the large number of possible graphs and low frequency of occurrence of the same graph in a MCMC sampling. Our solution is to estimate the marginal posterior probabilities for each edge inclusion, which is calculated as:

$$\hat{p}_{jk} = \frac{\sum_{t=1}^T \mathbb{1}_{((j,k) \in E_t)} w(\mathbf{G}_t)}{\sum_{t=1}^T w(\mathbf{G}_t)}$$

where  $\mathbb{1}_{((j,k) \in E_t)}$  is the indicator function for the existence of the edge linking node  $j$  and node  $k$  at iteration  $t$  and  $w(\mathbf{G}_t)$  is the graph weight (*holding time*) at iteration  $t$ . Then the adjacency matrix of the graph is selected based on the edges with posterior probabilities greater than a given threshold  $s$ . Two different thresholds can be considered:

- $s = 0.5$ , similar to the median probability model proposed by Barbieri and Berger (2004);
- the Bayesian False Discovery rate (BFDR; Müller, Parmigiani and Rice, 2007)

$$\text{BFDR} = \frac{\sum_{j < k} (1 - \hat{p}_{jk}) \mathbb{1}_{(\hat{p}_{jk} \geq s)}}{\sum_{j < k} \mathbb{1}_{(\hat{p}_{jk} \geq s)}}$$

where  $s$  is selected so that BFDR is below 0.05.

Generally, we preferred the second criterion.

## 4.2 Posterior partition

We can obtain point estimates of the partition by exploiting a decision theoretic approach based on a specific loss function, solving an optimization problem as

$$\hat{\rho} = \underset{\rho \in \mathcal{C}}{\operatorname{argmin}} \mathbb{E}[L(\tilde{\rho}, \rho) \mid \mathbf{Y}_1, \dots, \mathbf{Y}_n] = \underset{\rho \in \mathcal{C}}{\operatorname{argmin}} \sum_{\rho^* \in \mathcal{C}} L(\rho^*, \rho) \underbrace{P(\tilde{\rho} = \rho^* \mid \mathbf{Y}_1, \dots, \mathbf{Y}_n)}_{\text{posterior similarity matrix}},$$

where  $\tilde{\rho}$  is the *true* partition,  $\mathcal{C}$  is the space of all possible partitions, and  $L(\cdot, \cdot) : \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}$  denotes a loss function.

Different choices can be made for the loss function, such as the Binder loss function. Here, we resort to the Variation of Information (VI) loss function (Meilă, 2007).

It is unfeasible to scan the entire space  $\mathcal{C}$ . We restrict the optimization problem to a sub-optimal solution within the space  $\mathcal{C}^T \subseteq \mathcal{C}$  of the orders visited in  $T$  steps of the MCMC sampling.

## 4.3 Performance indexes: Kullback-Leibler

We used the Kullback-Leibler (KL) distance to compare the inferred precision matrix from **BDgraph** with the generating precision matrix during the simulations with synthetic data. The definition of KL distance is as follows.

Suppose that we have two multivariate normal distributions, with means  $\mu_0, \mu_1$  and with (non-singular) covariance matrices  $\Sigma_0, \Sigma_1$ . If the two distributions have the same dimension,  $k$ , then the relative entropy between the distributions is as follows:

$$D_{\text{KL}}(\mathcal{N}_0 \parallel \mathcal{N}_1) = \frac{1}{2} \left( \operatorname{tr}(\Sigma_1^{-1} \Sigma_0) - k + (\mu_1 - \mu_0)^\top \Sigma_1^{-1} (\mu_1 - \mu_0) + \ln \left( \frac{\det \Sigma_1}{\det \Sigma_0} \right) \right)$$

## 4.4 Performance indexes: Rand index

We used the Rand index (RI) to compare the estimated partition with the generating partition during the simulations with synthetic data. The RI is defined as follows.

Given a set of  $n$  elements  $S = \{o_1, \dots, o_n\}$  and two partitions of  $S$  to compare,  $X = \{X_1, \dots, X_r\}$ , a partition of  $S$  into  $r$  subsets, and  $Y = \{Y_1, \dots, Y_s\}$ , a partition of  $S$  into  $s$  subsets, define the following:

- $a$ , the number of pairs of elements in  $S$  that are in the same subset in  $X$  and in the same subset in  $Y$
- $b$ , the number of pairs of elements in  $S$  that are in different subsets in  $X$  and in different subsets in  $Y$
- $c$ , the number of pairs of elements in  $S$  that are in the same subset in  $X$  and in different subsets in  $Y$
- $d$ , the number of pairs of elements in  $S$  that are in different subsets in  $X$  and in the same subset in  $Y$

The Rand index,  $R$ , is:

$$R = \frac{a + b}{a + b + c + d} = \frac{a + b}{\binom{n}{2}}$$

Intuitively,  $a + b$  can be considered as the number of agreements between  $X$  and  $Y$  and  $c + d$  as the number of disagreements between  $X$  and  $Y$ .



## 5 Simulation study

We ran a total of 41 simulations, varying different hyperparameters, with 10000 iterations, of 2000 discarded as burnin, and  $\alpha_{\text{target}} = 0.234$ . The Beta was reparametrized with mean and variance, with mean set to the graph density. Simulations were divided into groups, depending on the parameter tuned, and posterior analysis was carried out on each simulation using the same plots and metrics:

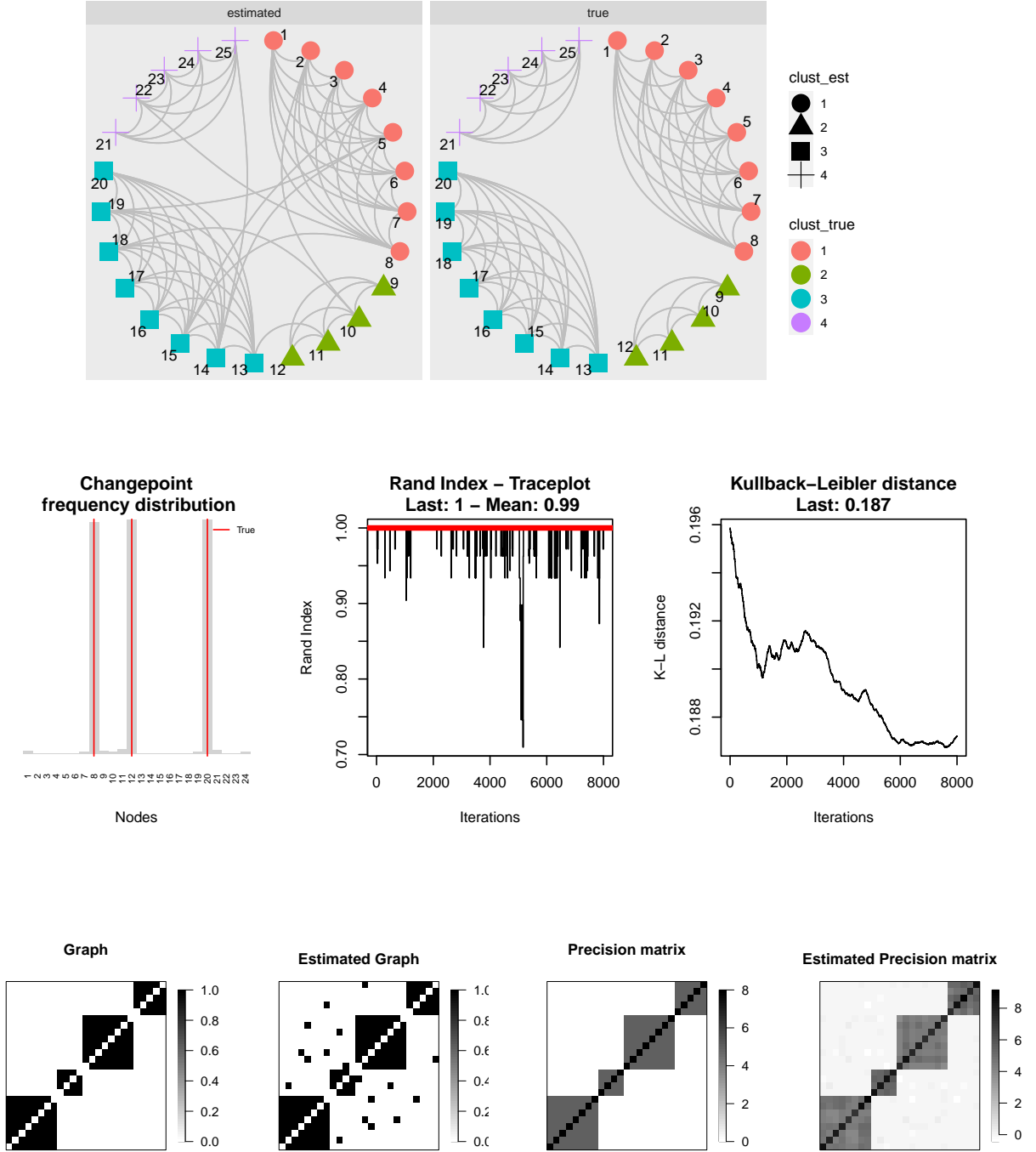
- $\rho$  estimated as in 4.2
- Mean acceptance rate
- RI to measure the similarity between partitions as in 4.4
- KL distance to compare estimated and generating precision matrices as in 4.3
- Plots of the posterior adjacency matrix as in 4.1
- Execution time

Data								Analysis						
sim_id	$n$	$p$	data_gen	seed	$\rho_0$	beta_sig2	$\rho_{\text{true}}$	$\rho_{\text{est}}$	accept	VI	RI	KL	time	
Group 1: varying seed														
01	500	25	BD	22111996	25	0.2	8,4,8,5	8,4,8,5	0.019	0.017	1.000	0.187	1.21400 mins	
02	500	25	BD	31051999	25	0.2	8,4,8,5	8,4,8,5	0.019	0.017	1.000	0.187	1.21800 mins	
03	500	25	BD	27051999	25	0.2	8,4,8,5	8,4,8,5	0.019	0.017	1.000	0.187	1.21700 mins	
04	500	25	BD	29061999	25	0.2	8,4,8,5	8,4,8,5	0.019	0.017	1.000	0.187	1.21700 mins	
05	500	25	BD	12091997	25	0.2	8,4,8,5	8,4,8,5	0.019	0.017	1.000	0.187	1.21600 mins	
06	500	25	BD	27091999	25	0.2	8,4,8,5	8,4,8,5	0.019	0.017	1.000	0.187	1.21500 mins	
07	500	25	BD	27121996	25	0.2	8,4,8,5	8,4,8,5	0.019	0.017	1.000	0.187	1.21700 mins	
Group 2: varying beta_sig2														
08	500	25	BD	27121996	25	0.062	8,4,8,5	8,4,8,5	0.066	0.034	1.000	0.191	1.22400 mins	
09	500	25	BD	27121996	25	0.078	8,4,8,5	8,4,8,5	0.080	0.043	1.000	0.186	1.20700 mins	
10	500	25	BD	27121996	25	0.093	8,4,8,5	8,4,8,5	0.075	0.039	1.000	0.188	1.20600 mins	
11	500	25	BD	27121996	25	0.108	8,4,8,5	8,4,8,5	0.079	0.049	1.000	0.188	1.20800 mins	
12	500	25	BD	27121996	25	0.124	8,4,8,5	8,4,8,5	0.067	0.040	1.000	0.188	1.20100 mins	
13	500	25	BD	27121996	25	0.139	8,4,8,5	8,4,8,5	0.060	0.035	1.000	0.188	1.20200 mins	
14	500	25	BD	27121996	25	0.154	8,4,8,5	8,4,8,5	0.056	0.030	1.000	0.193	1.20000 mins	
15	500	25	BD	27121996	25	0.169	8,4,8,5	8,4,8,5	0.047	0.024	1.000	0.191	1.20100 mins	
16	500	25	BD	27121996	25	0.185	8,4,8,5	8,4,8,5	0.034	0.022	1.000	0.189	1.20000 mins	
17	500	25	BD	27121996	25	0.2	8,4,8,5	8,4,8,5	0.019	0.017	1.000	0.187	1.16200 mins	
Group 3: varying initial partition														
18	500	25	BD	27121996	25	0.2	8,4,8,5	8,4,8,5	0.019	0.017	1.000	0.187	1.16300 mins	
19	500	25	BD	27121996	singletons	0.2	8,4,8,5	8,4,8,5	0.020	0.015	1.000	0.189	1.24800 mins	
Group 4: varying group numerosities														
20	500	25	BD	27121996	25	0.2	8,4,8,5	8,4,8,5	0.019	0.017	1.000	0.187	1.16000 mins	
21	500	25	BD	27121996	25	0.2	1,10,2,9,3	1,10,2,9,3	0.072	0.109	1.000	0.181	1.25500 mins	
22	500	25	BD	27121996	25	0.02	1,3,2,4,2,3,3,4,3	18,7	0.397	1.020	0.129	0.126	1.10700 mins	
23	500	25	BD	27121996	25	0.2	12,13	12,13	0.129	0.073	1.000	0.255	1.36700 mins	
Group 5: varying $n$														
24	500	25	BD	27121996	25	0.1	8,4,8,5	8,4,8,5	0.082	0.046	1.000	0.187	1.17300 mins	
25	400	25	BD	27121996	25	0.1	8,4,8,5	8,4,8,5	0.072	0.036	1.000	0.221	1.15900 mins	
26	300	25	BD	27121996	25	0.1	8,4,8,5	8,4,8,5	0.070	0.041	1.000	0.336	1.16800 mins	
27	200	25	BD	27121996	25	0.1	8,4,8,5	8,4,8,5	0.074	0.045	1.000	0.532	1.15400 mins	
28	100	25	BD	27121996	25	0.1	8,4,8,5	8,4,8,5	0.111	0.241	1.000	1.487	1.11800 mins	
29	50	25	BD	27121996	25	0.1	8,4,8,5	20,5	0.236	0.539	0.273	4.212	1.09000 mins	
30	20	25	BD	27121996	25	0.1	8,4,8,5	25	0.280	0.241	0.000	11.854	1.07600 mins	
Group 6: varying $p$														
31	500	5	BD	27121996	5	0.0625	3,2	3,2	0.543	0.752	1.000	0.006	0.79960 mins	
32	500	10	BD	27121996	10	0.0625	5,5	5,5	0.188	0.115	1.000	0.028	0.85055 mins	
33	500	15	BD	27121996	15	0.0625	5,5,5	5,5,5	0.129	0.087	1.000	0.045	1.09400 mins	
34	500	20	BD	27121996	20	0.0625	5,5,5,5	5,5,5,5	0.103	0.070	1.000	0.118	1.20100 mins	
35	500	25	BD	27121996	25	0.0625	5,5,5,5,5	5,5,5,5,5	0.104	0.074	1.000	0.170	1.33600 mins	
36	500	30	BD	27121996	30	0.0625	5,5,5,5,5,5	5,5,5,5,5,5	0.080	0.061	1.000	0.140	1.49600 mins	
37	500	35	BD	27121996	35	0.0625	5,5,5,5,5,5,5	5,5,5,5,5,5,5	0.066	0.149	1.000	0.205	1.71700 mins	
38	500	40	BD	27121996	40	0.0625	5,5,5,5,5,5,5,5	5,5,5,5,5,5,5,5	0.061	0.384	1.000	0.282	1.99900 mins	
39	500	45	BD	27121996	45	0.0625	5,5,5,5,5,5,5,5,5	5,5,5,5,5,10,10	0.071	0.761	0.756	0.396	2.45600 mins	
40	500	50	BD	27121996	50	0.0625	5,5,5,5,5,5,5,5,5,5	25,5,5,5,5,5	0.059	0.853	0.364	0.367	3.05700 mins	
Group 7: using noised block structure														
41	500	25	B	27121996	25	0.2	8,4,8,5	8,4,8,5	0.013	0.104	1.000	0.176	1.03100 mins	

### 5.1 Group 1: changing seed

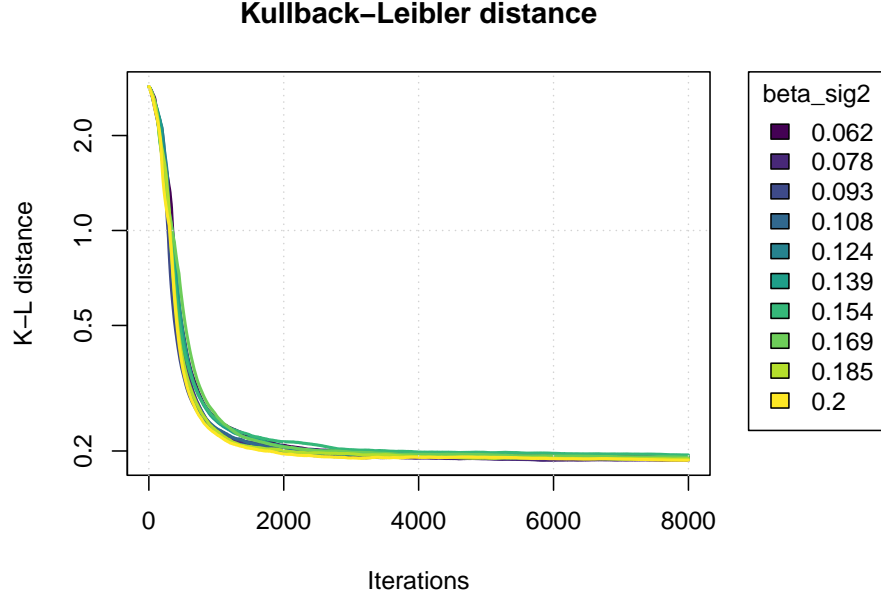
A single full posterior analysis is reported below as example. For all the following groups, only relevant plots highlighting differences are included.

Results are consistent for every seed, with an estimated adjacency matrix of the graph close to the generating one. The estimated partition coincide with the generating partition, and the KL distances converge to 0.187 for all the simulations in the group.



## 5.2 Group 2: changing the variance of the Beta prior

The results for group 2 are similar to the ones of the first group. Changing the variance of the Beta prior does not yield significant changes in posterior analysis. The only notable difference is that an increase in variance yields a decrease in mean acceptance rate.



Data								Analysis					
sim_id	<i>n</i>	<i>p</i>	data_gen	seed	$\rho_0$	beta_sig2	$\rho_{\text{true}}$	$\rho_{\text{est}}$	accept	VI	RI	KL	time
08	500	25	BD	27121996	25	0.062	8,4,8,5	8,4,8,5	0.066	0.034	1.000	0.191	1.22400 mins
09	500	25	BD	27121996	25	0.078	8,4,8,5	8,4,8,5	0.080	0.043	1.000	0.186	1.20700 mins
10	500	25	BD	27121996	25	0.093	8,4,8,5	8,4,8,5	0.075	0.039	1.000	0.188	1.20600 mins
11	500	25	BD	27121996	25	0.108	8,4,8,5	8,4,8,5	0.079	0.049	1.000	0.188	1.20800 mins
12	500	25	BD	27121996	25	0.124	8,4,8,5	8,4,8,5	0.067	0.040	1.000	0.188	1.20100 mins
13	500	25	BD	27121996	25	0.139	8,4,8,5	8,4,8,5	0.060	0.035	1.000	0.188	1.20200 mins
14	500	25	BD	27121996	25	0.154	8,4,8,5	8,4,8,5	0.056	0.030	1.000	0.193	1.20000 mins
15	500	25	BD	27121996	25	0.169	8,4,8,5	8,4,8,5	0.047	0.024	1.000	0.191	1.20100 mins
16	500	25	BD	27121996	25	0.185	8,4,8,5	8,4,8,5	0.034	0.022	1.000	0.189	1.20000 mins
17	500	25	BD	27121996	25	0.2	8,4,8,5	8,4,8,5	0.019	0.017	1.000	0.187	1.16200 mins

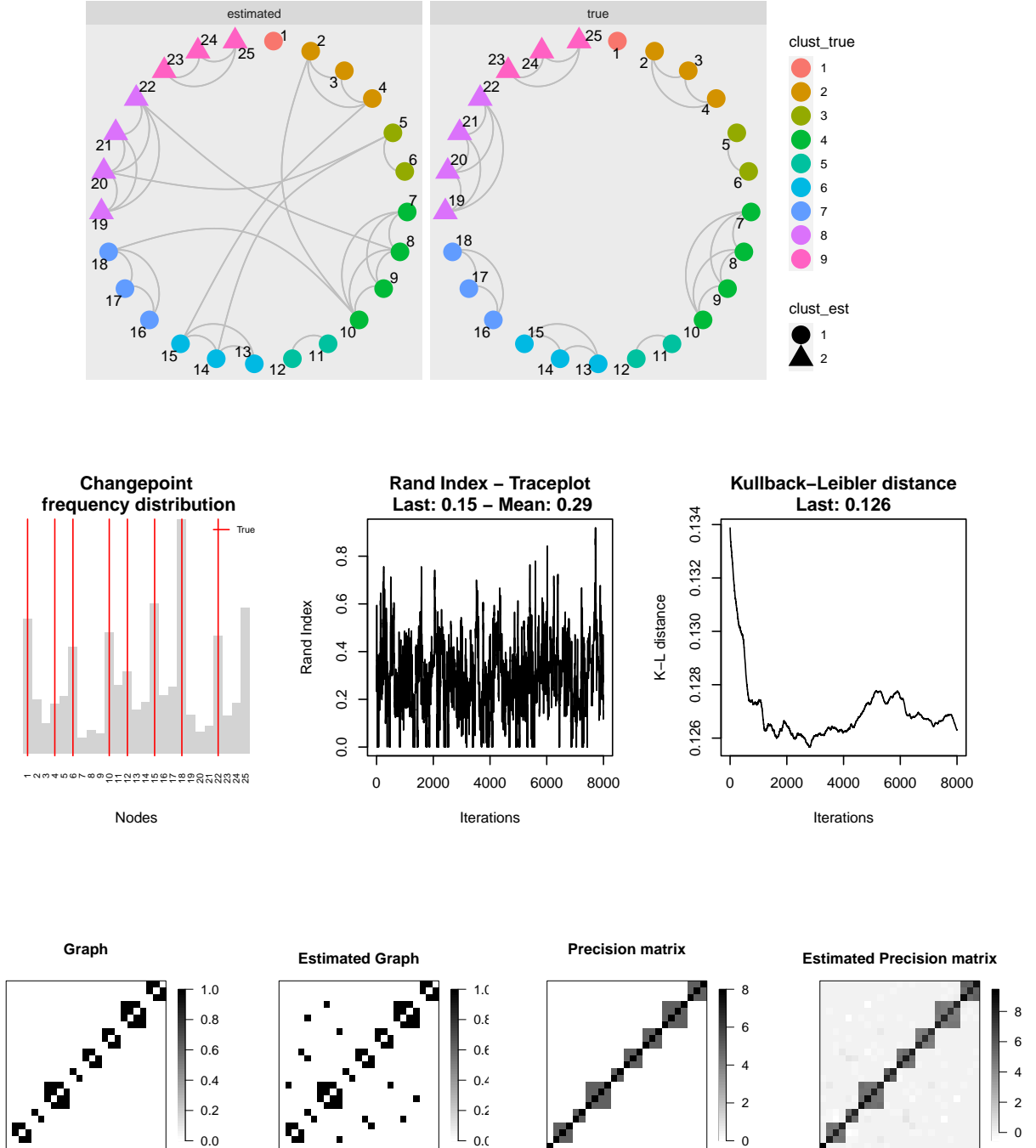
## 5.3 Group 3: varying initial partition

Starting from different initial partitions (i.e. one single partition or all singletons) provide results consistent with the simulation shown in group 1.

#### 5.4 Group 4: varying generating partition's cluster numerosities

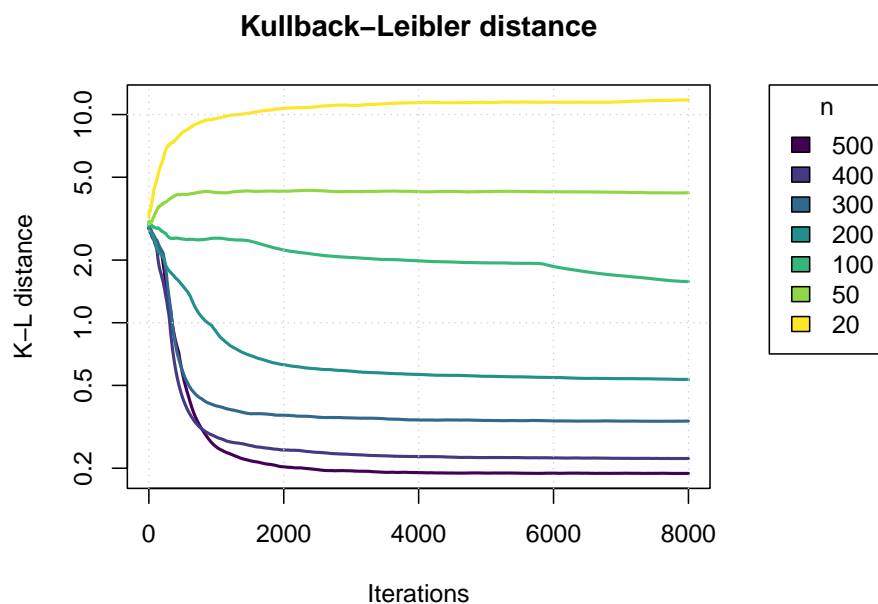
Changing the generating partition's cluster numerosities does not generally influence the result. The only exception comes with small and numerous clusters, as in simulation 22.

The graph is reconstructed accurately. The partition estimate, however, is far from the generating one. It should be noted that this situation is highly unrepresentative of real data that would be meaningful to be studied with this kind of model. Moreover, most of the clusters are indeed recognized as shown in the barplot below: the problem lies in the criteria for selecting the clusters.



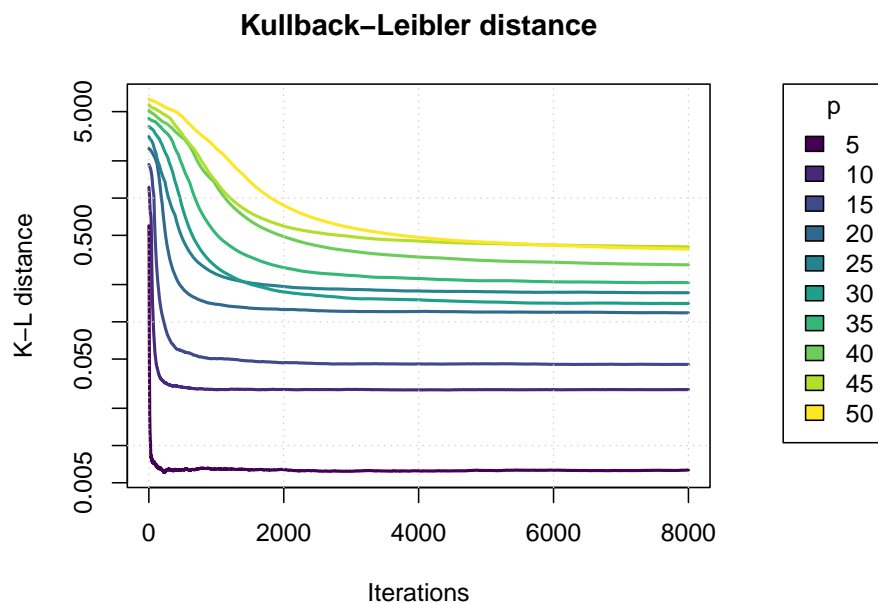
### 5.5 Group 5: varying number of observations

As  $n$  diminishes and becomes comparable to  $p$ , the KL distance plateaus on progressively higher values, and VI and RI behave accordingly. The acceptance rates rises in *more difficult* situations.



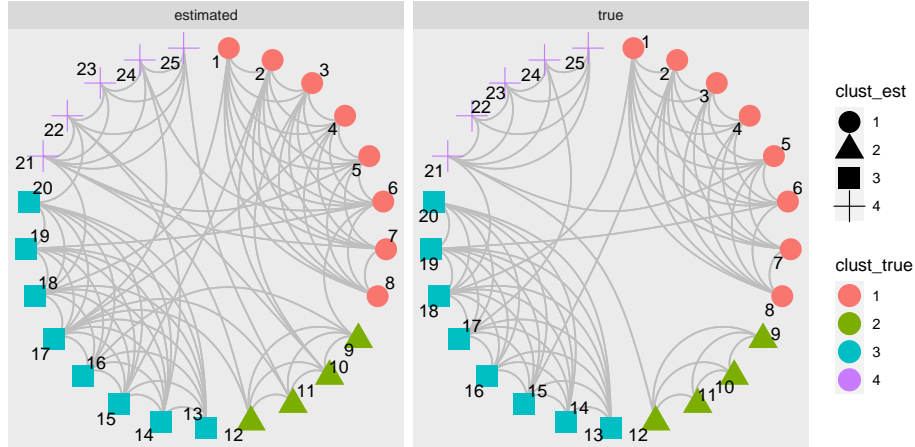
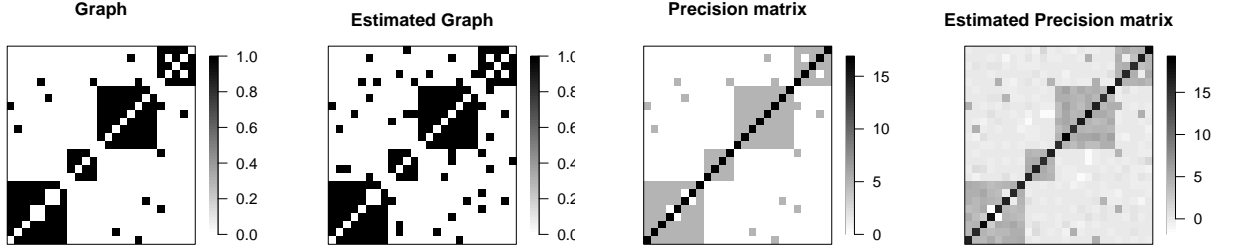
### 5.6 Group 6: varying number of nodes

Increasing  $p$  yields consequences comparable to diminishing  $n$ .



## 5.7 Group 7: using noised block structure as data generator

The data generator for this simulation is a noised block structure, namely with a small probability of having edges between groups and that some edges inside groups are missing. The results are good, since with few exceptions the correct block structure is retrieved.



## 6 Conclusions and further developments

In this work, we have introduced a flexible prior to infer a block-structured graph under an ordering constraint on the nodes, borrowing concepts from the theory of changepoint models and using an adaptive approach. To carry out posterior inference, we have fully implemented a Block Gibbs sampling strategy which allowed us to simultaneously perform structural learning of the graph and clustering of the nodes. From the numerical simulations ran to test and evaluate the model, we obtained that the graph selection based on the BFDR led to a precise identification of the underlying graph structure and the variable clustering, as confirmed by the values of the indexes used to assess the model performance (*i.e.*, VI, RI, KL).

Due to the nature of the work and the limitation of the project timeline to one semester, several challenges remain open for further investigation. One possible direction for future development is, for example, finer hyperparameter tuning. Indeed, for several aspects of the model (*e.g.*, the hyperparameter of the prior for  $\theta$  and  $\sigma$  in the partition prior), we relied on hyperparameters validated in the literature. Even if the selected ones yielded results coherent with the nature of the problem, more structured approaches such as hyperpriors or Bayesian optimization could be introduced to tackle the problem. Furthermore, another crucial step will entail evaluating the model on existing data. Finally, it should be worth investigating two facts related to speed performance: we noticed that overall the mean acceptance rates was under the target one, we suspect that it might be related to a fast convergence in the first 2000 iterations discarded as burnin. The other

aspect is related to the computational complexity of the execution, which we didn't study in detail, yet using the data from the simulations we were able to guess that it seems to scale with  $p^2$ , a linear regression can be found in B.

## A Mathematical details

### A.1 Graph ratio split and merge

After simplifying common factors, the resulting expression in the split case is

$$\begin{aligned} \frac{P(\mathbf{G} \mid \mathbf{z}')}{P(\mathbf{G} \mid \mathbf{z})} &= \left( \frac{1}{B(\alpha, \beta)} \right)^{M+1} \\ &\times \frac{\prod_{l=1}^{S-1} f_B(C'_l, C'_S) f_B(C'_l, C'_{S+1})}{\prod_{l=1}^{S-1} f_B(C_l, C_S)} && \text{interactions with terms before} \\ &\times \frac{\prod_{m=S+2}^{M+1} f_B(C'_S, C'_m) f_B(C'_{S+1}, C'_m)}{\prod_{m=S+1}^M f_B(C_S, C_m)} && \text{interactions with terms after} \\ &\times \frac{f_B(C'_S, C'_{S+1}) f_B(C'_S, C'_S) f_B(C'_{S+1}, C'_{S+1})}{f_B(C_S, C_S)} && \text{internal interactions} \end{aligned}$$

where with  $C_j$  ( $C'_j$ ) we denote group  $j$  in the current (proposed) partition and

$$f_B(C_u, C_v) = B(\alpha + S_{uv}, \beta + S_{uv}^*)$$

Moreover,  $S$  is the index of the cluster that is being split into two, thus

$$\begin{cases} C_m = C'_m & \forall m < S \\ C_S = C'_S \cup C'_{S+1} \\ C_{m-1} = C'_m & \forall m > S + 1 \end{cases}$$

current	$C_1$	$\dots$	$C_S$	$\dots$	$C_M$
proposed	$C'_1$	$\dots$	$C'_S$	$C'_{S+1}$	$C'_{M+1}$

### A.2 Graph ratio shuffle

After simplifying common factors, the resulting expression in the split case is

$$\begin{aligned} \frac{P(\mathbf{G} \mid \mathbf{z}')}{P(\mathbf{G} \mid \mathbf{z})} &= \frac{\prod_{l=1}^{S-1} f_B(C'_l, C'_S) f_B(C'_l, C'_{S+1})}{\prod_{l=1}^{S-1} f_B(C_l, C_S) f_B(C_l, C_{S+1})} && \text{interactions with terms before} \\ &\times \frac{\prod_{l=S+2}^M f_B(C'_l, C'_S) f_B(C'_l, C'_{S+1})}{\prod_{l=S+2}^M f_B(C_l, C_S) f_B(C_l, C_{S+1})} && \text{interactions with terms after} \\ &\times \frac{f_B(C'_S, C'_{S+1}) f_B(C'_S, C'_S) f_B(C'_{S+1}, C'_{S+1})}{f_B(C_S, C_{S+1}) f_B(C_S, C_S) f_B(C_{S+1}, C_{S+1})} && \text{internal interactions} \end{aligned}$$

where with  $C_j$  ( $C'_j$ ) we denote group  $j$  in the current (proposed) partition and

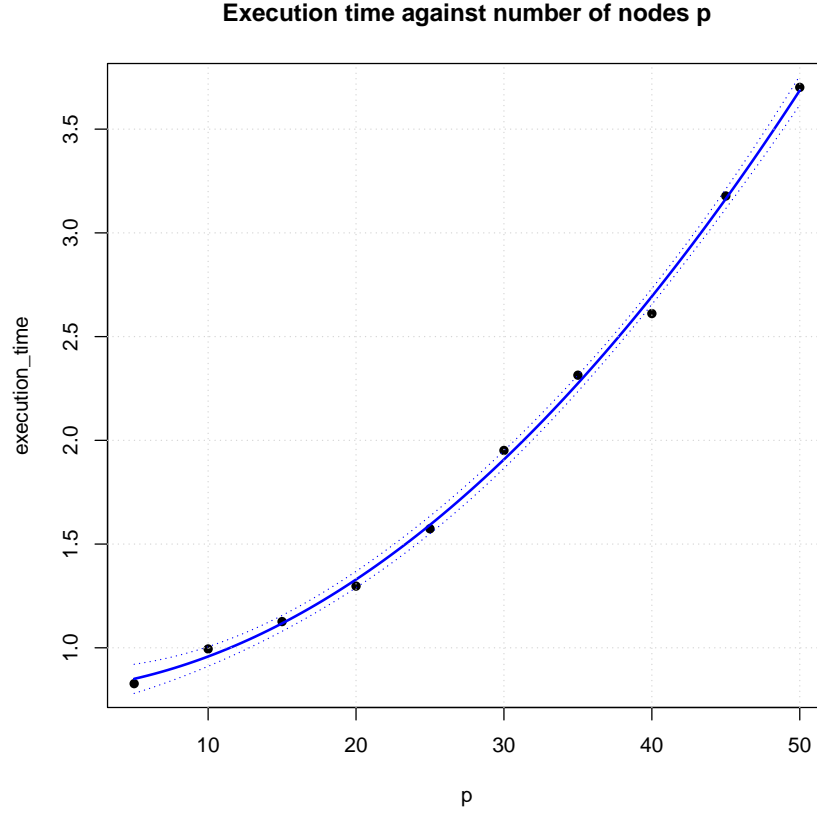
$$f_B(C_u, C_v) = B(\alpha + S_{uv}, \beta + S_{uv}^*)$$

Moreover,  $S$  is the index of the cluster that is being shuffled with the  $(S + 1)$ -th, thus

$$\begin{cases} C_m = C'_m & \forall m < S \\ C_S \cup C_{S+1} = C'_S \cup C'_{S+1} \\ C_m = C'_m & \forall m > S + 1 \end{cases}$$

current	$C_1$	$\dots$	$C_S$	$C_{S+1}$	$\dots$	$C_M$
proposed	$C'_1$	$\dots$	$C'_S$	$C'_{S+1}$	$\dots$	$C'_M$

## B Computational complexity



## C Glossary

**EPPF** Exchangeable Partition Probability Function. 4

**GGM** Gaussian Graphical Models. 2

**KL** Kullback-Leibler. 8, 9, 10, 13, 14

**RI** Rand index. 8, 9, 13

**SBM** Stochastic Block Model. 3, 4

**VI** Variation of Information. 8, 13, 14



## References

- Benson, A. and N. Friel (2018). ‘Adaptive MCMC for Multiple Changepoint Analysis with Applications to Large Datasets’. In: *Electronic Journal of Statistics* 12.2. ISSN: 1935-7524. DOI: 10.1214/18-EJS1418.
- Martínez, A. F. and R. H. Mena (2014). ‘On a Nonparametric Change Point Detection Model in Markovian Regimes’. In: *Bayesian Analysis* 9.4. ISSN: 1936-0975. DOI: 10.1214/14-BA878.
- Mohammadi, A. and E. C. Wit (2015). ‘Bayesian Structure Learning in Sparse Gaussian Graphical Models’. In: *Bayesian Analysis* 10.1. ISSN: 1936-0975. DOI: 10.1214/14-BA889.
- Barbieri, M. M. and J. O. Berger (2004). ‘Optimal predictive model selection’. In: *The Annals of Statistics* 32.3.
- Müller, P., G. Parmigiani and K. M. Rice (2007). ‘FDR and Bayesian Multiple Comparisons Rules’. In: Meilă, M. (2007). ‘Comparing clusterings—an information based distance’. In: *Journal of Multivariate Analysis* 98.5, pp. 873–895.
- Atay-Kayis, A. and H. Massam (2005). ‘A Monte Carlo Method for Computing the Marginal Likelihood in Nondecomposable Gaussian Graphical Models’. In: *Biometrika* 92.2, pp. 317–335. ISSN: 1464-3510, 0006-3444. DOI: 10.1093/biomet/92.2.317.
- Boom, W. van den, M. De Iorio and A. Beskos (2022). ‘Bayesian Learning of Graph Substructures’. In: *Bayesian Analysis* -1 (-1). ISSN: 1936-0975. DOI: 10.1214/22-BA1338. arXiv: 2203.11664 [stat].
- Boom, W. van den, A. Beskos and M. De Iorio (2022). ‘The G-Wishart Weighted Proposal Algorithm: Efficient Posterior Computation for Gaussian Graphical Models’. In: *Journal of Computational and Graphical Statistics*, pp. 1–10. ISSN: 1061-8600, 1537-2715. DOI: 10.1080/10618600.2022.2050250. arXiv: 2108.01308 [stat].
- Colombi, A., R. Argiento, L. Paci and A. Pini (2022). ‘Learning block structured graphs in Gaussian graphical models’. In: *arXiv preprint arXiv:2206.14274*.
- Geng, J., A. Bhattacharya and D. Pati (2018). *Probabilistic Community Detection with Unknown Number of Communities*. arXiv: 1602.08062 [math, stat].
- Legramanti, S., T. Rigon, D. Durante and D. B. Dunson (2022). ‘Extended stochastic block models with application to criminal networks’. In: *The Annals of Applied Statistics* 16.4, pp. 2369–2395.
- Schmidt, M. N. and M. Mørup (2013). ‘Non-Parametric Bayesian Modeling of Complex Networks’. In: *IEEE Signal Processing Magazine* 30.3, pp. 110–128. ISSN: 1053-5888. DOI: 10.1109/MSP.2012.2235191. arXiv: 1312.5889 [stat].
- Wang, H. and S. Z. Li (2012). ‘Efficient Gaussian Graphical Model Determination under G-Wishart Prior Distributions’. In: *Electronic Journal of Statistics* 6 (none). ISSN: 1935-7524. DOI: 10.1214/12-EJS669.

