SCALABLE MULTIPLE NETWORK INFERENCE WITH THE JOINT GRAPHICAL HORSESHOE

Camilla Lingjærde
MRC Biostatistics Unit
University of Cambridge
camilla.lingjaerde@mrc-bsu.cam.ac.uk

Benjamin P. Fairfax
MRC Weatherall Institute for Molecular Medicine
University of Oxford
benjamin.fairfax@oncology.ox.ac.uk

Sylvia Richardson
MRC Biostatistics Unit
University of Cambridge
sylvia.richardson@mrc-bsu.cam.ac.uk

Hélène Ruffieux
MRC Biostatistics Unit
University of Cambridge
helene.ruffieux@mrc-bsu.cam.ac.uk

June 24, 2022

ABSTRACT

Network models are useful tools for modelling complex associations. If a Gaussian graphical model is assumed, conditional independence is determined by the non-zero entries of the inverse covariance (precision) matrix of the data. The Bayesian graphical horseshoe estimator provides a robust and flexible framework for precision matrix inference, as it introduces local, edge-specific parameters which prevent over-shrinkage of non-zero off-diagonal elements. However, for many applications such as statistical omics, the current implementation based on Gibbs sampling becomes computationally inefficient or even unfeasible in high dimensions. Moreover, the graphical horseshoe has only been formulated for a single network, whereas interest has grown in the network analysis of multiple data sets that might share common structures. We propose (i) a scalable expectation conditional maximisation (ECM) algorithm for obtaining the posterior mode of the precision matrix in the graphical horseshoe, and (ii) a novel joint graphical horseshoe estimator, which borrows information across multiple related networks to improve estimation. We show, on both simulated and real omics data, that our single-network ECM approach is more scalable than the existing graphical horseshoe Gibbs implementation, while achieving the same level of accuracy. We also show that our joint-network proposal successfully leverages shared edge-specific information between networks while still retaining differences, outperforming state-of-the-art methods at any level of network similarity.

Keywords Bayesian graphical models · Cancer genomics · Expectation conditional maximisation · Gene networks · Genomics · Graphical horseshoe · Horseshoe prior · High-dimensional inference · Integrative analysis · Multi omics · Network models

1 Introduction

In statistical omics, network models are increasingly popular for representing complex associations and assessing pathway activity. With such models, the links between genes, proteins or other types of omics data can be represented and studied, providing valuable insight into functional relationships. The progress of high-throughput genomic technologies has led to the collection of large, genome-wide data sets, and the availability of biomeasurements of different types has enabled the development of integrative modelling approaches which can increase statistical power while providing detailed insight into complex biological mechanisms (Someren et al. [2002], Karczewski and Snyder [2018]).

If a Gaussian graphical model is assumed, an association (conditional independence) network can be estimated by determining the non-zero entries of the inverse covariance (precision) matrix of the data. For this purpose, Li et al. [2019a] propose *the graphical horseshoe estimator*. This Bayesian model formulation leads to many desirable properties, particularly for identifying weak edges. Indeed, its local edge-specific parameters prevents over-shrinkage of non-zero off-diagonal elements, resulting in a highly flexible framework. However, in the high-dimensional settings commonly needed for investigating biological networks, Gibbs sampling for the graphical horseshoe becomes computationally inefficient or even unfeasible. Moreover, the graphical horseshoe has only been formulated for a single network, whereas interest has grown in the network analysis of multiple data sets that might share common structures. Such related data sets could be different tissues, conditions or patient subgroups, or different omics types, such as gene levels and the protein levels encoded by these genes. A joint approach that utilises the common information while preserving the differences will both have increased statistical power and provide insight into the different mechanisms in play.

In this paper, we propose a computationally efficient expectation conditional maximisation (ECM) algorithm for obtaining the posterior mode of the precision matrix in the graphical horseshoe which allows us to tackle realistic biological inference. Building on this efficient implementation, we then propose a novel joint model that permits borrowing information between multiple networks using the graphical horseshoe prior. We provide the two R packages `fastGHS` and `jointGHS`, which implement the single and joint methods respectively.

The paper is organised as follows. In Section 2, we provide a data-driven motivation for our work, by illustrating the need for a flexible and scalable approach on a multi-condition gene regulation problem in monocytes. In Section 3, we introduce the standard graphical horseshoe and discuss its challenges. In Section 4, we describe our algorithm and scalable inference procedure, and in Section 5, we present our joint graphical horseshoe model formulation which allows for simultaneous inference of multiple networks with shared information. In Section 6, we demonstrate the performance of our proposed methodology on simulated data, and in Section 7, we apply it to the monocyte gene regulation study. Finally, we highlight possible extensions in Section 8.

2 Data and motivating example

We start by introducing our motivating data set, which illustrates the need for a scalable joint graphical framework. The data is from an expression quantitative trait locus (eQTL) study in CD14+ monocytes. The genetic variants are single nucleotide polymorphisms (SNPs) determined using Illumina arrays and monocyte expression was quantified before and after immune stimulation via exposition to inflammation proxies, namely, interferon-γ (IFN-γ) or differing durations of lipopolysaccharide (LPS 2h or LPS 24h) (Fairfax et al. [2014]). The data were obtained from 432 healthy European individuals, more precisely, the number of samples available in each condition was $n_{\text{unstim}} = 413$, $n_{\text{IFN-}\gamma} = 366$, $n_{\text{LPS2h}} = 260$, $n_{\text{LPS24h}} = 321$ for unstimulated cells, and IFN-γ-, LPS 2h- and LPS 24h- stimulated cells, respectively. Monocytes are key components of the innate immune system, therefore studying gene regulation in monocytic conditions is used to clarify the genetic basis of diverse immune-mediated diseases.

Previous studies (Fairfax et al. [2014], Lee et al. [2014], Kim et al. [2014]) suggest that gene stimulation triggers *trans*-regulatory activity, i.e., genetic regulation of distal genes, leading to a beneficial environment for hotspot genetic variants to establish. Hotspots are SNPs regulating the levels of large numbers of gene products, thereby potentially representing important contributors to disease pathogenesis (Yao et al. [2017]).

While Fairfax et al. [2014] mainly report condition-specific *trans*-regulatory activities, they also observe effects across all conditions. The largest hotspot identified by Ruffieux et al. [2020] is persistent across all four conditions. Specifically, using their global-local hotspot modelling approach ATLASQTL, the SNP rs6581889 was found to be the top hotspot in the IFN-γ, unstimulated and LPS 2h studies (associated with 333, 242 and 96 transcripts, respectively), and it was the second largest hotspot in the LPS 24h study (associated with 18 transcripts). This hotspot is located on chromosome 12, only a few Kb away from two genes which it controls (*cis* action), namely, *LYZ* and *YEATS4*, which are thought to play a central role in the pathogenesis of immune disorders (Fairfax et al. [2012]).

To investigate further the possible role of this hotspot in disease risk and progression, and its mechanisms of action, we analyse the network formed by the 381 genes associated with rs6581889 in at least one of the conditions (permutation-based FDR < 0.05). Such an analysis helps characterising the complex interplays among the genes controlled, i.e., the direct effects on the distal *trans* genes or the indirect effects, mediated via other genes controlled by the hotspot (typically via proximal *cis* genes, such as *LYZ* and *YEATS4*). Although graphical modelling approaches seem particularly appropriate to disentangle direct and mediated effects, they haven't been employed thus far.

With its desirable theoretical properties as well as its high performance in numerical studies, the graphical horseshoe estimator would be a natural choice for graph inference in each of the four conditions. Unfortunately, the graphical horseshoe Gibbs sampler of Li et al. [2019a] doesn't scale to the problem dimensions ($p = 381$ genes and relatively few

observations in each condition). Further, the finding of Ruffieux et al. [2020] about the top hotspot being persistent across all four conditions makes a joint approach, which so far has not been employed on the data, highly appropriate. Such an analysis could provide insight into what co-regulatory activity is common or differs between the conditions. Although also relevant for identifying common structures across the conditions, the Bayesian spike-and-slab joint graphical lasso (Li et al. [2019b]) and the joint graphical lasso (Danaher et al. [2014]) are also infeasible or too computationally demanding for inference using the monocyte data set. Indeed, with a joint approach, a total of $289\,560$ links need to be inferred — this extreme high dimensionality makes it clear that a highly scalable and computationally efficient method is needed.

Hereafter, we aim to enable effective inference for our monocyte regulation problem — as well as for similar questions which are increasingly encountered in practice — by proposing an expectation conditional maximisation (ECM) joint graphical horseshoe approach that (i) borrows strength across shared patterns *while* highlighting differences across networks; (ii) is highly scalable, permitting joint inference with realistic genomic problem sizes.

3 Problem statement

Consider a network model where each node is associated with some measurable attribute. Observed values of the multivariate random vector $\boldsymbol{x} = (X_1, \ldots, X_p)^T$ of node attributes, each entry corresponding to one of p variables, can then be used to infer a graph under suitable model assumptions. Given multivariate Gaussian node attributes, with $n \times p$ observation matrix $\boldsymbol{X}$ with i.i.d. rows $\boldsymbol{x_1}, \ldots, \boldsymbol{x_n} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$, we can infer a *partial correlation network* by estimating the inverse covariance matrix, or precision matrix, $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$. The partial correlation between nodes i and j conditioned upon all others is then given by

$$\rho_{ij|V\setminus\{i,j\}} = -\frac{\theta_{ij}}{\sqrt{\theta_{ii}\theta_{jj}}},$$

where the θ_{ij}'s are the entries of $\boldsymbol{\Theta}$ and V is the set of all node pairs (Kolaczyk [2009]). For Gaussian variables, correlation equal to zero is equivalent to independence, and so a conditional independence graph can be constructed by determining the non-zero entries of the precision matrix. The graph is assumed to be *sparse*, meaning that the precision matrix has mostly zero elements. The *sparsity* is a measurement giving the number of edges in the edge set E relative to the number of potential edges in the graph, $2|E|/(p^2 - p)$, where $|\cdot|$ denotes the size of the set. The precision matrix must also necessarily be positive definite $\boldsymbol{\Theta} \succ 0$, so that it is invertible.

In a Gaussian graphical model setting, network estimation requires a sparse estimate for the posterior mode of the precision matrix $\boldsymbol{\Theta}$. In a high-dimensional setting, achieving both sparsity and positive definiteness requires regularisation methods, and there is significant literature on this problem, both frequentist and Bayesian. Notable frequentist methods include the graphical lasso (Friedman et al. [2008]), the neighbourhood selection of Meinshausen and Bühlmann [2006] and the graphical SCAD (Fan et al. [2009]). In later years, Bayesian methods such as the Bayesian graphical lasso (Wang et al. [2012]), the spike-and-slab prior for precision matrices (Wang [2015]) and the graphical horseshoe model of Li et al. [2019a] have gained popularity. The latter adapts the horseshoe prior of Carvalho et al. [2010] to a graphical setting.

The graphical horseshoe model puts horseshoe priors on the off-diagonal elements of the precision matrix, encouraging sparse solutions. An uninformative prior is put on the diagonal elements, and the positive definiteness constraint is respected. Due to symmetry, it is sufficient to consider the upper off-diagonal elements of $\boldsymbol{\Theta}$. Normal scale mixtures with half-Cauchy hyperpriors are used on the off-diagonal elements. The hierarchy of the model is as follows:

$$\begin{aligned} \theta_{ii} &\propto 1, \\ \theta_{ij|i<j} &\sim \mathcal{N}(0, \lambda_{ij}^2\tau^2), \\ \lambda_{ij|i<j} &\sim \mathrm{C}^+(0,1), \\ \tau &\sim \mathrm{C}^+(0,1), \end{aligned} \tag{1}$$

where $\mathrm{C}^+(0,1)$ is the half-Cauchy distribution with density $p(x) \propto (1+x^2)^{-1},\ x > 0$, and $1 \leq i, j \leq p$. In (1), the $\lambda_{ij|i<j}$'s are local shrinkage parameters that avoid over-shrinking non-zero element-wise effects while τ is a global shrinkage parameter that ensures sparsity.

To enable practical inference for the posterior distribution of $\boldsymbol{\Theta}$ in networks of realistic sizes, we next develop a faster alternative to the Gibbs sampling procedure proposed by Li et al. [2019a], inspired by the expectation conditional maximisation (ECM) approach introduced by Li and McCormick [2019] for the spike-and-slab prior.

4 An ECM algorithm for estimating the graphical horseshoe

We detail the updates of an ECM algorithm under the graphical horseshoe prior (1). The ECM approach, described first by Meng and Rubin [1993], is a generalised EM algorithm (Dempster et al. [1977]) where a complex maximisation step (M-step) is replaced with several computationally simpler conditional maximisation steps (CM-steps).

4.1 Full conditional posteriors

As in Li et al. [2019a], the full conditional posteriors of the local and global $\lambda_{ij|i<j}$'s and τ can be derived by introducing the augmented variables $\nu_{ij|i<j}$ and ξ. We next employ the following reparameterisation, introducing the latent ν_{ij} and writing

$$\begin{aligned}\lambda_{ij}^2|\nu_{ij} &\sim \text{InvGamma}(1/2, 1/\nu_{ij}),\\ \nu_{ij} &\sim \text{InvGamma}(1/2, 1).\end{aligned}$$

Using a key observation from Makalic and Schmidt [2015], we find the full conditional posteriors as

$$\begin{aligned}\lambda_{ij}^2|\cdot &\sim \text{InvGamma}(1, 1/\nu_{ij} + \theta_{ij}^2/(2\tau^2)),\\ \nu_{ij}|\cdot &\sim \text{InvGamma}(1, 1 + 1/\lambda_{ij}^2),\end{aligned} \tag{2}$$

where $\cdot$ denotes all other variables. The latent variables can be collected in the latent matrix $\boldsymbol{N} = (\nu_{ij})$. Similarly, we let $\tau^2|\xi \sim \text{InvGamma}(1/2, 1/\xi)$ and $\xi \sim \text{InvGamma}(1/2, 1)$, and get inverse Gamma full conditional posteriors for them as well. To obtain conditional posteriors for the precision matrix and the local scale parameters, each column and row of the matrices $\boldsymbol{\Theta}$ and $\boldsymbol{\Lambda} = (\lambda_{ij}^2)$ are partitioned from a $p \times p$ matrix of parameters. Without loss of generality, we describe the updates for the last row and column. As in Wang et al. [2012], they are partitioned out in the matrices

$$\begin{aligned}\boldsymbol{\Theta} &= \begin{pmatrix} \boldsymbol{\Theta}_{(-p)(-p)} & \boldsymbol{\theta}_{(-p)p} \\ \boldsymbol{\theta}_{(-p)p}^T & \theta_{pp} \end{pmatrix},\ \boldsymbol{S} = \begin{pmatrix} \boldsymbol{S}_{(-p)(-p)} & \boldsymbol{s}_{(-p)p} \\ \boldsymbol{s}_{(-p)p}^T & s_{pp} \end{pmatrix},\\ \boldsymbol{\Lambda} &= \begin{pmatrix} \boldsymbol{\Lambda}_{(-p)(-p)} & \boldsymbol{\lambda}_{(-p)p} \\ \boldsymbol{\lambda}_{(-p)p}^T & 1 \end{pmatrix},\end{aligned}$$

where $\boldsymbol{S} = \boldsymbol{X}^T\boldsymbol{X}$ is the scatter matrix of the observed data $\boldsymbol{X}$. The diagonal elements of $\boldsymbol{\Lambda}$ are not of relevance and can be set to an arbitrary value such as 1. The posterior distribution for the last column (and row) of $\boldsymbol{\Theta}$ can be obtained as

$$\begin{aligned}&p(\boldsymbol{\theta}_{(-p)p}, \theta_{pp}|\boldsymbol{\Theta}_{(-p)(-p)}, \boldsymbol{S}, \boldsymbol{\Lambda}, \tau)\\ &\propto (\theta_{pp} - \boldsymbol{\theta}_{(-p)p}^T\boldsymbol{\Theta}_{(-p)(-p)}^{-1}\boldsymbol{\theta}_{(-p)p})^{n/2}\\ &\quad\times \exp\left\{-\boldsymbol{s}_{(-p)p}^T\boldsymbol{\theta}_{(-p)p} - s_{pp}\theta_{pp}/2 - \boldsymbol{\theta}_{(-p)p}^T(\boldsymbol{\Lambda}^*\tau^2)^{-1}\boldsymbol{\theta}_{(-p)p}/2\right\}.\end{aligned}$$

With a variable change, the conditional distributions can be reformulated as

$$\begin{aligned}\boldsymbol{\theta}_{(-p)p}|\boldsymbol{\Theta}_{(-p)(-p)}, \boldsymbol{S}, \boldsymbol{\Lambda}, \tau &\sim \text{Normal}(-C\boldsymbol{s}_{(-p)p}, C),\\ \theta_{pp} - \boldsymbol{\theta}_{(-p)p}^T\boldsymbol{\Theta}_{(-p)(-p)}^{-1}\boldsymbol{\theta}_{(-p)p}|\boldsymbol{\Theta}_{(-p)(-p)}, \boldsymbol{S}, \boldsymbol{\Lambda}, \tau &\sim \text{Gamma}(n/2+1, s_{pp}/2),\end{aligned} \tag{3}$$

where $C = \{s_{pp}\boldsymbol{\Theta}_{(-p)(-p)}^{-1} + (\boldsymbol{\Lambda}^*\tau^2)^{-1}\}^{-1}$ and $\boldsymbol{\Lambda}^* = \text{diag}(\boldsymbol{\lambda}_{(-p)p})$. By iteratively permuting each row and column to be the last, the conditional posterior of all elements of the precision matrix can then be found row and column wise.

4.2 ECM algorithm

Given the estimates from the previous iteration l, the objective function is given by

$$
\begin{aligned}
Q(\boldsymbol{\Theta}, \boldsymbol{\Lambda}, \tau | \boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)}, \tau^{(l)}) =& \mathrm{E}_{\boldsymbol{N}, \xi | \boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)}, \tau^{(l)}, \boldsymbol{S}} \left\{ \log p(\boldsymbol{\Theta}, \boldsymbol{\Lambda}, \tau, \boldsymbol{N}, \xi | \boldsymbol{S}) | \boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)}, \tau^{(l)} \right\} \\
=& \frac{n}{2} \log \left(\det \boldsymbol{\Theta} \right) - \frac{1}{2} \mathrm{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + \sum_{i<j} \Big\{ -4 \log \left(\lambda_{ij} \right) - \frac{\theta_{ij}^2}{2\tau^2 \lambda_{ij}^2} \\
& - 2\mathrm{E}_{\cdot|\cdot} \left\{ \log \left(\nu_{ij} \right) \right\} - \left(\frac{1}{\lambda_{ij}^2} + 1 \right) \mathrm{E}_{\cdot|\cdot} \left(\frac{1}{\nu_{ij}} \right) \Big\} - \left(\frac{p(p-1)}{2} + 3 \right) \log \left(\tau \right) \\
& - 2\mathrm{E}_{\cdot|\cdot} \left\{ \log \left(\xi \right) \right\} - \left(1 + \frac{1}{\tau^2} \right) \mathrm{E}_{\cdot|\cdot} \left(\frac{1}{\xi} \right) + \text{const.},
\end{aligned} \tag{4}
$$

where $\mathrm{E}_{\cdot|\cdot}(\cdot)$ denotes $\mathrm{E}_{\boldsymbol{N}, \xi | \boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)}, \tau^{(l)}, \boldsymbol{S}}(\cdot)$ and const. is a constant not depending on $\boldsymbol{\Theta}$, $\boldsymbol{\Lambda}$ or τ.

In the E-step of the algorithm, the conditional expectations in (4) are computed. After the E-step is computed, the CM-step performs the maximisation with respect to $(\boldsymbol{\Theta}, \boldsymbol{\Lambda}, \tau)$.

4.2.1 The E-step

From (2), we have that the full conditional distributions of the ν_{ij}'s and ξ are inverse Gamma, so

$$
\begin{aligned}
\mathrm{E}_{\cdot|\cdot} \left\{ \log \left(\nu_{ij} \right) \right\} &= \log \left(1 + \frac{1}{\lambda_{ij}^{(l)^2}} \right) - \psi(1), \\
\mathrm{E}_{\cdot|\cdot} \left(\frac{1}{\nu_{ij}} \right) &= \frac{1}{1 + 1/\lambda_{ij}^{(l)^2}} = \frac{\lambda_{ij}^{(l)^2}}{\lambda_{ij}^{(l)^2} + 1} =: \lambda_{ij}^{*^{(l)}}, \\
\mathrm{E}_{\cdot|\cdot} \left\{ \log \left(\xi \right) \right\} &= \log \left(1 + \frac{1}{\tau^{(l)^2}} \right) - \psi(1), \\
\mathrm{E}_{\cdot|\cdot} \left(\frac{1}{\xi} \right) &= \frac{1}{1 + 1/\tau^{(l)^2}} = \frac{\tau^{(l)^2}}{\tau^{(l)^2} + 1} =: \tau^{*^{(l)}},
\end{aligned} \tag{5}
$$

where $\psi(\cdot)$ is the digamma function. Inserting (5) into (4) and disregarding terms that do not involve $\boldsymbol{\Theta}$, $\boldsymbol{\Lambda}$ or τ, the objective function becomes

$$
\begin{aligned}
Q(\boldsymbol{\Theta}, \boldsymbol{\Lambda}, \tau | \boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)}, \tau^{(l)}) =& \frac{n}{2} \log \left(\det \boldsymbol{\Theta} \right) - \frac{1}{2} \mathrm{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + \sum_{i<j} \left\{ -4 \log \left(\lambda_{ij} \right) - \frac{\theta_{ij}^2}{2\tau^2 \lambda_{ij}^2} - \frac{\lambda_{ij}^{*^{(l)}}}{\lambda_{ij}^2} \right\} \\
& - \left(\frac{p(p-1)}{2} + 3 \right) \log \left(\tau \right) - \frac{\tau^{*^{(l)}}}{\tau^2} + \text{const.}
\end{aligned} \tag{6}
$$

4.2.2 The CM-step

The CM-step maximises (6) with respect to $(\boldsymbol{\Theta}, \boldsymbol{\Lambda}, \tau)$ in a coordinate ascent fashion. The following closed-form updates are obtained for τ^2 and λ_{ij}^2

$$
\tau^{(l+1)^2} = \frac{2 \sum_{i<j} \left\{ \frac{\theta_{ij}^2}{\lambda_{ij}^2} \right\} + 4\tau^{*^{(l)}}}{p(p-1) + 6}, \tag{7}
$$

$$
\lambda_{ij}^{(l+1)^2} = \frac{\lambda_{ij}^{*^{(l)}} + \theta_{ij}^2 / (2\tau^2)}{2}. \tag{8}
$$

There is no closed form for the update of the precision matrix, however, (3) gives the updates for the last row and column of $\boldsymbol{\Theta}$:

$$\theta_{pp}^{(l+1)} = \boldsymbol{\theta}_{(-p)p}^{(l+1)^T} \left(\boldsymbol{\Theta}_{(-p)(-p)}^{(l+1)}\right)^{-1} \boldsymbol{\theta}_{(-p)p}^{(l+1)} + \frac{n}{s_{pp}},$$
$$\boldsymbol{\theta}_{(-p)p}^{(l+1)} = -\left\{ s_{pp}\left(\boldsymbol{\Theta}_{(-p)(-p)}^{(l+1)}\right)^{-1} + \frac{1}{\left(\tau^{(l+1)}\right)^2}\left(\boldsymbol{\Lambda}^{*^{(l+1)}}\right)^{-1} \right\}^{-1} \boldsymbol{s}_{(-p)p}, \tag{9}$$

setting $l+1 = l$ at each iteration. By iteratively permuting each row and column to be the last, all elements of the precision matrix can be updated row and column wise. With this CM-step update, it is ensured that $Q(\boldsymbol{\Theta}^{(l+1)}, \boldsymbol{\Lambda}^{(l+1)}, \tau^{(l+1)} | \boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)}, \tau^{(l)}) \geq Q(\boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)}, \tau^{(l)} | \boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)}, \tau^{(l)})$ (Meng and Rubin [1993]). By iterating between the E-step and the CM-step until convergence, we obtain an estimator of the posterior mode $\widehat{\boldsymbol{\Theta}}$. The full derivations for this section are given in the appendix.

One of the main computational advantages of the ECM approach over stochastic search is that the posterior mode is fast to obtain. The estimates are computed directly and a full stochastic search is not necessary. Further, the entries corresponding to unidentified edges tend to converge to values close to zero and the separation with the identified edges increases as the algorithm converges – such an observation has also been reported by others in the context of EM or variational inference (Kook et al. [2021]). We hereafter refer to this ECM implementation as "fastGHS".

4.3 Global shrinkage parameter selection

While we have treated τ as an unknown parameter up until now, it can not be updated in the ECM approach like the other variables as it collapses to zero in very sparse settings (Scott and Berger [2010]), and therefore needs to be fixed. In a non-graphical setting, Van Der Pas et al. [2014] fix the global shrinkage parameter for the horseshoe estimator to be the estimated sparsity level. Bhadra et al. [2017] use a similar empirical Bayes approach for their horseshoe+ estimator, letting the global scale be of the order of the expected proportion of non-null effects. Similarly, based on the approach of Piironen and Vehtari [2017], others in similar non-graphical settings have fixed the global shrinkage parameter by making prior assumptions about the effective model size, derived from the model shrinkage factors (Williams et al. [2018]). This is not transferable to our graphical setting due to the iterative nature of the updates (9).

We propose to select τ^2 using the AIC criterion for Gaussian graphical models (Akaike et al. [1973]). For a given global shrinkage parameter τ^2 and corresponding precision matrix estimate $\widehat{\boldsymbol{\Theta}}_{\tau^2}$, the AIC score is given by

$$\text{AIC}(\tau^2) = \frac{n}{n-1}\text{tr}(\boldsymbol{S}\widehat{\boldsymbol{\Theta}}_{\tau^2}) - n\log\left\{\det(\widehat{\boldsymbol{\Theta}}_{\tau^2})\right\} + 2|E_{\tau^2}|,$$

where tr is the trace, $\boldsymbol{S} = \boldsymbol{X}^T\boldsymbol{X}$ is the scatter matrix and $|E_{\tau^2}|$ is the size of the corresponding edge set.

Figure A.10 in the appendix shows that for small τ^2, small increases lead to large changes in the AIC score. However, for sufficiently large values the AIC score stabilises as the global shrinkage parameter increases. Thus, instead of attempting to identify the globally AIC minimising value of τ^2, which is computationally exhausting, we start with a small value and increase it until the AIC has stabilised. This approach shares similarities with the "dynamic posterior search" of Ročková and George [2018]. Formally, considering τ^2 from a suitable grid of M increasing values $\{\tau_1^2, \ldots, \tau_M^2\}$, we select the τ^2_{AIC} given by

$$\tau^2_{\text{AIC}} = \min\left\{\tau_m^2 : |\text{AIC}(\tau_m^2) - \text{AIC}(\tau_{m-1}^2)| < \epsilon\right\},$$

for some convergence tolerance ϵ.

5 Multiple network inference

In this section, we describe the *joint graphical horseshoe* for multiple network inference. By sharing information through common latent variables, the method gives more precise estimates for networks with any level of similarity. The heavy tails of the horseshoe prior permits effectively capturing network-specific edges, a property that few Bayesian methods developed for similar purposes share. The resulting joint graphical horseshoe estimator *simultaneously* shares information between networks *and* captures their differences. In addition, due to the scalability of the ECM implementation, our method allows for joint network modelling for a larger number of networks than existing Bayesian approaches do.

5.1 Joint graphical horseshoe model formulation

Given K networks with p nodes each, with $n_k \times p$ observation matrices $\boldsymbol{X}_k$ for $k = 1, \ldots, K$, we are interested in the precision matrices $\{\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_K\}$. We let the k^{th} precision matrix follow the hierarchical model

$$\begin{aligned} \theta_{iik} &\propto 1, \\ \theta_{ijk|i<j} &\sim \mathcal{N}(0, \lambda_{ijk}^2 \tau_k^2), \\ \lambda_{ijk|i<j} &\sim \text{C}^+(0, 1), \\ \tau_k &\sim \text{C}^+(0, 1), \end{aligned}$$

for $1 \leq i, j \leq p$. This is the standard graphical horseshoe model for each network separately. To share information across networks, we introduce the latent variables $\nu_{ij|i<j}$ and write

$$\begin{aligned} \lambda_{ijk}^2 | \nu_{ij} &\sim \text{InvGamma}(1/2, 1/\nu_{ij}), \\ \nu_{ij} &\sim \text{InvGamma}(1/2, 1). \end{aligned}$$

We then derive the full conditional posteriors. Because the λ_{ijk}'s of the different data sets are independent given the ν_{ij}'s, we have

$$\lambda_{ijk}^2 | \cdot \sim \text{InvGamma}(1, 1/\nu_{ij} + \theta_{ijk}^2/(2\tau_k^2)),$$

similarly to the standard graphical horseshoe. Hence, information is now shared across networks through the common latent variable ν_{ij}.

The full conditional posterior of the ν_{ij}'s now depends on the λ_{ijk}'s of all K networks:

$$p(\nu_{ij} | \cdot) \propto \text{InvGamma}\left(\frac{K+1}{2}, 1 + \sum_{k=1}^{K} \frac{1}{\lambda_{ijk}^2}\right). \tag{10}$$

We introduce the latent variables ξ_k for the τ_k^2's with a similar parameterisation $\tau_k^2 | \xi_k \sim \text{InvGamma}(1/2, 1/\xi_k)$ and $\xi_k \sim \text{InvGamma}(1/2, 1)$, and get inverse Gamma full conditional posteriors for them as well. The global scales are kept network-specific to allow for different sparsity levels across networks. The derivation of (10) is given in the appendix.

5.2 ECM approach

With a multiple network approach, the E-step and CM-step in the ECM algorithm are very similar to the single network version. Since the networks are independent given the common latent variables ν_{ij}, we can perform the maximisation of the λ_{ijk}'s and the θ_{ijk}'s for $k = 1, \ldots, K$ separately. The main difference is that the expectation of ν_{ij} now depends on all K networks.

5.2.1 E-step

In the E-step, the ξ_k's follow the same distribution as in the standard graphical horseshoe within each network. This means that

$$\begin{aligned} \text{E}_{\cdot|\cdot}\left\{\log\left(\xi_k\right)\right\} &= \log\left(1 + \frac{1}{{\tau_k^{(l)}}^2}\right) - \psi(1), \\ \text{E}_{\cdot|\cdot}\left(\frac{1}{\xi_k}\right) &= \frac{1}{1 + 1/{\tau_k^{(l)}}^2} = \frac{{\tau_k^{(l)}}^2}{{\tau_k^{(l)}}^2 + 1} =: {\tau_k^*}^{(l)}. \end{aligned}$$

The distribution of the ν_{ij}'s now depends on the local shrinkage parameters of all K networks. Using the full conditional distribution (10), we get

$$\mathrm{E}_{\cdot|\cdot}\left\{\log\left(\nu_{ij}\right)\right\} = \log\left(1+\sum_{k=1}^{K}\frac{1}{\lambda_{ijk}^{(l)^2}}\right) - \psi\left(\frac{K+1}{2}\right),$$
$$\mathrm{E}_{\cdot|\cdot}\left(\frac{1}{\nu_{ij}}\right) = \frac{K}{2\left(1+\sum_{k=1}^{K}1/\lambda_{ijk}^{(l)^2}\right)} =: \lambda_{ij\cdot}^{*}(l). \tag{11}$$

5.2.2 CM-step

In the CM-step for multiple networks, the global shrinkage parameter for each graph k is updated as the maximising value (7) within the network:

$$\tau_k^{(l+1)^2} = \frac{2\sum_{i<j}\left\{\frac{\theta_{ijk}^2}{\lambda_{ijk}^2}\right\} + 4\tau_k^{*^{(l)}}}{p(p-1)+6}.$$

We obtain the updates for the λ_{ijk}^2's in the multiple network setting by replacing the expectation $\mathrm{E}_{\cdot|\cdot}\left(\frac{1}{\nu_{ij}}\right) = \lambda_{ij}^{*}(l)$ in (8) by $\lambda_{ij\cdot}^{*}(l)$ found in the E-step in (11):

$$\lambda_{ijk}^{(l+1)^2} = \frac{\lambda_{ij\cdot}^{*^{(l)}} + \theta_{ijk}^2/(2\tau_k^2)}{2}. \tag{12}$$

The precision matrices $\boldsymbol{\Theta}_k$ are also updated separately for each network, as they are independent given the ν_{ij}'s. Setting $l+1=l$ at each iteration, we get the ordinary graphical horseshoe block updates (Li et al. [2019a]) given by

$$\theta_{ppk}^{(l+1)} = \boldsymbol{\theta}_{(-p)pk}^{(l+1)^T}\left(\boldsymbol{\Theta}_{(-p)(-p)k}^{(l+1)}\right)^{-1}\boldsymbol{\theta}_{(-p)pk}^{(l+1)} + \frac{n_k}{s_{ppk}},$$
$$\boldsymbol{\theta}_{(-p)pk}^{(l+1)} = -\left(s_{ppk}\left(\boldsymbol{\Theta}_{(-p)(-p)k}^{(l+1)}\right)^{-1} + \frac{1}{\left(\tau_k^{(l+1)}\right)^2}\left(\boldsymbol{\Lambda}_k^{*^{(l+1)}}\right)^{-1}\right)^{-1}\boldsymbol{s}_{(-p)pk},$$

where the matrix partitioning is analogous to (9). We iterate between the E-step and the CM-step until convergence is achieved for all K graphs, when the updates for all precision matrix elements of all K graphs differ from their previous estimate in absolute value by less than some tolerance threshold. We hereafter refer to this ECM implementation of our joint graphical network model as "jointGHS".

5.3 Global shrinkage parameter selection

As in the single network setting, we need to fix the global shrinkage parameters τ_k in the K networks. We do this for each network separately, using the approach described in Section 4.3. This way, we allow for different sparsity levels across networks, avoiding the over- or under-selection of edges that might occur for some networks should we assume the same sparsity level for all. After each τ_k has been selected, we run the joint method with these values. This means that the single-network approach is ran on each network separately before the joint analysis is performed. In our implementation, we do this in parallel for computational efficiency.

5.4 More on the heavy horseshoe tail

In the joint graphical horseshoe, information is shared through the common latent parameter ν_{ij}. The larger the full conditional expectation of $1/\nu_{ij}$ in (11) is, the larger are the CM-updates (12) for the local scales λ_{ijk}, $k=1,\ldots,K$, and hence the larger the updates for the corresponding precision matrix elements. Thus, a large posterior expected value of $1/\nu_{ij}$ signifies strong evidence for the edge (i,j) being present in all networks. It is also clear from the CM-updates (12) that even if the conditional expectation of $1/\nu_{ij}$ is close to zero, this does not imply that the updates for all the local scales will be close to zero. That is, thanks to the heavy tail of the half-Cauchy distribution, if there is enough evidence from the data, an edge can be identified in an individual network even though the common latent parameter signifies no edge. This is illustrated by Figure 1, which shows partial correlations estimated by the joint graphical horseshoe for

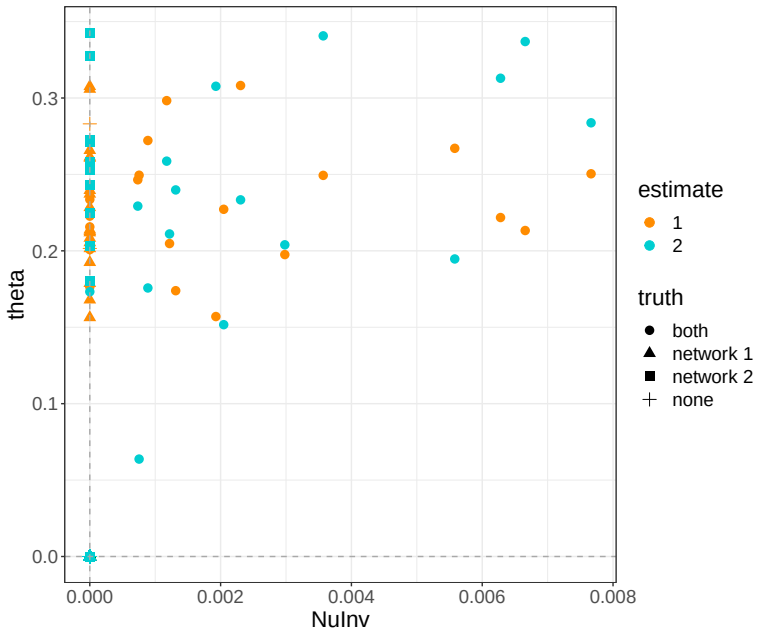

Figure 1: The size of the partial correlations estimated by the joint graphical horseshoe, plotted against the corresponding inverse shared latent parameters for local scales. The data sets correspond to $K = 2$ networks with 40% edge agreement, with $p = 100$ nodes, and $n_1 = 100$ and $n_2 = 150$ observations respectively. Both networks have a true sparsity of 0.02. The points are colored according to the network for which they are estimated, and the shape indicates whether the corresponding edge truly is present in both, one or neither networks.

$K = 2$ networks, plotted against the posterior expectations of the corresponding shared latent parameters $1/\nu_{ij}$. The true networks have 99 edges each, of which they have 39 in common. The details of the data generation and analysis are given the the appendix. Figure 1 shows that the expectation of $1/\nu_{ij}$ is only far from zero when an edge is present in both networks. When this expectation is close to zero, i.e., shared information is not found, posterior output still captures edges (i.e. non-zero θ_{ijk}) specific to each network. This illustrates how the joint graphical horseshoe estimator can simultaneously share information between networks and capture their differences.

5.5 Posterior checks with the Bayesian bootstrap

The joint graphical horseshoe adapts well to the level of similarity between networks, but it is not meant to be applied to a set with many highly similar networks and a few unrelated or less similar networks. This limitation generally applies to joint graphical methods, including the spike-and-slab joint graphical lasso (Li et al. [2019b]) and joint graphical lasso (Danaher et al. [2014]). In such a case, the highly similar networks dominate the inference of the common latent variables ν_{ij} in the joint graphical horseshoe. Due to the heavy tail of the horseshoe, when $1/\nu_{ij}$ is small, the local scales can adapt and identify edges individually on the network-level. However, a large $1/\nu_{ij}$ tends to lead to non-zero precision matrix elements for all networks.

To account for this possibility, we propose to carry out a posterior check to evaluate whether the joint network estimates are in strong contradiction with the single network estimates, suggesting that a joint analysis should be reconsidered. This routine uses the Bayesian bootstrap (Rubin [1981]) and is described in the appendix; it is also implemented in our R package `jointGHS`.

6 Simulations

To evaluate the performance of our approach, we have performed comprehensive simulation studies in R (R Core Team [2013]). The details of the simulation study are given in the appendix, and the corresponding code is available on Github (`https://github.com/Camiling/jointGHS_simulations`). We have generated data as close as possible to our multiomic application of interest, with non-zero partial correlations between 0.1 and 0.2 in magnitude and with

the *scale-free property* (i.e. the degree distribution follows a power-law distribution), a known trait in multiomic data (Kolaczyk [2009]). We assess graph accuracy by the *precision*, which is the fraction of the inferred edges that are actually present in the true graph (also known as positive predictive value), and the *recall*, which is the fraction of the edges in the true graph that are present in the inferred one (also known as sensitivity). Because the inferred networks of the different methods do not necessarily have the same sparsity, there is some consideration needed when comparing their precision and recall. For example, the recall tends to increase when the number of edges increases, favouring methods that over-select edges. In the discussion of the results, we therefore put more emphasis on the precision since we consider false positives to be a larger concern than false negatives: in an omics application, we would rather identify few but highly reliable associations than many but less accurate ones. We consider the recall to be an informative additional measure, particularly in situations when two methods have comparable precision.

Our numerical experiments are divided into three parts. In Subsection 6.1, we assess the statistical and computational performance of fastGHS in a single network setting, comparing it to the Gibbs sampling version of Li et al. [2019a] and to the graphical lasso of Friedman et al. [2008]. In Subsection 6.2, we assess the edge-selection performance of the joint graphical horseshoe in a multiple network setting, comparing it to the Bayesian spike-and-slab joint graphical lasso of Li et al. [2019b] and to the joint graphical lasso of Danaher et al. [2014]. Finally, in Subsection 6.3 we demonstrate the benefits of joint modelling, showing how the accuracy of the joint graphical horseshoe increases with the number of related networks.

6.1 Comparison to the Gibbs sampling scheme for single networks

In this subsection we assess how the performance of our ECM implementation of the graphical horseshoe compares to the Gibbs sampler by Li et al. [2019a]. To serve as baseline reference we also provide the results of the widely used graphical lasso lasso algorithm for Gaussian graphical model inference (Friedman et al. [2008]). We consider settings with different numbers of nodes $p \in \{50, 100\}$ and observations $n \in \{100, 200\}$. For each setting, we construct a $p \times p$ precision matrix and sample $N = 20$ data sets with n observations from the corresponding multivariate Gaussian distribution. For each generated data set, we use the different graph reconstruction methods to obtain precision matrix estimates.

In a graphical setting, $n = 200$ and $p = 100$ gives rise to a high-dimensional problem since there are $(p^2 - p)/2 = 4950$ potential edges. For the Gibbs sampling implementation of the graphical horseshoe, a larger p, such as 200, leads to computational problems as the algorithm entails singular updates, likely a result overflow not being properly dealt with. This holds for both the original MATLAB implementation and our translation into R, where the algorithm halts as it attempts to solve a singular system. In this comparison we therefore only consider up to $p = 100$ nodes.

Table 1 indicates the performance of fastGHS is comparable to the Gibbs sampler, and tends to have a slightly smaller standard error. The two graphical horseshoe implementations perform better than the graphical lasso in terms of both precision and recall in most of the cases. The only exception is the third case, where the graphical lasso has the best performance in terms of precision but slightly smaller recall.

Another limitation of the Gibbs sampler is the time use. Figure 2 shows the CPU time on a logarithmic scale used to infer a network for different numbers of nodes p, using Gaussian graphical data sets with $n = 100$ observations. When Gibbs sampling is feasible to compute, the fast ECM estimator for the graphical horseshoe is substantially faster. For 90 nodes, fastGHS is 30 times faster than the Gibbs sampler. For even larger p, only the scalable ECM estimator can be used, which in practice is limited only by the available memory to store the current $p \times p$ matrix updates for $\boldsymbol{\Theta}$, $\boldsymbol{\Lambda}$ and $\boldsymbol{N}$.

6.2 Comparison with other joint network inference methods

We next investigate the performance of our joint graphical horseshoe estimator, jointGHS, through comparison with the Bayesian spike-and-slab joint graphical lasso of Li et al. [2019b] and the joint graphical lasso of Danaher et al. [2014]. We restrict ourselves to a setting with $K = 2$ graphs with $p = 50$ nodes each because the latter methods are either infeasible within reasonable time (still running after multiple days) or very time consuming (taking more than 24 hours) for a larger number of nodes and graphs. By varying the similarity, i.e., proportion of common edges, between the graphs, we assess the performance of the methods in different settings. Namely, we consider six settings, with the similarity between the two graphs varying from 0% edge disagreement (i.e., the same edge set) to 100% edge disagreement (i.e., no common edges). For each setting, we construct two $p \times p$ precision matrices with the desired level of similarity and sample $N = 100$ data sets from each of the two corresponding multivariate Gaussian distributions, with $n_1 = 50$ observations for the first graph and $n_2 = 80$ observations for the second. In all settings, both graphs have true sparsity 0.04, corresponding to 49 edges.

Table 1: Simulation results for our ECM implementation of the graphical horseshoe (fastGHS), the Gibbs sampling implementation of the graphical horseshoe (GHS) and the graphical lasso (Glasso) applied to multivariate Gaussian data from graphs with different numbers of vertices p and observations n. The results are averaged over $N = 20$ simulations, and shows sparsity, precision and recall as well as their standard errors in parentheses. For each case the highest value of the precision is marked in bold, and so is the precision of any other method within one standard error of it.

Case	True sparsity	p	n	Method	Estimated Sparsity	Precision	Recall
1	0.04	50	100	Glasso	0.019 (0.006)	0.80 (0.12)	0.37 (0.08)
				GHS	0.017 (0.002)	**0.91** (0.06)	0.38 (0.05)
				fastGHS	0.017 (0.002)	**0.94** (0.04)	0.39 (0.05)
2	0.04	50	200	Glasso	0.020 (0.003)	0.88 (0.08)	0.44 (0.03)
				GHS	0.017 (0.002)	**0.98** (0.03)	0.42 (0.04)
				fastGHS	0.017 (0.002)	**0.99** (0.02)	0.43 (0.04)
3	0.02	100	100	Glasso	0.011 (0.002)	**0.61** (0.08)	0.33 (0.04)
				GHS	0.015 (0.001)	0.49 (0.06)	0.37 (0.04)
				fastGHS	0.015 (0.001)	0.46 (0.05)	0.35 (0.03)
4	0.02	100	200	Glasso	0.008 (0.001)	0.86 (0.06)	0.36 (0.02)
				GHS	0.009 (0.001)	**0.91** (0.05)	0.40 (0.03)
				fastGHS	0.009 (0.001)	**0.93** (0.05)	0.41 (0.04)

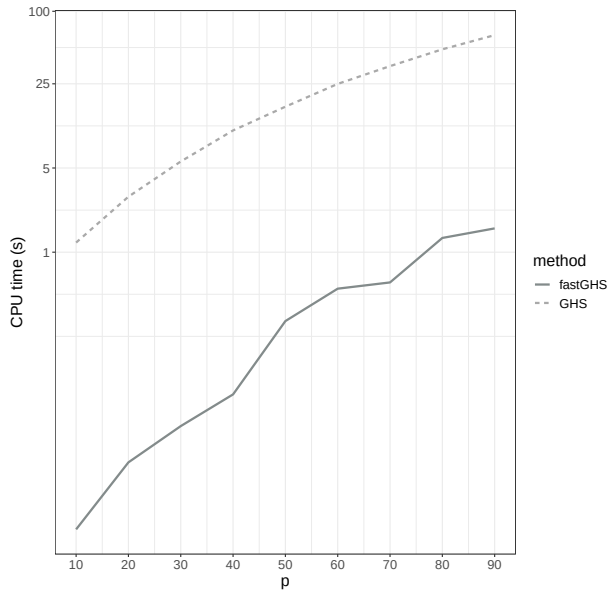

Figure 2: CPU time in seconds on a logarithmic scale used to infer a network for various numbers of nodes p with $n = 100$ observations, for our ECM implementation of the graphical horseshoe (fastGHS) and the Gibbs sampling implementation of the graphical horseshoe (GHS). The computations were performed on a 16-core Intel Xeon CPU, 2.60 GHz.

6.2.1 Estimation accuracy

Table 2 shows the performance of the joint network approaches. Starting with the joint graphical lasso, we see that its performance in terms of precision is low in all settings. When the default AIC-based selection criteria for sparsity- and similarity-selection is used, the method tends to severely over-select edges. This leads to high recall with many edges correctly identified, but very low precision, with almost ten times as many estimated edges than there are in the true network.

The joint graphical horseshoe (jointGHS) either outperforms of performs similarly to the spike-and-slab joint graphical lasso (SSJGL) in terms of precision in all settings. We further see that SSJGL has higher sparsity levels than jointGHS, which in general tends to give higher recall. This is true for highly similar networks, where SSJGL performs best in terms of recall. However, as the dissimilarity between the networks increases, the recall of the spike-and-slab method quickly decreases, and it becomes similar to that of jointGHS. This is due to the fact that SSJGL shrinks all precision matrices towards a common structure, so it over-selects edges that are absent in some networks while being present in others. This is exemplified further in Subsection 6.2.2. In terms of precision, the difference in performance between the two methods increases as the dissimilarity between the networks increases, and the joint graphical horseshoe largely outperforms its spike-and-slab counterpart in terms of all measures: the performance of the latter method is more sensitive to the degree of similarity and appears to deteriorate in settings where little information is shared between networks, while the joint graphical horseshoe adapts to this and so both precision and recall remains relatively high even for completely unrelated networks. This is due to the local scales λ_{ijk}, which flexibly capture isolated effects thanks to their heavy Cauchy tails. The versatility of the joint graphical lasso ensures that information is only shared to the degree that it improves the model fit, which makes it highly suitable to agnostic settings where the similarity between the networks of interest is unknown or not very high.

6.2.2 Ability to capture edges on the individual-graph level

Figure 3 further illustrates the benefits of the horseshoe heavy tailed local scales for capturing graph-specific edges. The estimated precision matrix elements of two graphs are plotted against each other, for our joint graphical horseshoe and for the spike-and-slab joint graphical lasso. The example is the same as in 5.4, with networks reconstructed from two data sets corresponding to $K = 2$ graphs with 40% edge agreement, with $p = 100$ nodes and $n_1 = 100$ and $n_2 = 150$ observations respectively. Both graphs have a true sparsity of 0.02.

The joint graphical horseshoe method identifies both common and graph-specific edges. When an edge is found to be present in only one of the graphs, the corresponding precision matrix element in the other graph is found to be zero, ensuring that a false positive is not reported due to excess shrinkage towards a common graph. This does result in the method in some instances only identifying an edge in one network even though it actually is present in both, meaning it is more inclined to false negatives than false positives. On the contrary, the spike-and-slab joint graphical lasso shrinks excessively towards a common graph and as a result almost no graph-specific edges are identified. While the spike-and-slab joint graphical lasso does well in capturing edges common to both graphs, an edge in one graph tends to lead to an inferred edge in both and hence a large number of false positives. This explains its excellent performance for very similar networks, but poorer performance as the similarity between two networks decreases in Table 2. In these settings, the joint graphical horseshoe successfully avoids over-shrinking the networks towards each other, whilst preserving graph-specific information. These specific traits of the graphical horseshoe makes it exceptionally suitable for joint approaches when preserving network specific features is also important.

6.3 Increased accuracy with joint modelling

We next investigate the statistical power gain obtained with jointGHS, as a function of the number of networks modelled jointly. To do this, we use the joint graphical horseshoe to reconstruct $K \in \{2, 4, 10\}$ graphs with $p = 50$ nodes and various similarity of the true graph structures. The larger K gives a setting where neither the spike-and-slab joint graphical lasso of (Li et al. [2019b]) nor the joint graphical lasso of Danaher et al. [2014] is feasible for inference. The results are averaged over $N = 40$ simulations, and show the precision and recall for the first estimated graph in each setting, reconstructed from $n = 80$ observations. All graphs have true sparsity 0.04.

Figure 4 shows how the precision and recall of the joint graphical horseshoe changes with the available information, i.e., when the number of graphs used in the joint procedure increases. It also illustrates how the accuracy changes with the level of edge disagreement between the networks. The results are compared to those of the single-network ECM implementation of the graphical horseshoe (fastGHS).

Although the simulated graph structure remains the same in all settings, the sparsity of the inferred jointGHS graphs will vary with K and the level of similarity. In order to make a fair comparison, we obtained single-network estimates

Table 2: Performance of the joint graphical horseshoe, (jointGHS), the spike-and-slab joint graphical lasso (SSJGL) and the joint graphical lasso (JGL) in simulations, reconstructing $K = 2$ graphs with various similarity of the true graph structures. The edge disagreement between the two graphs is shown as the percentage of edges in one network not present in the other. There are $p = 50$ nodes in each graph. The results are averaged over $N = 100$ simulations, and shows the sparsity, precision and recall for both of the $K = 2$ estimated graphs. The standard errors are shown in parentheses. For each case the highest value of the precision is marked in bold, and so is the precision of any other method within one standard error of it. The first graph is reconstructed from $n_1 = 50$ observations from the corresponding Gaussian graphical distribution, and the second from $n_2 = 80$ observations. All graphs have true sparsity 0.04.

		$n_1 = 50$			$n_2 = 80$		
Disagr. %	Method	Sparsity	Precision	Recall	Sparsity	Precision	Recall
0	JGL	0.312 (0.011)	0.11 (0.01)	0.83 (0.05)	0.269 (0.011)	0.13 (0.01)	0.91 (0.04)
	SSJGL	0.025 (0.003)	**0.85** (0.09)	0.53 (0.06)	0.025 (0.003)	0.85 (0.09)	0.53 (0.06)
	jointGHS	0.017 (0.002)	**0.82** (0.08)	0.34 (0.04)	0.016 (0.001)	**0.93** (0.06)	0.38 (0.03)
20	JGL	0.310 (0.02)	0.11 (0.01)	0.83 (0.05)	0.266 (0.017)	0.14 (0.01)	0.90 (0.04)
	SSJGL	0.025 (0.003)	0.77 (0.10)	0.48 (0.05)	0.025 (0.003)	0.79 (0.10)	0.49 (0.05)
	jointGHS	0.016 (0.002)	**0.79** (0.10)	0.32 (0.04)	0.016 (0.001)	**0.93** (0.06)	0.38 (0.03)
40	JGL	0.291 (0.048)	0.12 (0.02)	0.82 (0.06)	0.252 (0.059)	0.14 (0.05)	0.83 (0.08)
	SSJGL	0.020 (0.002)	**0.76** (0.09)	0.38 (0.04)	0.020 (0.002)	0.77 (0.09)	0.39 (0.05)
	jointGHS	0.016 (0.002)	**0.76** (0.08)	0.30 (0.04)	0.010 (0.002)	**0.98** (0.04)	0.25 (0.05)
60	JGL	0.291 (0.049)	0.12 (0.02)	0.82 (0.07)	0.247 (0.058)	0.15 (0.05)	0.84 (0.07)
	SSJGL	0.021 (0.003)	0.63 (0.09)	0.33 (0.05)	0.021 (0.003)	0.70 (0.08)	0.37 (0.05)
	jointGHS	0.016 (0.002)	**0.74** (0.08)	0.29 (0.04)	0.010 (0.002)	**0.98** (0.04)	0.25 (0.05)
80	JGL	0.299 (0.040)	0.11 (0.02)	0.81 (0.06)	0.259 (0.046)	0.14 (0.04)	0.89 (0.06)
	SSJGL	0.022 (0.004)	0.52 (0.08)	0.27 (0.04)	0.021 (0.003)	0.60 (0.09)	0.32 (0.05)
	jointGHS	0.015 (0.002)	**0.70** (0.11)	0.26 (0.04)	0.010 (0.002)	**0.98** (0.04)	0.25 (0.05)
100	JGL	0.313 (0.012)	0.11 (0.01)	0.83 (0.05)	0.251 (0.009)	0.16 (0.01)	0.98 (0.02)
	SSJGL	0.029 (0.003)	0.29 (0.06)	0.21 (0.03)	0.029 (0.003)	0.55 (0.06)	0.40 (0.05)
	jointGHS	0.015 (0.002)	**0.65** (0.12)	0.24 (0.04)	0.013 (0.002)	**0.95** (0.06)	0.31 (0.05)

with the same sparsity as the joint estimates in each setting, making the fastGHS results vary with K and the level of similarity as well. We refer to the appendix for the details.

As expected, the accuracy of the joint estimate increases with the number of graphs since more shared information is available. The joint approach clearly outperforms the single network approach in terms of both precision and recall, and the improvement grows with the number of graphs K. This applies to all levels of edge disagreement.

This example illustrates that the networks need not be very similar in order to benefit from the joint procedure, and hence we see that there is little to lose yet much to gain from using jointGHS for multiple graph estimation.

7 Application to a study of hotspot activity with stimulated monocyte expression

Returning to the monocyte QTL data set from Section 2, we now apply our proposed methodology to estimate conditional independence among the gene levels under genetic control. Specifically, the finding of Ruffieux et al. [2020] about the top hotspot SNP (rs6581889, on chromosome 12) being persistent across all four monocytic conditions (unstimulated cells, IFN-γ-, LPS 2h- & LPS 24h-stimulated cells) makes a joint graphical approach particularly relevant to study the interplay within and across the different gene networks. The number of genes associated with the top

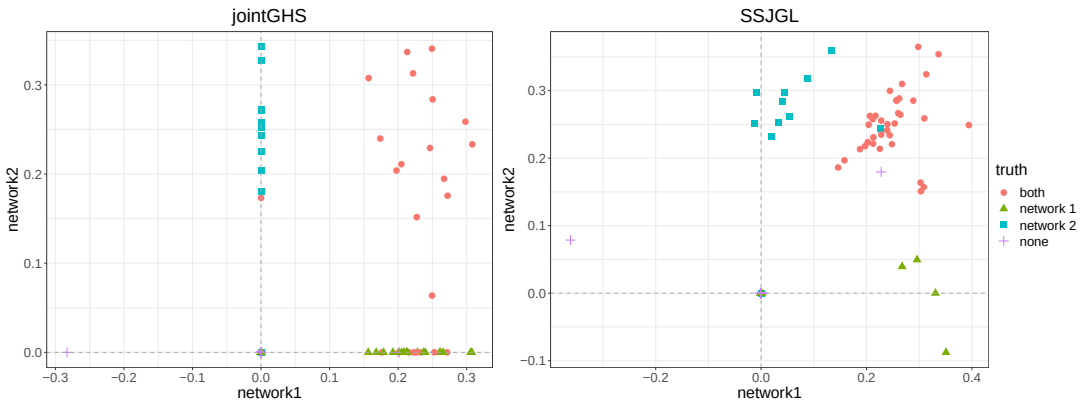

Figure 3: Comparison of estimated precision matrix elements (scaled by the diagonal as when finding partial correlations) of two networks with 40% edge agreement and $p = 100$ nodes, for the joint graphical horseshoe and for the spike-and-slab joint graphical lasso separately. The points' shape and color indicates whether the corresponding edge is simulated as present is both, one or neither network.

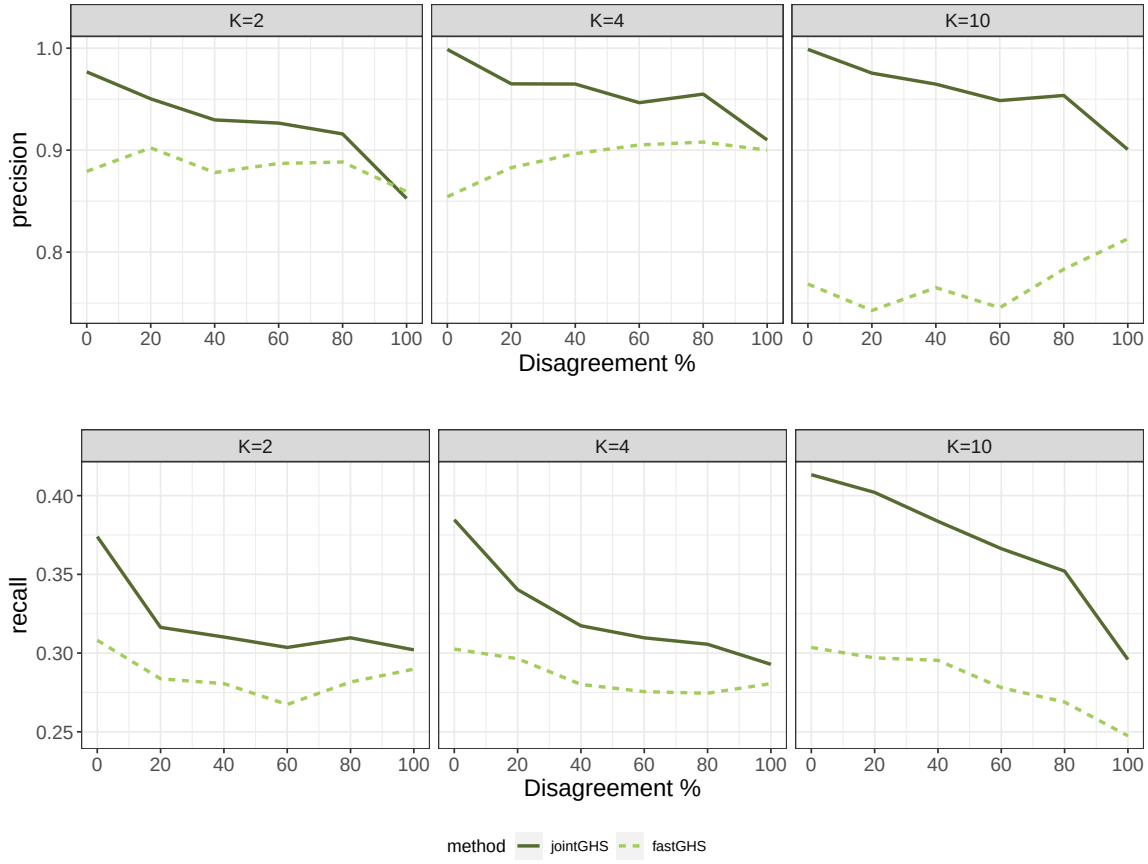

Figure 4: Performance of the joint graphical horseshoe (jointGHS) and single network graphical horseshoe (fastGHS), reconstructing $K \in \{2, 4, 10\}$ graphs with $p = 50$ nodes and various similarity of the true graph structures. The edge disagreement between the two graphs is shown as the percentage of edges in one network not present in the other. The results are averaged over $N = 40$ simulations, and show the precision and recall for the first estimated graph in each setting, reconstructed from $n = 80$ observations. All graphs have true sparsity 0.04.

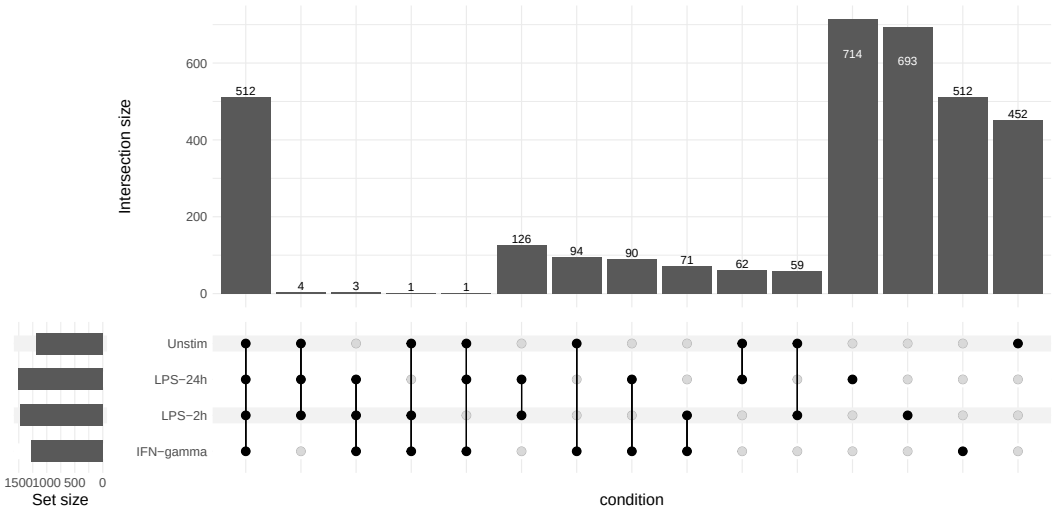

Figure 5: Upset plot of the joint graphical horseshoe graphs of the monocyte data, an alternative to a venn diagram showing the number of edges solely shared between conditions (Conway et al. [2017]). For each intersection, the number of edges shared only by the corresponding conditions is shown. The total number of edges for each condition is represented on the left barplot. Every possible intersection is represented by the bottom plot, and their occurence is shown in the top barplot.

hotspot *in each condition* was 294, 88, 16 and 215 respectively (permutation-based FDR < 0.05, Ruffieux et al. [2020]); hereafter we focus on the $p = 381$ genes associated with the hotspot in at least one condition. For further details on the data and preprocessing steps, we refer to Ruffieux et al. [2020].

We use jointGHS to jointly estimate the precision matrix, and hence network structure, of the genes in the four conditions. As discussed in Section 2, neither the spike-and-slab joint graphical lasso nor the joint graphical lasso is feasible for this problem. Figure 5 shows the number of edges sorely shared between the inferred networks of each condition. It indicates that many edges are common to all four networks, meaning that the joint method has identified a fair amount of shared information. There are no three conditions with more shared information compared to a last condition, but there are many pairs of conditions with edges shared only between them. We see that LPS 2h and LPS 24h is the pair with the most edges shared only between them, which is expected as the two conditions correspond to an exposition to differing durations of a same lipopolysaccharide activation. After this, the unstimulated and IFN-γ networks, and LPS 24h and IFN-γ networks, are the pairs with the most edges shared only between them. We also see that the joint graphical lasso has been able to capture many edges specific to each network, with the LPS 24h having the most unique edges. This can in part be due to the fact that this network has the most edges overall, a difference that should be kept in mind when considering the number of edges shared. Next, it would be sensible to ask whether groups of edges shared by two or more conditions pertain to known pathway activations and whether the pathways of genes in edges unique to one stimulated condition are indicative of some functional mechanisms specific to that stimulation. While such questions fall outside the scope of this publication, they illustrate how our joint network findings can motivate new mechanistic studies.

Hotspot control

Using permutation testing to derive empirical p-values, we find that the subnetwork of genes controlled by the top hotspot has significantly more links than the overall network ($p < 0.01$), except in the IFN-γ network, suggesting a hotspot-induced increase in activity. We similarly find that, in all conditions, there is a significant enrichment of genes associated with the top hotspot among the neighbours of the *cis* gene *LYZ*. This may suggest a mediation of the hotspot effect on *trans* genes via the *LYZ* gene — a hypothesis already examined in different studies (Fairfax et al. [2012], Ruffieux et al. [2021]) but which would require experimental validation or dedicated inspection, e.g., with Mendelian randomization analysis. The findings are summarised in Table 3.

Comparing the posterior output of jointGHS to that of our single-network ECM implementation of the graphical horseshoe (fastGHS) can highlight the biological insight gained from a joint approach. To get directly comparable networks, we find fastGHS estimates with the same sparsity levels as the jointGHS estimates, for each condition

Table 3: Sparsity of the jointGHS subnetwork of top hotspot controlled genes in the joint graphical horseshoe graph, as well as the overall sparsity of the full network, and the proportion of neighbour genes of LYZ and YEATS4 in the joint graphical horseshoe graph that are mediated by the top hotspot, as well as the overall proportion, in the different conditions. For each condition, if the proportion is higher than the overall proportion it is marked in bold. If the proportion is found to be significantly higher than what we can expect from a randomly sampled subset of genes of the same size it is marked by $*$ (empirical p-value < 0.05) or $**$ (empirical p-value < 0.01) .

		IFN-gamma	LPS 2h	LPS 24h	Unstim
Sparsity	Overall	0.018	0.020	0.021	0.016
	Controlled by top hotpot	**0.019**	**0.042****	**0.183****	**0.022***
Controlled by top hotspot	Overall	0.77	0.23	0.04	0.56
	LYZ neighbourhood	**0.78**	**0.42***	**0.33****	**0.77***
	YEATS4 neighbourhood	0.62	0.17	**0.22****	0.43

separately (we refer to the appendix for the details). The jointGHS finds that the subgraph of genes controlled by the top hotspot is denser than what fastGHS finds; this agrees with the expectation that the hotspot will trigger substantial activity among the controlled genes (Ruffieux et al. [2020]). The *cis* genes *LYZ* and *YEATS4* also have a more central role in the joint network, with more *trans-trans* edges among the neighbours of *LYZ* and *YEATS4*. The joint method also identifies more *cis-trans* relationships, which lends further support to the above *cis* mediation hypothesis. Moreover, many *trans* genes directly associated with the *cis* genes have very high degree, highlighting their interplay with other *trans* genes as potentially relevant for the disease-driving mechanisms. All these observations highlight the biological insight gained by sharing information across networks with jointGHS.

Hub genes

Investigating hub genes in the jointGHS networks can help gain better understanding of the immune response driving mechanisms. Table A.1 in the appendix shows the node degree of the genes with degree larger than the 90^{th} percentile in the respective jointGHS networks of the different conditions. Remarkably, the Autoimmune Regulator (*AIRE*) gene, which is highly expressed in monocytes, has by far the most links to other genes in all conditions but IFN-γ, where it has the second most. This gene is known to play an important role in immunity through gene and autoantigen activation and regulation, and negative selection of autoreactive T-cells in the thymus (Liston et al. [2003], Kyewski and Klein [2006], Peterson et al. [2008]). Mutations in this gene have been associated with autoimmune polyendocrinopathy-candidiasis-ectodermal dystrophy (*APECED*), distinguished by multi-organ autoimmunity (Mathis and Benoist [2007], Akirav et al. [2011]). Similarly, the arylformamidase (*AFMID*) gene, also known as kynurenine formamidase, is found to have amongst the higher degrees in all four conditions in addition to being associated with the top hotspot in all; it also has a link to the *cis* gene *LYZ* in all four conditions. Arylformamidase is a rate-limiting enzyme in tryptophan conversion, and deficiency is associated with immune system abnormalities (Hugill et al. [2015], Dobrovolsky et al. [2005]). Another gene with many links consistent across all conditions is Galectin-3 (*LGALS3*), whose expression is found to modulate T-cell growth and apoptosis (Yang et al. [1996]). Studies indicate that it affects numerous biological processes through specific interactions with a variety of intra- and extracellular proteins, and has a regulatory role in both innate and adaptive immunity (Dumic et al. [2006], Bernardes et al. [2006]). It has been found to be a negative regulator of lipopolysaccharide (LPS) mediated inflammation (Li et al. [2008], Fermino et al. [2011]), and lack of Galectin-3 has been directly linked to higher IFN-γ levels (Nishi et al. [2007], Radosavljevic et al. [2012]). Finally, the Aftiphilin (*AFTPH*) gene has many links in all conditions, including to the *cis* gene *YEATS4*. Aftiphilin has been found to be differentially expressed in the immune cells of tumor patients, suggesting it might affect tumor development through immune cell regulation (Huang et al. [2021]).

Network-specific activity

As illustrated in our simulation experiments, jointGHS is not only tailored to the detection of shared structures across networks, but it is also capable to effectively identify what differs between them. This feature is particularly relevant in the context of the monocyte study as it is informative to also examine specific effects of the different stimulations. The Cytochrome C Oxidase Subunit 6A1 (*COX6A1*) gene has large degree in both LPS 2h and LPS 24h and a moderately large degree in IFN-γ, but not in the unstimulated group. The oxidative phosphorylation pathway and immune system processes both include *COX6A1* (Wang et al. [2019]), and the gene has been shown to have key functions in the replication of influenza A viruses (Hao et al. [2008]), making it noteworthy that this gene's activity is found elevated

only in the stimulated conditions. For example, we identify the PHD-finger 1 protein encoding gene (*PHF1*) to have high degree only in the IFN-γ network, in which it is also found to be controlled by the top hotspot. Further, IFN-γ is the only condition in which it has a link to the *cis* gene *LYZ*. The PHD-finger 1 protein is an essential factor for epigenetic regulation and genome maintenance, and contains two kinds of histone reader modules; a Tudor domain and two PHD fingers, of which the function of the latter is less defined (Baker et al. [2008], Liu et al. [2018]). The centrality of *PHF1* in the network of the IFN-γ stimulated monocytes has implications for a potential role in the immune reaction, and provides a relevant alley for further studies. Additional details and discussion from our analysis can be found in the appendix.

While it is reassuring that our method identifies genes known from literature to be relevant, this type of validation is biased towards gene and protein functions that have already been explored. We believe though that jointGHS could serve to generate further unexplored hypotheses about genetic co-regulation and co-expression across the stimulated monocyte networks — this would deserve further follow-up research. More generally, our findings illustrate the potential of the joint graphical horseshoe for gaining deeper insight into the mechanisms at play among large networks of cellular and/or molecular variables for multiple conditions or tissues.

8 Discussion and Conclusions

We have introduced a novel scalable expectation conditional maximisation (ECM) algorithm for obtaining the posterior mode of the precision matrix in the graphical horseshoe, and a novel joint graphical horseshoe estimator for multiple network estimation. Through simulations, we have shown that the single network estimator achieves results that are comparable to the existing Gibbs sampler while being far more scalable. We have also shown that in a multiple network setting, our joint network estimator successfully shares information between networks while capturing their differences, outperforming comparable methods such as the joint graphical lasso and the spike-and-slab joint graphical lasso. This holds for any level of network similarity, even when there is little or no information to share between networks. This clear advantage of the joint graphical horseshoe can be attributed to the horseshoe heavy-tailed local scales, which are able to adapt even when no common information is found for a specific edge, so that network-specific edges can be identified. This means that the joint estimator can be used even at minimal levels of graph similarity and still outperform single-network methods — to our knowledge, no existing joint graphical modelling approach enjoys this property. Hence, jointGHS stands out as a joint approach capable to also pinpoint differences across networks, which, in practice, is often of great interest, sometimes even more than the identification of shared structures. Additionally, the scalability of the method makes it applicable to larger data sets than comparable methods. Finally, while our ECM implementation does not provide a fully Bayesian solution, parameter uncertainty could still be quantified using an additional bootstrapping procedure.

We have applied jointGHS to an eQTL monocyte data set with four conditions, whose joint graphical modelling relationships had not been investigated thus far. We found that the inferred networks indicated a hotspot-induced increase in *trans* gene regulatory activity in all conditions. More generally, compared to a separate analysis of the conditions, biologically-supported findings were stronger in the joint analysis and agreed more with previous results (Ruffieux et al. [2020]). Our results may have implications for the identification of genes driving a range of immune-mediated diseases, with some less studied genes coming up as putatively relevant in addition to those with known immune-related function. This application illustrates how jointGHS can be used to generate hypotheses about disease-driving mechanisms.

There are many possible extensions. For instance, a natural continuation given the increasing prevalence of longitudinal studies would be to formulate a time variant version of the model, with one network per point in time. This could be particularly profitable for studies aimed at understanding disease progression. An autoregression-like approach could be developed, where information is shared between successive time points.

To conclude, we have proposed a scalable method for single and joint graph inference that is suitable for a wide range of applications. Relevant settings include omics problems with single or multiple data types, subgroups or conditions, or just any type of data for which a multivariate Gaussian assumption is reasonable. Altogether, thanks to the flexibility of its shrinkage profile, the joint graphical horseshoe constitutes an efficient and robust framework to boost statistical power for the estimation of weak edge effects, as well as the detection of similarities and differences across multiple networks.

Software

The ECM graphical horseshoe approach for single or multiple networks has been implemented in the R packages fastGHS (`https://github.com/Camiling/fastGHS`) and jointGHS (`https://github.com/Camiling/jointGHS`),

and has all subroutines implemented in C++ for computational efficiency. R code for the simulations and data analyses in this paper is available at `https://github.com/Camiling/jointGHS_simulations` and `https://github.com/Camiling/jointGHS_analysis`.

Data

The CD14+ monocyte gene expression data and genotyping data collected from individuals with European ancestry used in this study is provided by Fairfax et al. [2012, 2014]. The raw expression data is generated with HumanHT-12 v4 arrays and freely available for downloading from ArrayExpress45 (accession E-MTAB-2232). The raw genotyping data is generated by Illumina HumanOmniExpress-12 arrays, and deposited in the European Genome-Phenome Archive (accessions EGAD00010000144 and EGAD00010000520). The genotyping data requires a data access agreement (see Fairfax et al. [2012, 2014] and `https://www.well.ox.ac.uk/research/research-groups/julian-knight-group/research-projects/data-access`.

Funding

This research is funded by the UK Medical Research Council programme MRC MC UU 00002/10 (C.L., H.R. and S.R.), Aker Scholarship (C.L.), Lopez–Loreta Foundation (H.R.) and Wellcome Intermediate Clinical Fellowship 01488/Z/16/Z (B.P.F.).

References

EP van Someren, Lodewyk FA Wessels, Eric Backer, and Marcel JT Reinders. Genetic network modeling. *Pharmacogenomics*, 3(4):507–525, 2002.

Konrad J Karczewski and Michael P Snyder. Integrative omics for health and disease. *Nature Reviews Genetics*, 19(5): 299–310, 2018.

Yunfan Li, Bruce A Craig, and Anindya Bhadra. The graphical horseshoe estimator for inverse covariance matrices. *Journal of Computational and Graphical Statistics*, pages 1–24, 2019a.

Benjamin P Fairfax, Peter Humburg, Seiko Makino, Vivek Naranbhai, Daniel Wong, Evelyn Lau, Luke Jostins, Katharine Plant, Robert Andrews, Chris McGee, et al. Innate immune activity conditions the effect of regulatory variants upon monocyte gene expression. *Science*, 343(6175):1246949, 2014.

Mark N Lee, Chun Ye, Alexandra-Chloé Villani, Towfique Raj, Weibo Li, Thomas M Eisenhaure, Selina H Imboywa, Portia I Chipendo, F Ann Ran, Kamil Slowikowski, et al. Common genetic variants modulate pathogen-sensing responses in human dendritic cells. *Science*, 343(6175):1246980, 2014.

Sarah Kim, Jessica Becker, Matthias Bechheim, Vera Kaiser, Mahdad Noursadeghi, Nadine Fricker, Esther Beier, Sven Klaschik, Peter Boor, Timo Hess, et al. Characterizing the genetic basis of innate immune response in tlr4-activated human monocytes. *Nature Communications*, 5(1):1–7, 2014.

Chen Yao, Roby Joehanes, Andrew D Johnson, Tianxiao Huan, Chunyu Liu, Jane E Freedman, Peter J Munson, David E Hill, Marc Vidal, and Daniel Levy. Dynamic role of trans regulation of gene expression in relation to complex traits. *The American Journal of Human Genetics*, 100(4):571–580, 2017.

Hélène Ruffieux, Anthony C Davison, Jörg Hager, Jamie Inshaw, Benjamin P Fairfax, Sylvia Richardson, and Leonardo Bottolo. A global-local approach for detecting hotspots in multiple-response regression. *The Annals of Applied Statistics*, 14(2):905–928, 2020.

Benjamin P Fairfax, Seiko Makino, Jayachandran Radhakrishnan, Katharine Plant, Stephen Leslie, Alexander Dilthey, Peter Ellis, Cordelia Langford, Fredrik O Vannberg, and Julian C Knight. Genetics of gene expression in primary immune cells identifies cell type–specific master regulators and roles of hla alleles. *Nature Genetics*, 44(5):502–510, 2012.

Zehang Li, Tyler Mccormick, and Samuel Clark. Bayesian joint spike-and-slab graphical lasso. In *International Conference on Machine Learning*, pages 3877–3885. PMLR, 2019b.

Patrick Danaher, Pei Wang, and Daniela M Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):373–397, 2014.

Eric D Kolaczyk. *Statistical Analysis of Network Data: Methods and Models*. Springer Science & Business Media, 2009.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34:1436–1462, 2006.

Jianqing Fan, Yang Feng, and Yichao Wu. Network exploration via the adaptive lasso and scad penalties. *The Annals of Applied Statistics*, 3(2):521, 2009.

Hao Wang et al. Bayesian graphical lasso models and efficient posterior computation. *Bayesian Analysis*, 7(4):867–886, 2012.

Hao Wang. Scaling it up: Stochastic search structure learning in graphical models. *Bayesian Analysis*, 10(2):351–377, 2015.

Carlos M Carvalho, Nicholas G Polson, and James G Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97 (2):465–480, 2010.

Zehang Richard Li and Tyler H McCormick. An expectation conditional maximization approach for gaussian graphical models. *Journal of Computational and Graphical Statistics*, 28(4):767–777, 2019.

Xiao-Li Meng and Donald B Rubin. Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Enes Makalic and Daniel F Schmidt. A simple sampler for the horseshoe estimator. *IEEE Signal Processing Letters*, 23 (1):179–182, 2015.

Jeong Hwan Kook, Kelly A Vaughn, Dana M DeMaster, Linda Ewing-Cobbs, and Marina Vannucci. Bvar-connect: A variational bayes approach to multi-subject vector autoregressive models for inference on brain connectivity networks. *Neuroinformatics*, 19(1):39–56, 2021.

James G Scott and James O Berger. Bayes and empirical-bayes multiplicity adjustment in the variable-selection problem. *The Annals of Statistics*, pages 2587–2619, 2010.

Stéphanie L Van Der Pas, Bas JK Kleijn, and Aad W Van Der Vaart. The horseshoe estimator: Posterior concentration around nearly black vectors. *Electronic Journal of Statistics*, 8(2):2585–2618, 2014.

Anindya Bhadra, Jyotishka Datta, Nicholas G Polson, and Brandon Willard. The horseshoe+ estimator of ultra-sparse signals. *Bayesian Analysis*, 12(4):1105–1131, 2017.

Juho Piironen and Aki Vehtari. Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11(2):5018–5051, 2017.

Donald R Williams, Juho Piironen, Aki Vehtari, and Philippe Rast. Bayesian estimation of gaussian graphical models with predictive covariance selection. *arXiv preprint arXiv:1801.05725*, 2018.

Hirotsugu Akaike, Boris Nikolaevich Petrov, and Frigyes Csaki. Information theory and an extension of the maximum likelihood principle. *Second International Symposium on Information Theory*, pages 267–281, 1973.

Veronika Ročková and Edward I George. The spike-and-slab lasso. *Journal of the American Statistical Association*, 113(521):431–444, 2018.

Donald B Rubin. The bayesian bootstrap. *The Annals of Statistics*, pages 130–134, 1981.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. URL `http://www.R-project.org/`.

Jake R Conway, Alexander Lex, and Nils Gehlenborg. Upsetr: an r package for the visualization of intersecting sets and their properties. *Bioinformatics*, 2017.

Hélène Ruffieux, Benjamin P Fairfax, Isar Nassiri, Elena Vigorito, Chris Wallace, Sylvia Richardson, and Leonardo Bottolo. Epispot: an epigenome-driven approach for detecting and interpreting hotspots in molecular qtl studies. *The American Journal of Human Genetics*, 108(6):983–1000, 2021.

Adrian Liston, Sylvie Lesage, Judith Wilson, Leena Peltonen, and Christopher C Goodnow. Aire regulates negative selection of organ-specific t cells. *Nature Immunology*, 4(4):350–354, 2003.

Bruno Kyewski and Ludger Klein. A central role for central tolerance. *Annu. Rev. Immunol.*, 24:571–606, 2006.

Pärt Peterson, Tõnis Org, and Ana Rebane. Transcriptional regulation by aire: molecular mechanisms of central tolerance. *Nature Reviews Immunology*, 8(12):948–957, 2008.

Diane Mathis and Christophe Benoist. A decade of aire. *Nature Reviews Immunology*, 7(8):645–650, 2007.

Eitan M Akirav, Nancy H Ruddle, and Kevan C Herold. The role of aire in human autoimmune disease. *Nature Reviews Endocrinology*, 7(1):25–33, 2011.

Alison J Hugill, Michelle E Stewart, Marianne A Yon, Fay Probert, I Jane Cox, Tertius A Hough, Cheryl L Scudamore, Liz Bentley, Gary Wall, Sara E Wells, et al. Loss of arylformamidase with reduced thymidine kinase expression leads to impaired glucose tolerance. *Biology Open*, 4(11):1367–1375, 2015.

Vasily N Dobrovolsky, John F Bowyer, Michael K Pabarcus, Robert H Heflich, Lee D Williams, Daniel R Doerge, Björn Arvidsson, Jonas Bergquist, and John E Casida. Effect of arylformamidase (kynurenine formamidase) gene inactivation in mice on enzymatic activity, kynurenine pathway metabolites and phenotype. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1724(1-2):163–172, 2005.

Ri-Yao Yang, DANIEL K Hsu, and FU-TONG LIu. Expression of galectin-3 modulates t-cell growth and apoptosis. *Proceedings of the National Academy of Sciences*, 93(13):6737–6742, 1996.

Jerka Dumic, Sanja Dabelic, and Mirna Flögel. Galectin-3: an open-ended story. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1760(4):616–635, 2006.

Emerson Soares Bernardes, Neide M Silva, Luciana Pereira Ruas, Jose Roberto Mineo, Adriano Motta Loyola, Daniel K Hsu, Fu-Tong Liu, Roger Chammas, and Maria Cristina Roque-Barreira. Toxoplasma gondii infection reveals a novel regulatory role for galectin-3 in the interface of innate and adaptive immunity. *The American Journal of Pathology*, 168(6):1910–1920, 2006.

Yubin Li, Mousa Komai-Koma, Derek S Gilchrist, Daniel K Hsu, Fu-Tong Liu, Tabitha Springall, and Damo Xu. Galectin-3 is a negative regulator of lipopolysaccharide-mediated inflammation. *The Journal of Immunology*, 181(4): 2781–2789, 2008.

Marise Lopes Fermino, Claudia Danella Polli, Karina Alves Toledo, Fu-Tong Liu, Dan K Hsu, Maria Cristina Roque-Barreira, Gabriela Pereira-da Silva, Emerson Soares Bernardes, and Lise Halbwachs-Mecarelli. Lps-induced galectin-3 oligomerization results in enhancement of neutrophil activation. *PloS ONE*, 6(10):e26004, 2011.

Yumiko Nishi, Hideki Sano, Tatsuo Kawashima, Tomoaki Okada, Toshihisa Kuroda, Kyoko Kikkawa, Sayaka Kawashima, Masaaki Tanabe, Tsukane Goto, Yasuo Matsuzawa, et al. Role of galectin-3 in human pulmonary fibrosis. *Allergology International*, 56(1):57–65, 2007.

Gordana Radosavljevic, Vladislav Volarevic, Ivan Jovanovic, Marija Milovanovic, Nada Pejnovic, Nebojsa Arsenijevic, Daniel K Hsu, and Miodrag L Lukic. The roles of galectin-3 in autoimmunity and tumor progression. *Immunologic Research*, 52(1):100–110, 2012.

Zhong-Yin Huang, Ming-Ming Shao, Jian-Chu Zhang, Feng-Shuang Yi, Juan Du, Qiong Zhou, Feng-Yao Wu, Sha Li, Wei Li, Xian-Zhen Huang, et al. Single-cell analysis of diverse immune phenotypes in malignant pleural effusion. *Nature Communications*, 12(1):1–12, 2021.

Lamei Wang, Yu Huang, Xiaolong Wang, and Yulin Chen. Label-free lc-ms/ms proteomics analyses reveal proteomic changes accompanying mstn ko in c2c12 cells. *BioMed Research International*, 2019, 2019.

Linhui Hao, Akira Sakurai, Tokiko Watanabe, Ericka Sorensen, Chairul A Nidom, Michael A Newton, Paul Ahlquist, and Yoshihiro Kawaoka. Drosophila rnai screen identifies host genes important for influenza virus replication. *Nature*, 454(7206):890–893, 2008.

Lindsey A Baker, C David Allis, and Gang G Wang. Phd fingers in human diseases: disorders arising from misinterpreting epigenetic marks. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 647(1-2):3–12, 2008.

Ruiqiong Liu, Jie Gao, Yang Yang, Rongfang Qiu, Yu Zheng, Wei Huang, Yi Zeng, Yongqiang Hou, Shuang Wang, Shuai Leng, et al. Phd finger protein 1 (phf1) is a novel reader for histone h4r3 symmetric dimethylation and coordinates with prmt5–wdr77/crl4b complex to promote tumorigenesis. *Nucleic Acids Research*, 46(13):6608–6626, 2018.

Han Liu, Kathryn Roeder, and Larry Wasserman. Stability approach to regularization selection (stars) for high dimensional graphical models. In *Advances in Neural Information Processing Systems*, pages 1432–1440, 2010.

Ido Amit, Ami Citri, Tal Shay, Yiling Lu, Menachem Katz, Fan Zhang, Gabi Tarcic, Doris Siwak, John Lahad, Jasmine Jacob-Hirsch, et al. A module of negative feedback regulators defines growth factor signaling. *Nature Genetics*, 39 (4):503–512, 2007.

Josue E Rodriguez and Donald R Williams. Bayesian bootstrapped correlation coefficients. *The Quantitative Methods for Psychology*, 18(1):39–54, 2022. doi:10.20982/tqmp.18.1.p039.

A Appendix

A.1 Derivation of the ECM algorithm

A.1.1 Deriving the objective function

The objective function for the graphical horseshoe ECM algorithm is as follows:

$$
\begin{aligned}
Q(\boldsymbol{\Theta}, \boldsymbol{\Lambda}, \tau | \boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)}, \tau^{(l)}) =& \mathrm{E}_{\boldsymbol{N}, \xi | \boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)}, \tau^{(l)}, \boldsymbol{S}}[\log p(\boldsymbol{\Theta}, \boldsymbol{\Lambda}, \tau, \boldsymbol{N}, \xi | \boldsymbol{S}) | \boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)}, \tau^{(l)}] \\
=& \mathrm{E}_{\cdot|\cdot}\big[\log\big\{p(\boldsymbol{S}|\boldsymbol{\Theta})p(\boldsymbol{\Theta}|\boldsymbol{\Lambda}, \tau)p(\boldsymbol{\Lambda}|\boldsymbol{N})p(\boldsymbol{N})p(\tau|\xi)p(\xi)\big\} | \boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)}, \tau^{(l)}\big] \\
& + \text{const.} \\
=& \mathrm{E}_{\cdot|\cdot}\bigg[\frac{n}{2}\log\left(\det \boldsymbol{\Theta}\right) - \frac{1}{2}\mathrm{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + \sum_{i<j}\Big\{ -\frac{1}{2}\log(\lambda_{ij}^2\tau^2) - \frac{\theta_{ij}^2}{2\lambda_{ij}^2\tau^2} \\
& + \frac{1}{2}\log\left(\frac{1}{\nu_{ij}}\right) + \frac{3}{2}\log\left(\frac{1}{\lambda_{ij}^2}\right) - \frac{1}{\nu_{ij}\lambda_{ij}^2} + \frac{3}{2}\log\left(\frac{1}{\nu_{ij}}\right) - \frac{1}{\nu_{ij}}\Big\} \\
& + \frac{1}{2}\log\left(\frac{1}{\xi}\right) + \frac{3}{2}\log\left(\frac{1}{\tau^2}\right) - \frac{1}{\tau^2\xi} + \frac{3}{2}\log\left(\frac{1}{\xi}\right) - \frac{1}{\xi}\bigg] + \text{const.} \\
=& \frac{n}{2}\log\left(\det \boldsymbol{\Theta}\right) - \frac{1}{2}\mathrm{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + \sum_{i<j}\Big\{ -\log\left(\lambda_{ij}\right) - \log\left(\tau\right) - \frac{\theta_{ij}^2}{2\tau^2\lambda_{ij}^2} \\
& - \frac{1}{2}\mathrm{E}_{\cdot|\cdot}\big[\log\left(\nu_{ij}\right)\big] - 3\log\left(\lambda_{ij}\right) - \frac{1}{\lambda_{ij}^2}\mathrm{E}_{\cdot|\cdot}\left[\frac{1}{\nu_{ij}}\right] - \frac{3}{2}\mathrm{E}_{\cdot|\cdot}\big[\log\left(\nu_{ij}\right)\big] \\
& - \mathrm{E}_{\cdot|\cdot}\left[\frac{1}{\nu_{ij}}\right]\Big\} - 2\mathrm{E}_{\cdot|\cdot}\big[\log\left(\xi\right)\big] - 3\log\left(\tau\right) - \left(1 + \frac{1}{\tau^2}\right)\mathrm{E}_{\cdot|\cdot}\left[\frac{1}{\xi}\right] + \text{const.} \\
=& \frac{n}{2}\log\left(\det \boldsymbol{\Theta}\right) - \frac{1}{2}\mathrm{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + \sum_{i<j}\Big\{ -4\log\left(\lambda_{ij}\right) - \frac{\theta_{ij}^2}{2\tau^2\lambda_{ij}^2} \\
& - 2\mathrm{E}_{\cdot|\cdot}\big[\log\left(\nu_{ij}\right)\big] - \mathrm{E}_{\cdot|\cdot}\left[\frac{1}{\nu_{ij}}\right] - \frac{1}{\lambda_{ij}^2}\mathrm{E}_{\cdot|\cdot}\left[\frac{1}{\nu_{ij}}\right]\Big\} - \left(\frac{p(p-1)}{2} + 3\right)\log\left(\tau\right) \\
& - 2\mathrm{E}_{\cdot|\cdot}\big[\log\left(\xi\right)\big] - \left(1 + \frac{1}{\tau^2}\right)\mathrm{E}_{\cdot|\cdot}\left[\frac{1}{\xi}\right] + \text{const.} \\
=& \frac{n}{2}\log\left(\det \boldsymbol{\Theta}\right) - \frac{1}{2}\mathrm{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + \sum_{i<j}\Big\{ -4\log\left(\lambda_{ij}\right) - \frac{\theta_{ij}^2}{2\tau^2\lambda_{ij}^2} - 2\mathrm{E}_{\cdot|\cdot}\big[\log\left(\nu_{ij}\right)\big] \\
& - \left(\frac{1}{\lambda_{ij}^2} + 1\right)\mathrm{E}_{\cdot|\cdot}\left[\frac{1}{\nu_{ij}}\right]\Big\} - \left(\frac{p(p-1)}{2} + 3\right)\log\left(\tau\right) - 2\mathrm{E}_{\cdot|\cdot}\big[\log\left(\xi\right)\big] \\
& - \left(1 + \frac{1}{\tau^2}\right)\mathrm{E}_{\cdot|\cdot}\left[\frac{1}{\xi}\right] + \text{const.}
\end{aligned}
\tag{13}
$$

A.1.2 Deriving the E-step

Given the updates for the ν_{ij}'s and τ

$$
\begin{aligned}
&\mathrm{E}_{\cdot|\cdot}\left\{\log\left(\nu_{ij}\right)\right\} = \log\left(1+\frac{1}{\lambda_{ij}^{(l)^2}}\right) - \psi(1),\\
&\mathrm{E}_{\cdot|\cdot}\left(\frac{1}{\nu_{ij}}\right) = \frac{1}{1+1/\lambda_{ij}^{(l)^2}} = \frac{\lambda_{ij}^{(l)^2}}{\lambda_{ij}^{(l)^2}+1} =: \lambda_{ij}^{*^{(l)}},\\
&\mathrm{E}_{\cdot|\cdot}\left\{\log\left(\xi\right)\right\} = \log\left(1+\frac{1}{\tau^{(l)^2}}\right) - \psi(1),\\
&\mathrm{E}_{\cdot|\cdot}\left(\frac{1}{\xi}\right) = \frac{1}{1+1/\tau^{(l)^2}} = \frac{\tau^{(l)^2}}{\tau^{(l)^2}+1} =: \tau^{*^{(l)}},
\end{aligned}
\tag{14}
$$

the objective function becomes

$$
\begin{aligned}
Q(\boldsymbol{\Theta},\boldsymbol{\Lambda},\tau|\boldsymbol{\Theta}^{(l)},\boldsymbol{\Lambda}^{(l)},\tau^{(l)}) =& \frac{n}{2}\log\left(\det\boldsymbol{\Theta}\right) - \frac{1}{2}\mathrm{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + \sum_{i<j}\Big\{-4\log\left(\lambda_{ij}\right) - \frac{\theta_{ij}^2}{2\tau^2\lambda_{ij}^2} - 2\log\left(1+\frac{1}{\lambda_{ij}^{(l)^2}}\right)\\
&- \left(\frac{1}{\lambda_{ij}^2}+1\right)\left(\frac{\lambda_{ij}^{(l)^2}}{\lambda_{ij}^{(l)^2}+1}\right)\Big\} - \left(\frac{p(p-1)}{2}+3\right)\log\left(\tau\right) - 2\log\left(1+\frac{1}{\tau^{(l)^2}}\right)\\
&- \left(1+\frac{1}{\tau^2}\right)\left(\frac{\tau^{(l)^2}}{\tau^{(l)^2}+1}\right) + \text{const.}\\
=& \frac{n}{2}\log\left(\det\boldsymbol{\Theta}\right) - \frac{1}{2}\mathrm{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + \sum_{i<j}\Big\{-4\log\left(\lambda_{ij}\right) - \frac{\theta_{ij}^2}{2\tau^2\lambda_{ij}^2} - 2\log\left(\frac{\lambda_{ij}^{(l)^2}+1}{\lambda_{ij}^{(l)^2}}\right)\\
&- \left(\frac{1}{\lambda_{ij}^2}+1\right)\left(\frac{\lambda_{ij}^{(l)^2}}{\lambda_{ij}^{(l)^2}+1}\right)\Big\} - \left(\frac{p(p-1)}{2}+3\right)\log\left(\tau\right) - 2\log\left(\frac{\tau^{(l)^2}+1}{\tau^{(l)^2}}\right)\\
&- \left(1+\frac{1}{\tau^2}\right)\left(\frac{\tau^{(l)^2}}{\tau^{(l)^2}+1}\right) + \text{const.}\\
=& \frac{n}{2}\log\left(\det\boldsymbol{\Theta}\right) - \frac{1}{2}\mathrm{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + \sum_{i<j}\Big\{-4\log\left(\lambda_{ij}\right) - \frac{\theta_{ij}^2}{2\tau^2\lambda_{ij}^2} + 2\log\left(\lambda_{ij}^{*^{(l)}}\right)\\
&- \left(\frac{1}{\lambda_{ij}^2}+1\right)\lambda_{ij}^{*^{(l)}}\Big\} - \left(\frac{p(p-1)}{2}+3\right)\log\left(\tau\right) + 2\log\left(\tau^{*^{(l)}}\right)\\
&- \left(1+\frac{1}{\tau^2}\right)\tau^{*^{(l)}} + \text{const.}\\
=& \frac{n}{2}\log\left(\det\boldsymbol{\Theta}\right) - \frac{1}{2}\mathrm{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + \sum_{i<j}\Big\{-4\log\left(\lambda_{ij}\right) - \frac{\theta_{ij}^2}{2\tau^2\lambda_{ij}^2} - \frac{\lambda_{ij}^{*^{(l)}}}{\lambda_{ij}^2}\Big\}\\
&- \left(\frac{p(p-1)}{2}+3\right)\log\left(\tau\right) - \frac{\tau^{*^{(l)}}}{\tau^2} + \text{const.}
\end{aligned}
\tag{15}
$$

A.1.3 Deriving the CM-step

The maximisation of the objective function (15) with respect to τ is easily found by solving

$$\begin{aligned}
\frac{\partial Q}{\partial \tau} &= 0 \\
\sum_{i<j} \left\{ \frac{\theta_{ij}^2}{\lambda_{ij}^2 \tau^3} \right\} - \left(\frac{p(p-1)}{2} + 3 \right) \frac{1}{\tau} + \frac{2}{\tau^3} \tau^{*^{(l)}} &= 0 \\
2 \sum_{i<j} \left\{ \frac{\theta_{ij}^2}{\lambda_{ij}^2} \right\} - (p(p-1)+6)\tau^2 + 4\tau^{*^{(l)}} &= 0
\end{aligned}$$

which has a closed-form solution that gives the update

$$\tau^{(l+1)^2} = \frac{2 \sum_{i<j} \left\{ \frac{\theta_{ij}^2}{\lambda_{ij}^2} \right\} + 4\tau^{*^{(l)}}}{p(p-1)+6}.$$

Similarly, we can maximise (13) with respect to the λ_{ij}^2's to find the updates for $\mathbf{\Lambda}$:

$$\begin{aligned}
\frac{\partial Q}{\partial \lambda_{ij}^2} &= 0 \\
-\frac{2}{\lambda_{ij}^2} + \frac{\theta_{ij}}{2\tau^2 (\lambda_{ij}^2)^2} + \frac{\lambda_{ij}^{*^{(l)}}}{(\lambda_{ij}^2)^2} &= 0 \\
2\lambda_{ij}^2 - \frac{\theta_{ij}^2}{2\tau^2} - \lambda_{ij}^{*^{(l)}} &= 0
\end{aligned}$$

which is solved by

$$\lambda_{ij}^{(l+1)^2} = \frac{\lambda_{ij}^{*^{(l)}} + \theta_{ij}^2/(2\tau^2)}{2}.$$

A.1.4 Deriving full conditional posteriors for multiple networks

The full conditional posteriors of the ν_{ij}'s depends on the λ_{ijk}'s of all K networks. They are given by

$$\begin{aligned}
p(\nu_{ij}|\cdot) &= p(\nu_{ij}|\{\lambda_{ijk}^2\}_{k=1}^K) \\
&\propto p(\{\lambda_{ijk}^2\}_{k=1}^K|\nu_{ij}) p(\nu_{ij}) \\
&= \left(\prod_{k=1}^K p(\lambda_{ijk}^2|\nu_{ij}) \right) p(\nu_{ij}) \\
&= \left(\prod_{k=1}^K \frac{1}{\nu_{ij}^{1/2}} \exp\left(-\frac{1}{\nu_{ij}\lambda_{ijk}^2} \right) \right) \frac{1}{\nu_{ij}^{3/2}} \exp\left(-\frac{1}{\nu_{ij}} \right) \\
&= \frac{1}{\nu_{ij}^{(K+3)/2}} \exp\left(-\frac{1}{\nu_{ij}} \left(1 + \sum_{k=1}^K \frac{1}{\lambda_{ijk}^2} \right) \right) \\
&\propto \text{InvGamma}\left(\frac{K+1}{2}, 1 + \sum_{k=1}^K \frac{1}{\lambda_{ijk}^2} \right).
\end{aligned}$$

A.2 Simulation study details

To evaluate the performance our proposed methodology, we have done comprehensive simulation studies in R (R Core Team [2013]). We have used our R packages `fastGHS` and `jointGHS` to perform the single-network graphical horseshoe ECM routine and the joint graphical horseshoe ECM routine respectively, and the code for the simulations is available on Github (`https://github.com/Camiling/jointGHS_simulations`).

A.3 Single network inference

The first part of the simulation studies compares our graphical horseshoe ECM implementation (fastGHS) to the graphical horseshoe Gibbs sampler and the graphical lasso. The performance of the methods is assessed on multivariate Gaussian simulated data sets.

Generating networks and data. We aim to generate data similar to our omic application of interest. We have done this using the R package `huge`. The package provides the function `huge.generator()`, which generates multivariate Gaussian data with the *scale-free property* as it is a known characteristic of multiomic data (Kolaczyk [2009]). Given a number p of vertices, `huge` constructs a precision matrix $\boldsymbol{\Theta}$ with p edges. For a desired number of observations n, a data set can then be generated from the resulting multivariate Gaussian distribution with covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{\Theta}^{-1}$ and expectation vector $\mathbf{0}$. This results in a n by p data matrix $\boldsymbol{X}$, where each column corresponds to a node. As for the values of the non-zero partial correlations, we let them be in $[0.1, 0.2]$, a range we found to reflect the values observed in our omic application. Such data is generated $N = 20$ times. In our simulations, we consider different settings, i.e., different combinations of p and n. When $p = 50$, the final graph has p edges and thus a *sparsity* of 0.04. When $p = 100$, the sparsity of the graph is 0.02.

Data analysis details. In our simulation study, we wish to compare the graph reconstruction accuracy of the Gibbs sampling implementation of Li et al. [2019b] to our ECM implementation of the graphical horseshoe (fastGHS). We also include the graphical lasso of Friedman et al. [2008], a state-of-the-art frequentist method for Gaussian graph reconstruction. For each setting we consider, and each generated data set, we obtain a precision matrix estimate $\widehat{\boldsymbol{\Theta}}$ from the data $\boldsymbol{X}$ of interest for each of the methods.

We perform the ordinary graphical lasso using the R function `huge`, while we perform the Gibbs sampling procedure using the code of Li et al. [2019b] translated by us from MATLAB to R. A translation is necessary for us to perform the simulation study in R, but we have ensured that it gives identical results to those of the original MATLAB implementation.

The sparsity level of the graphical lasso is selected using the Stability Approach to Regularisation Selection (StARS), a selection method based on model stability (Liu et al. [2010]), with variability threshold $\beta = 0.03$. For the Gibbs sampling procedure of Li et al. [2019b], we draw $nmc = 1000$ MCMC samples after $burnin = 100$ burn-in iterations. In fastGHS, we use a convergence tolerance threshold of 0.001 for the precision matrix estimate and an AIC convergence threshold of 0.1 for the global scale parameter selection. The results are averaged over the $N = 20$ simulations for all settings, and we report the precision and recall, as well as the sparsity of the estimates.

Computational time simulation details. To illustrate the scalability of our ECM implementation for the graphical horseshoe, we save the CPU time used to infer a network for different numbers of nodes p, using Gaussian graphical data sets with $n = 100$ observations. We limit ourselves to $p \leq 90$, to permit comparison to the less scalable Gibbs sampler of Li et al. [2019b]. We run all methods on a 16-core Intel Xeon CPU, 2.60 GHz. We use the same parameters as for the single network simulation study, and consider the time used for one run, i.e. for a given value of τ^2 in the joint graphical horseshoe or one initial value in the Gibbs sampler.

A.3.1 Multiple network inference

Generating Gaussian graphs and data of specific similarity. We generate data for the joint graphical horseshoe simulation study using the same method as in 6.1 in the main manuscript, with the `huge` package in R to generate graphs and drawing data from the corresponding multivariate Gaussian distributions. The main difference is that we now have $K = 2$ graphs instead of one. Because we wish to investigate different settings with various similarity of the $K = 2$ true graph structures, we can not generate the graph structures independently. Starting by generating the first graph using `huge`, we instead modify its precision matrix in order to obtain a second graph with the desired level of similarity. Specifically, for the initial precision matrix we create, we permute a certain percentage of the edges by randomly reallocating them. This allows us to generate graphs of various similarity, ranging from 0% edge disagreement (i.e. the same edge set) to 100% edge disagreement (i.e. no common edges). In all settings, both graphs have true sparsity 0.04. For each pair of graphs, we sample $N = 100$ data sets from each of the two corresponding multivariate Gaussian distributions, with $n_1 = 50$ observations for the first graph and $n_2 = 80$ observations for the second.

Data analysis details. For each pair of generated data sets, we use the different joint graph reconstruction methods to obtain precision matrix estimates and assess their performance. In addition to our joint graphical horseshoe approach, we consider the Bayesian spike-and-slab joint graphical lasso of Li et al. [2019b] and the joint graphical lasso of Danaher et al. [2014]. Li et al. [2019b] provide R code for the Bayesian spike-and-slab at `https://github.com/`

`richardli/SSJGL`, and the joint graphical lasso is implemented by Danaher et al. [2014] in the R package `JGL`. The spike-and-slab joint graphical lasso (SSJGL) has a wide range of parameters that must be selected, and we use the values Li et al. [2019b] use for their own simulations, including an EM convergence tolerance of 0.0001. In the joint graphical lasso, the sparsity- and similarity controlling parameters must be selected. We do this using the approach suggested by Danaher et al. [2014], considering a grid of two parameters to find the values minimising their proposed adapted AIC score. In our joint graphical horseshoe approach, we use a convergence tolerance threshold of 0.001 for the precision matrix estimate and an AIC convergence threshold of 0.1 for the global scale parameter selection. For both graphs, the results are averaged over the $N = 100$ simulations for all settings and methods, and we report the precision and recall as well as the sparsity of the estimates. When comparing jointGHS and fastGHS, we get the fastGHS estimate of each network to the same sparsity as the corresponding jointGHS estimate by setting the global scale parameter to a small value, and slowly increasing it until the same sparsity is achieved.

A.4 Monocyte data

A.4.1 Data analysis details

For the jointGHS application, we use a convergence tolerance of 0.001 for the precision matrices. We use the same tolerance when applying fastGHS to each condition separately. Due to the large size of the problem, with $289\ 560$ potential links to be inferred, we use a looser AIC convergence threshold of 5 for the global scale parameter selection to speed up computations. As previously discussed, inference is not very sensitive to this choice: a wide range of global scale parameter values tend to give similar results. When comparing the results from jointGHS and fastGHS, we get the fastGHS estimate of each condition to the same sparsity as the corresponding jointGHS estimate by setting the global scale parameter to a small value, and slowly increasing it until the same sparsity is achieved.

Model parameters. Figure A.1 shows the scaled precision matrix elements of the jointGHS estimates plotted against the common latent parameters in all four conditions. We see that while many edges are found to be in common meaning information has been shared, there are still many network-specific edges that have been inferred. Thanks to the heavy horseshoe tail, we are able to capture these edges even though no common information about them is found between the conditions.

Degree distribution. Table A.1 shows the node degree of the genes with degree larger than the 90^{th} percentile in the respective jointGHS networks of the different conditions.

Cis genes and their neighbours. To investigate the role of the two *cis* genes *LYZ* and *YEATS4* in the different conditions, Figure A.2 shows their edges in the four conditions. It is clear that the two genes have a number of neighbours, and thus are relatively influential in the networks. These neighbours are fairly spread out on the chromosomes, in accordance with our expectations about *cis-trans* effects (Ruffieux et al. [2020]). While several edges are common to all conditions, we see that jointGHS identifies many edges on the specific condition level. Further, we see that a lot of the edges represent negative partial correlations, implying that the *cis* genes could have a role in the down-regulation as well as the up-regulation of the *trans* genes.

Investigating the neighbourhood of the two *cis* genes in each condition can also be useful for understanding the possible mediation effect of *LYZ* and *YEATS4*. Table A.2 shows the node degree of all neighbours in the jointGHS networks of the different conditions. As we see, *AFMID* is a neighbour of *LYZ* in all four conditions. Given its central role in each network, this potential interplay between *LYZ* and *AFMID* is noteworthy. Another notable observation is that *PHF1* is a neighbour of a *cis* gene only in IFN-γ. Given its importance only in this network, the possible association between *LYZ* and *PHF1* may provide further insight into the role of this gene. Another neighbour of *LYZ* in all conditions is MAF BZIP Transcription Factor F (*MAFF*), which is a regulator for growth factor signaling (Amit et al. [2007]). We also observe that *AFTPH*, which we found to have a high degree in all conditions, is a neighbour of *YEATS4* in all four conditions.

It can be relevant to investigate whether the neighbours of the two *cis* genes tend to be have more associations with the top hotspot. Table 3 in the main manuscript shows the proportion of the neighbours of *LYZ* and *YEATS4* respectively that are mediated by the top hotspot in each condition, and compares it to the total proportion of top hotspot controlled genes. We use permutation testing to assess whether more of the neighbours of each gene are controlled by the top hotspot than we would expect from a gene set of the same size; by randomly sampling 10000 gene sets with the same size as the neighbourhood, we calculate empirical p-values for the observed proportion of controlled neighbours. We see that *YEATS4* does not have significantly more mediated neighbours than the overall fraction, except in LPS 24h. *LYZ*, on the other hand, has, at a significance level 0.05, more mediated neighbours in all conditions.

Table A.1 : The genes with node degree larger than the 90^{th} percentile in the respective jointGHS networks of the different conditions. The genes that have node degree in the upper 10% in all four conditions are marked in bold. The genes that only have node degree in the upper 10% in one condition are marked in red. The genes that are controlled by the top hotspot in the condition in question are marked with $*$.

IFN-gamma		LPS-2h		LPS-24h		Unstimulated	
Gene	Degree	Gene	Degree	Gene	Degree	Gene	Degree
ACBD5*	33	**AIRE**	37	**AIRE**	38	**AIRE***	40
AIRE*	25	B3GNT5	28	TNK2	25	**AFMID***	33
AFMID*	24	**AFMID***	26	**AFMID***	25	PHRF1	24
LGALS3	23	**LGALS3**	24	COX6A1	24	**LGALS3**	23
SLC3A2	20	COX6A1	24	**LGALS3**	22	NBPF8	20
PHF1*	20	ACBD5	24	TBC1D9	21	AKR1D1*	20
*YEATS4**	18	**AFTPH**	22	EPSTI1	21	**AFTPH**	20
PTPLAD2*	18	PRPF8	21	CYP27A1	21	GIMAP1	19
GIMAP1	18	EPSTI1	21	YEATS4*	20	STAG3L3	18
COPZ1*	18	**IMPDH1**	19	STK24	20	**SLC3A2**	18
TNK2	17	GIMAP1	19	STAG3L3	20	ADAMTS4*	18
STK24	17	BIRC3	19	**AFTPH**	20	VHL*	17
SORL1	17	TBC1D15	18	CCL20	19	SGK3*	17
SNX17	17	**SORL1**	18	TNFSF14	18	GSDM1*	17
RELB	17	**SLC3A2**	18	**SORL1**	18	ZNF845	16
CYP27A1	17	CYP27A1	18	**SLC3A2**	18	YEATS4*	16
NLRP3	16	BFAR	18	**RELB**	18	TBC1D9	16
NBPF8	16	SNX17	17	PTPLAD2	18	KIAA0101*	16
KIAA0101*	16	PION	17	GNB4	18	JARID2	16
EPSTI1	16	NOD1	17	CRTC3	18	GLTSCR1	16
COX6A1	16	NLRP3	17	CD72	18	PTPLAD2	15
B3GNT5	16	ALPP	17	BATF3	18	LYZ*	15
SMC4	15	ZYX	16	TLK1	17	FLJ38717	15
PRPF8*	15	SGK3	16	SMC4	17	COX19	15
MAFF*	15	**RELB**	16	**IMPDH1**	17	ALPP*	15
LOC728457	15	NBPF8	16	HNRPR	17	AGTRAP*	15
IMPDH1*	15	LOC641522	16	DENND4C	17	ACBD5*	15
GSDM1*	15	KIAA0101*	16	COPZ1	17	TLK1	14
BATF3	15	HNRPR	16	CHRNA5*	17	**SORL1**	14
TNFSF15*	14	GTF2IRD2B	16	C4orf34	17	**RELB**	14
STAG3L3*	14	AGTRAP	16	TPM4	16	PCDHB9	14
SGK3*	14			SNX17	16	NLRP3	14
NFIL3	14			PLEKHB2	16	KLHL8*	14
HNRPR	14			PHRF1	16	**IMPDH1**	14
FLJ38717*	14			NFIL3	16	IL12RB1	14
BIRC3*	14					C4orf34*	14
AFTPH*	14					C12orf43	14
ADAMTS4*	14					BIRC3*	14

Table A.2 : The node degree of the neighbours of the cis genes LYZ and YEATS4 in the jointGHS networks of the different conditions. For each of the two cis genes, the genes that are a neighbour in all four conditions are marked in bold. The genes that are only a neighbour in one of the conditions are marked in red. The genes that are controlled by the top hotspot rs6581889 in the condition in question are marked with $*$.

LYZ

IFN-gamma		LPS-2h		LPS-24h		Unstimulated	
Gene	Degree	Gene	Degree	Gene	Degree	Gene	Degree
AFMID*	24	**AFMID***	26	**AFMID***	25	**AFMID***	33
LGALS3	23	LGALS3	24	YEATS4*	20	PHRF1	24
PHF1*	20	NDUFV3*	13	**MAFF**	14	LGALS3	23
NBPF8*	16	**MAFF***	12	**SNRNP48**	12	LOC202781	13
MAFF*	15	TPM3	11	MBD4	12	**KLHL28***	11
KLHL28*	11	**SNRNP48**	11	LRRFIP1	10	**SNRNP48***	10
SNRNP48*	9	LOC653086	11	**KLHL28***	10	NCAPD2*	10
LIN52*	9	MBOAT2	10	**LOC100128098**	8	**MAFF***	8
LOC100128098*	7	**KLHL28***	10	KIAA1751	8	**LOC100128098***	7
		TBCCD1	7			TBCCD1*	6
		MINK1	7			RAG1AP1*	6
		LOC100128098*	6			MEFV*	5
						MINK1*	4

YEATS4

IFN-gamma		LPS-2h		LPS-24h		Unstimulated	
Gene	Degree	Gene	Degree	Gene	Degree	Gene	Degree
AIRE*	25	AIRE	37	LGALS3	22	**AFTPH**	20
LGALS3	23	**AFTPH**	22	**TBC1D9**	21	**TBC1D9**	16
COPZ1*	18	**IMPDH1**	19	**AFTPH**	20	GLTSCR1	16
RELB	17	SORL1	18	**TLK1**	17	**TLK1**	14
IMPDH1*	15	**TBC1D9**	15	**IMPDH1**	17	**IMPDH1**	14
AFTPH*	14	HIST1H2BD	15	**COPZ1**	17	USP49*	13
TLK1	13	**TLK1**	14	AKR1D1	15	**COPZ1**	13
SLC4A5*	13	**COPZ1**	13	SC4MOL	14	BFAR	13
TBC1D15	12	ZNF131*	11	ZNF131*	11	SLC4A5*	11
CHST12*	12	MEFV	10	LYZ*	11	**TP53BP2***	8
SMCR5*	11	**TP53BP2***	9	ZNF738	10	DDX51*	7
LYRM7	11	**TMEM106A**	9	LRRFIP1	10	**TMEM106A***	6
TP53BP2*	10			XRCC2	9	TMEM128*	5
SC4MOL*	10			**TP53BP2***	9	CDK5RAP2	5
TMEM106A*	9			**TMEM106A**	9		
TBC1D9	9			MFSD11	8		
				DDX51*	7		
				TIPRL	6		

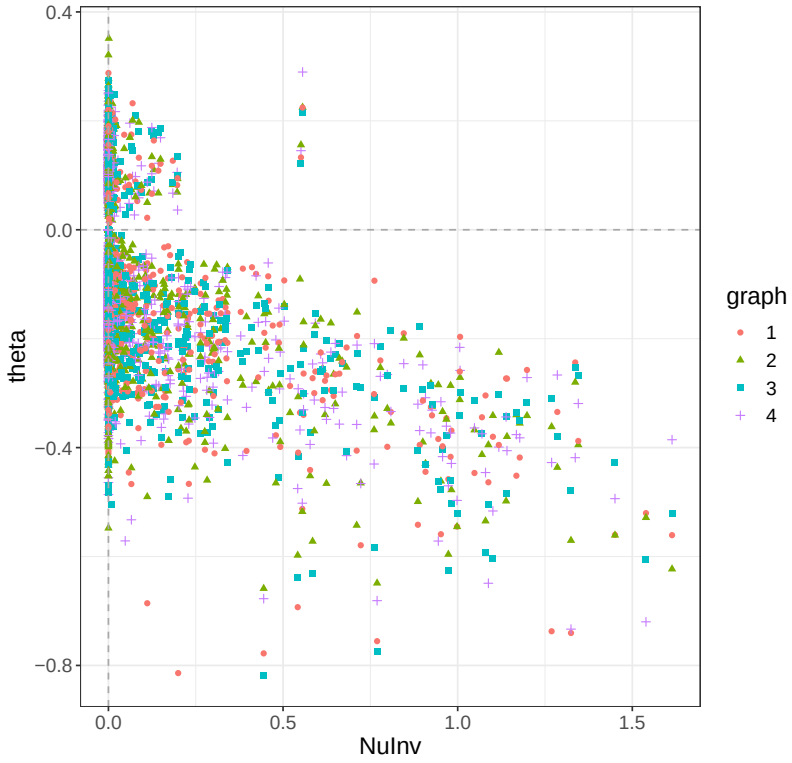

Figure A.1: Scaled precision matrix elements of the jointGHS graph plotted against the common latent parameters of all conditions.

Figure A.3 shows the density of the degree distribution of the jointGHS graph of each condition, as well as the degree of the *cis* genes *LYZ* and *YEATS4*. It appears that the degree distribution of the unstimulated network has more density on lower degrees, which is reasonable considering the unstimulated network is the sparsest among the four. As we see, the two *cis* genes have a relatively high degree in all conditions.

Top-hotspot mediated genes. Investigating whether the genes controlled by the top hotspot have more associations with each other can be insightful. In Table 3 in the main manuscript, we find the sparsity of the subnetwork of genes that are mediated by the top hotspot in each condition, and compares it to the sparsity of the whole network. We use permutation testing to assess whether the subnetwork of top hotspot controlled genes is denser than we would expect from a subnetwork of the same size; by randomly sampling 10000 subnetworks with the same size, we calculate empirical p-values for the observed sparsity. We see that the empirical p-value is 0.001 for all conditions except IFN-γ, implying the subnetworks are indeed denser in these conditions. While the difference is relatively small in the unstimulated network, it is larger in the LPS ones and in particular in LPS 24h.

Comparison to results from single network analysis. Table A.3 shows the sparsity of the estimated jointGHS graphs of the different conditions, as well as the percentage of the inferred edges that the jointGHS and fastGHS estimates at the same sparsity agree on. Figure A.4 shows an upset plot (Conway et al. [2017]), an alternative to a venn diagram that shows the number of edges shared between the inferred networks of each condition, for fastGHS and jointGHS separately. For each intersection, the number of edges shared only by the corresponding conditions is shown. It indicates that many edges are common to all four networks, meaning that the joint method has identified a fair amount of shared information. Compared to the fastGHS networks, we see that the intersection of edges present in all four conditions is larger in the jointGHS networks. This is as expected, as we, with the joint method, are better equipped to identify what is common and borrow information.

To understand more about what differs between the single and the joint estimates, we have a closer look at the genes controlled by the top hotspot rs6581889. The left panel of Figure A.5 compares the fastGHS and jointGHS density of the degree distribution of the subnetwork of genes controlled by the top hotspot in each condition. The density plots

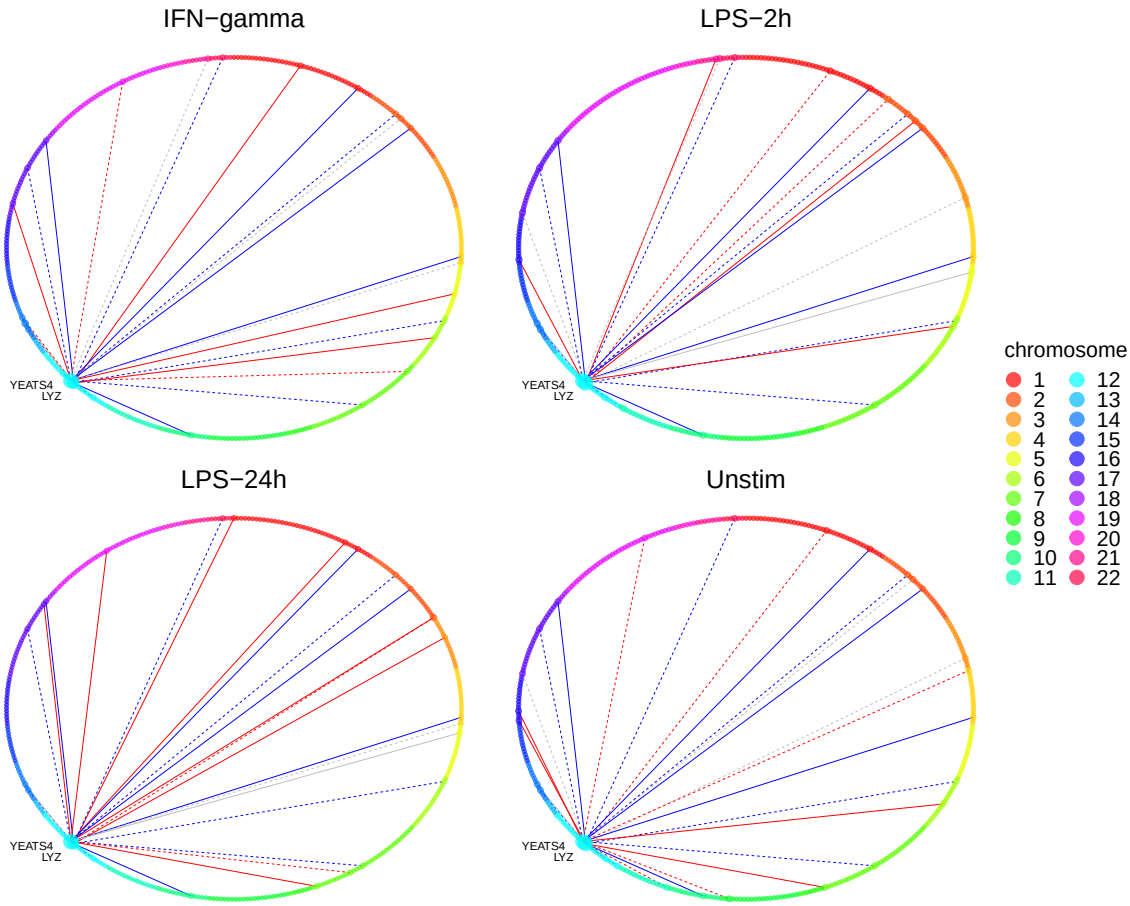

Figure A.2: Subnetworks consisting of the edges of the cis genes YEATS4 and LYZ in each condition of the monocyte data. A red edge indicates that the edge is unique to that condition, while a blue edge is common across all networks and a grey edge is present in two or three networks. Solid lines represent positive partial correlation while a dashed line represents negative partial correlation, and nodes are sized according to their edge degree in the subnetwork, with larger size representing a higher degree.

Table A.3 : The sparsity of the joint graphical horseshoe (jointGHS) estimated graphs of the different conditions in the monocyte data, as well as the percentage of the inferred edges that the jointGHS and the fastGHS estimate at the same sparsity agree on.

	IFN-gamma	LPS-2h	LPS-24h	Unstimulated
Sparsity	0.018	0.020	0.021	0.016
% Edge agreement with fastGHS	66.8	64.6	65.9	62.3

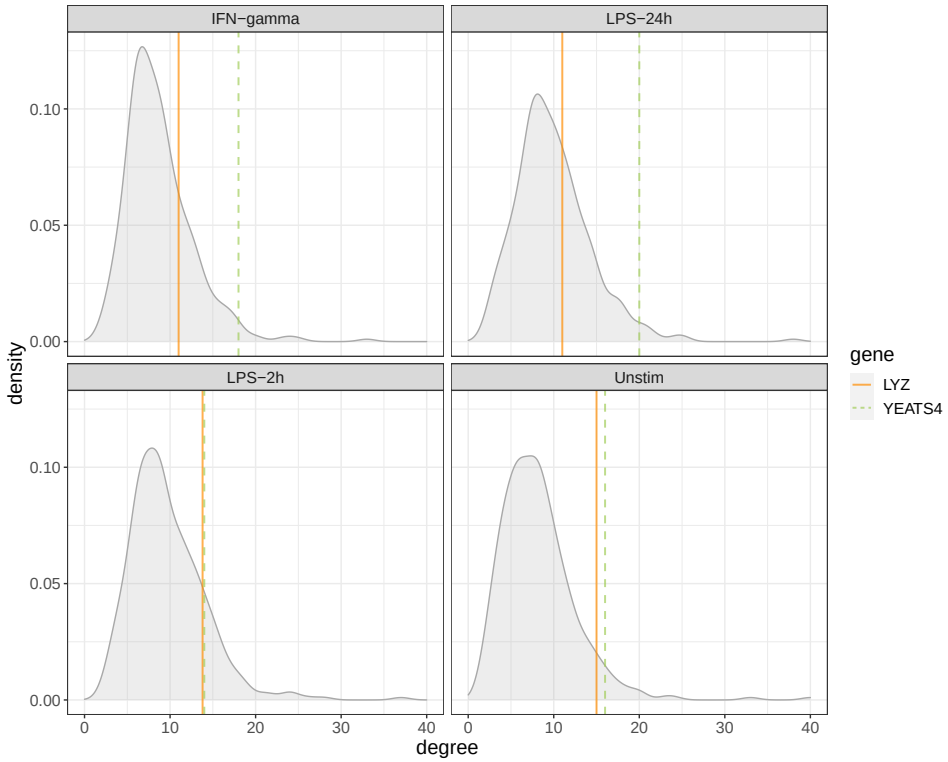

Figure A.3: Density plot of the degree distribution of the jointGHS graph of each condition, with the edge degree of LYZ and YEATS4 shown as vertical lines.

with the mean and median lines suggest that the jointGHS distribution is more shifted to the right in all conditions, implying that amongst the genes controlled by the top hotspot, the joint method identifies an overall higher activity level in terms of associations to other genes. This agrees with what we would expect, with the hotspot triggering substantial gene activity (Ruffieux et al. [2020]).

We next investigate the role of the two *cis* genes *LYZ* and *YEATS4* in the networks inferred by the two methods. In particular, we want to investigate if their role is common to all four conditions. To this end, we compare the neighbourhood status (with respect to the two *cis* genes) of the edges common to all conditions, namely, how close an edge is to a *cis* gene. If an edge is between a *cis* and a *trans* gene, that edge has order 1. If an edge is between two *trans* genes where at least one of them is a neighbour of a *cis* gene, than edge has order 2, and so on. Only the edges found to be common in all conditions in jointGHS but not in fastGHS are shown, and vice versa. The results from the comparison are shown in the right panel of Figure A.5. The edges identified by jointGHS but not fastGHS are in much closer proximity to the *cis* genes, implying that the joint method identifies a stronger effect from the *cis* genes, which agrees with previous findings (Ruffieux et al. [2020]). Similarly, there is not a single *cis-trans* edge identified by fastGHS but not jointGHS, while the opposite is not true. This again implies that the joint method is able to identify more of the *cis*-trans relationships.

We have now investigated what differs between the networks informed by fastGHS and jointGHS, but what about the edges the two methods agree on? The alluvial diagram in figure A.6 compares the order to the edges common to all conditions, and that fastGHS and jointGHS agree on, using the same definition of order as above. It shows that the two methods agree completely on all first order edges, meaning the direct effect of the *cis* genes on the *trans* genes in this case has sufficient evidence in the data to be captured even with the single-network approach. We also observe that the edges tend to have a lower order in the jointGHS estimates, implying a stronger influence from the *cis* genes has been captured. This strong effect of the *cis* genes observed in the jointGHS estimates agrees more with previous findings in this data set (Ruffieux et al. [2020]). In general, we see that two methods disagree more on the higher-order edges, where the effect of the *cis* genes can be said to be weaker. In such a setting, a joint approach that gains statistical power by sharing information between conditions can be highly useful, helping us gain more insight not only into the *cis-trans* effects but also role of the affected *trans* genes. In particular, many *trans* genes directly associated with the

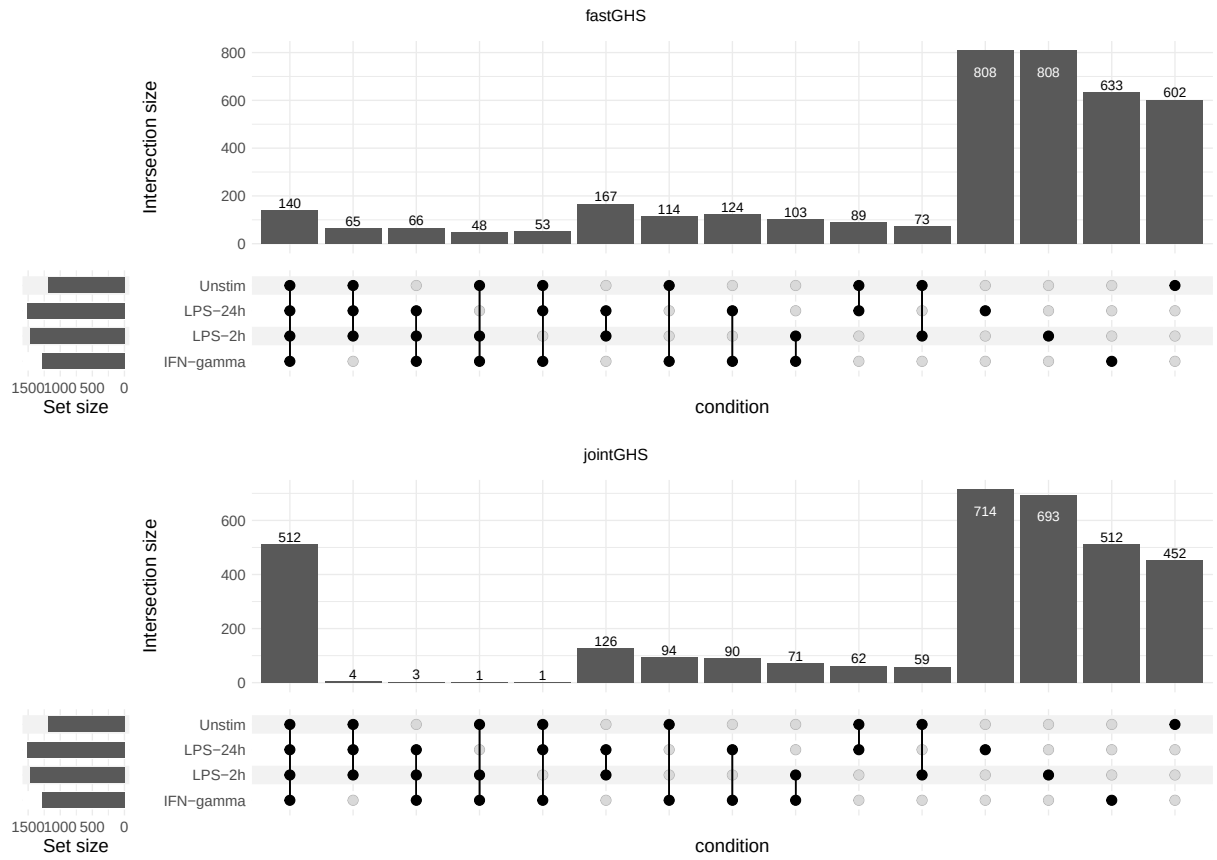

Figure A.4: Upset plot for the inferred edges of the fastGHS and jointGHS graphs of the monocyte data, an alternative to a venn diagram showing the number of shared edges between conditions. For each method, the total number of edges for each condition is represented on the left barplot. Every possible intersection is represented by the bottom plot, and their occurence is shown in the top barplot.

cis genes have a very high degree, highlighting their interplay with other *trans* genes as potentially relevant for the disease-driving mechanisms.

A.5 More on the choice of global shrinkage parameter

This section supplements Sections 2 and 4 in the main manuscript.

A.5.1 The deflation issue of the global shrinkage

Figure A.7 illustrates how the global shrinkage parameter shrinks to zero after just a few iterations, regardless of its initial value, when it is updated in the ECM algorithm. The Gaussian graphical data is generated using the procedure described in Section A.2, with $p = 50$ nodes and $n = 500$ observations in each data set.

A.5.2 The issue of under-selection

Our ECM approach tends to be conservative. Figure A.8 shows two trace plots for a Gaussian graphical data set with $p = 150$ nodes and $n = 500$ observations, one considering only small values of τ^2 and the other larger. Since we ultimately are interested in the partial correlations, we disregard the scale and look at the size of the scaled precision matrix elements $\theta_{ij}/\sqrt{\theta_{ii}\theta_{jj}}$. By comparing how these elements change as τ^2 increases, the plot illustrates that over-selection of edges is not a concern. This can in part be explained by the fact that the local scale parameters λ_{ij} adapt to the global shrinkage parameter, making the model flexible with respect to the choice of τ^2 as long as it is large enough to avoid overshrinkage to zero. Indeed, for small values of τ^2 we see that very few edges are captured, and

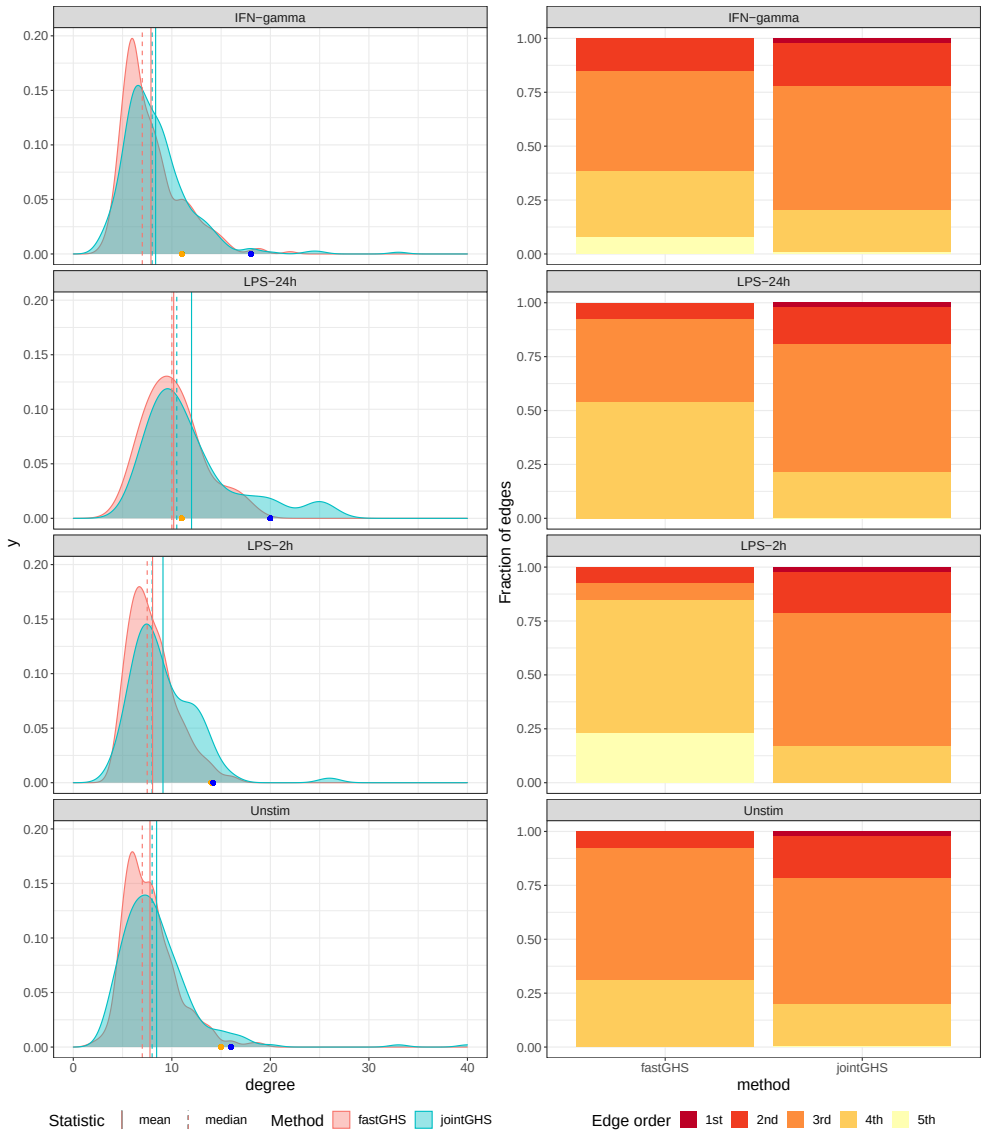

Figure A.5: Comparison of the results from fastGHS and jointGHS applied to the monocyte data. The left panel compares the density of the degree distribution of the subnetwork of genes controlled by the top hotspot in each condition, showing the mean and median as solid and dashed lines respectively. The degree of the cis genes LYZ and YEATS4 in the jointGHS networks are shown as orange and blue dots respectively. The right panel compares the neighbourhood status with respect to the cis genes of the edges common to all conditions. Only the edges found to be common in all conditions in jointGHS but not in fastGHS are shown, and vice versa.

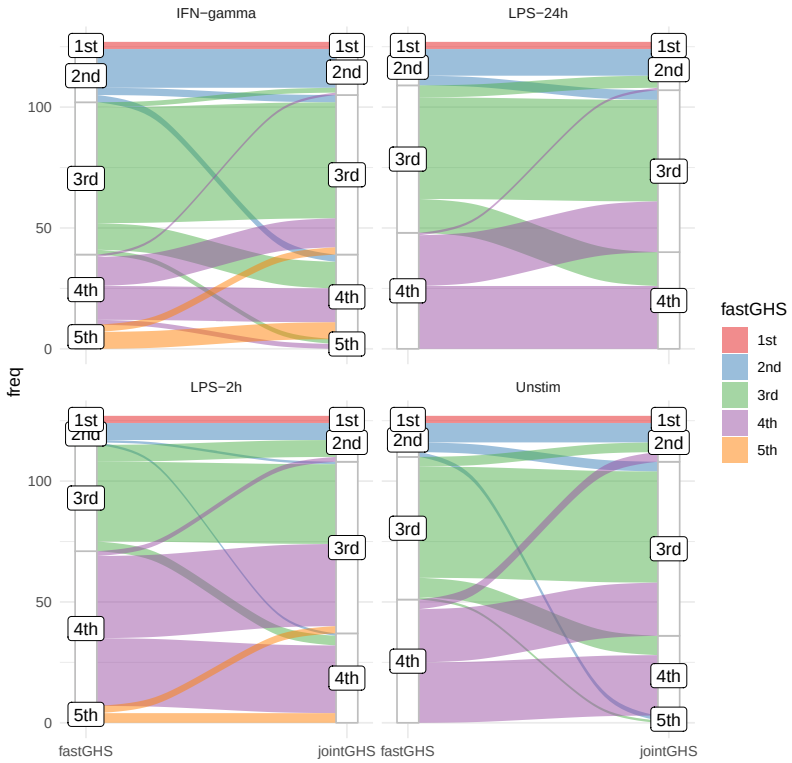

Figure A.6: Comparison of the order of the edges in the fastGHS estimated graphs to the jointGHS estimated graphs, in terms of distance to the cis genes.

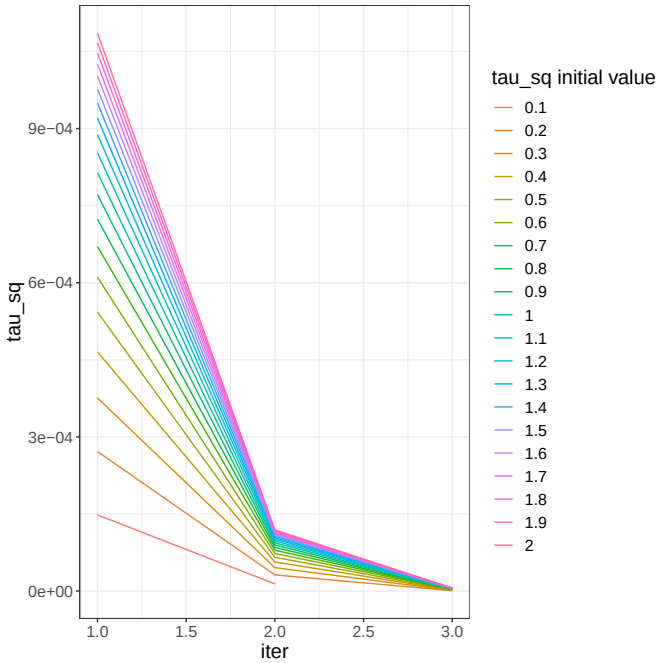

Figure A.7: The value of the global shrinkage parameter τ^2 at each iteration when it is not fixed, but updated as part of the ECM graphical horseshoe scheme. The results are shown for different initial values of τ^2 for a setting with $p = 50$ and $n = 500$.

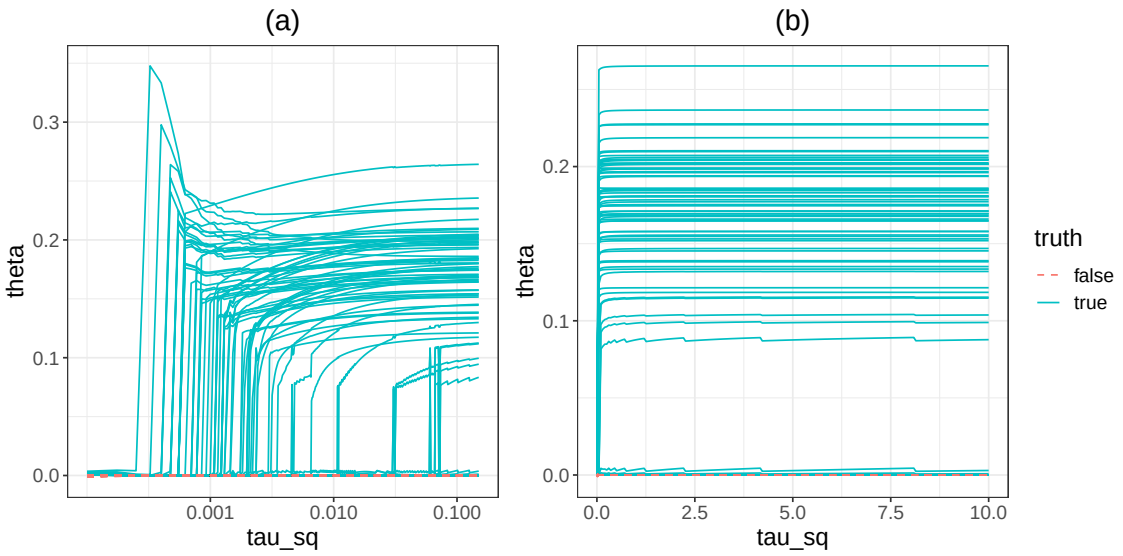

Figure A.8: Trace plots showing size of the scaled precision matrix elements estimated by the ECM graphical horseshoe change for (a) smaller τ^2 plotted on a logarithmic scale and (b) larger τ^2 on a normal scale, for a Gaussian graphical data set with $p = 150$ nodes and $n = 500$ observations. Each line represents one precision matrix element, and the lines are colored according to whether the corresponding edge is present in the true network or not. The non-zero scaled precision matrix elements are all equal to 0.2 in the simulated network.

the estimates deviate more from their true value of 0.2. It is clear that we should take care to ensure the global scale parameter is not too small. If high computational efficiency is desired, fixing it to a large value is possible. However, for general use we propose a data-driven way to select the parameter.

A.5.3 Avoiding under-selection with the AIC

Figure A.9 shows plots of the sparsity, precision and recall of ECM graphical horseshoe network estimates for different values of the fixed global shrinkage parameter $\tau^2 \in (0, 10]$, and for two data sets with (a) $p = 100$ and $n = 100$ with true graph sparsity 0.02 and (b) $p = 150$ and $n = 200$ with true graph sparsity 0.013. The estimated sparsity grows with τ^2, but only to a certain point. For large enough values of the global shrinkage parameter, the results are the same in terms of precision and recall. On the other hand, too small values of τ^2 can lead to severe under-selection of edges and very variable results.

Figure A.10 shows the second setting (b) for $\tau^2 \in (0, 2]$, where the values corresponding to the τ^2_{AIC} found by our AIC selection approach are marked as points. This rule leads to satisfactory results in terms of both precision and recall. A smaller value of τ^2 would result in reduced accuracy, and a larger value would not improve the estimate further.

A.6 Posterior checks with the Bayesian bootstrap

While the joint graphical horseshoe adapts well to the level of similarity between networks, issues may arise when the joint analysis is performed on a set with many highly similar networks and a few unrelated or less similar networks. In such a case, the highly similar networks might dominate the analysis through the common latent variables ν_{ij}. We have seen that, due to the heavy tail of the horseshoe, a small ν_{ij}^{-1} does not detoriate inference since the local scales still can escape and identify edges individually on the network level. However, a large ν_{ij}^{-1} tends to lead to non-zero precision matrix elements for all networks. A setting with many highly similar and a few unrelated or less similar network can result in large ν_{ij}^{-1} for the edges (i, j) common to the highly similar networks, risking wrongly included edges and thus reduced accuracy for the less similar ones.

To account for this possibility, after performing a joint network analysis it should be possible to check whether the edges of the joint network estimates strongly contradict the single network estimates. This could indicate that a joint approach is not suitable.

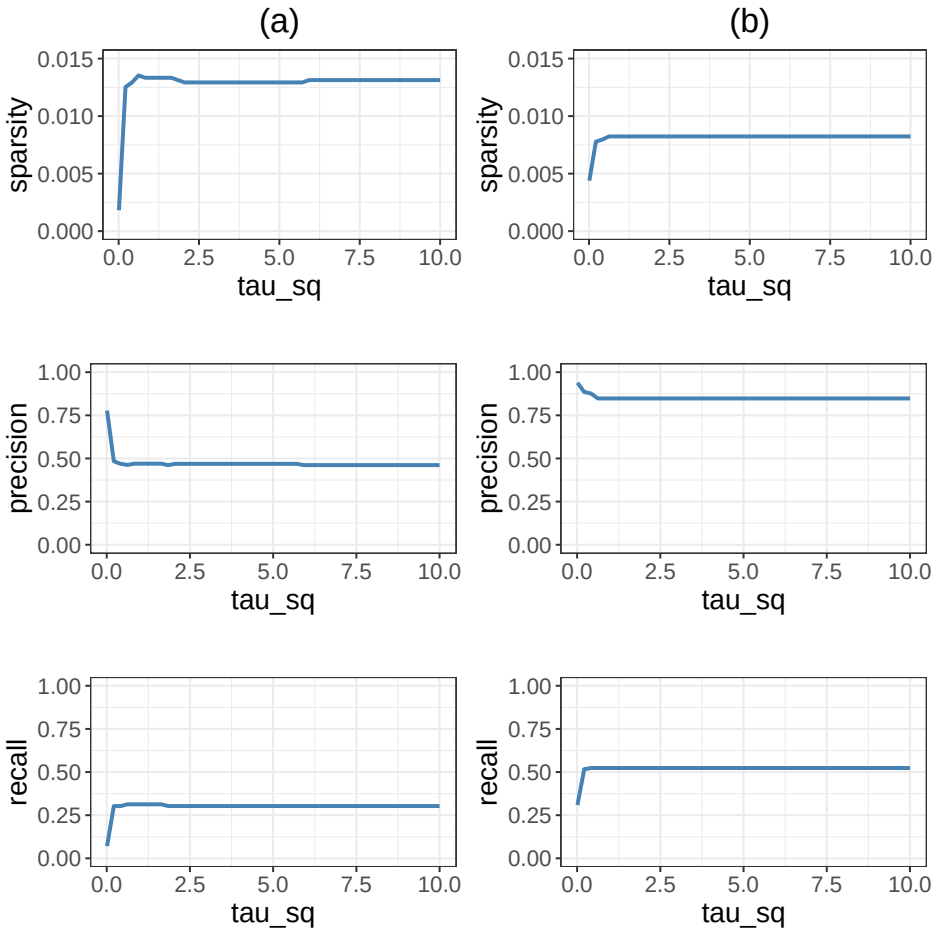

Figure A.9: Sparsity, precision and recall of ECM graphical horseshoe network estimates for different values of the fixed global shrinkage parameter τ^2 for data sets with (a) $p = 100$ and $n = 100$ with true graph sparsity 0.02 and (b) $p = 150$ and $n = 200$ with true graph sparsity 0.013.

For this purpose, we propose an approach for performing a posterior check for the suitability of a joint analysis. This check is implemented in our R package `jointGHS`. This extra step is more computationally demanding, but is optional and only necessary if there is doubt about the suitability of a joint approach.

A.6.1 Bayesian bootstrap procedure

We propose to use Bayesian bootstrapping (Rubin [1981]) to assess the suitability of a joint analysis. Specifically, for each network k and each inferred edge (i, j) found in the joint analysis, we assess whether the corresponding precision matrix element θ_{ijk} from the joint analysis is plausibly non-zero given its posterior Bayesian bootstrap distribution in the single network version.

In order to obtain a bootstrap sample ${\theta_{ijk}}^{(b)}$ for $b = 1, \ldots, B$, we start by drawing n_k weights from the Dirichlet distribution with n_k categories and parameters $\alpha = (1, \ldots, 1)$, i.e., the flat Dirichlet distribution. The resulting weight vector $\boldsymbol{w}^{(b)}$ is used to weigh the observations in the n_k by p observation matrix $\boldsymbol{X}_k$. For notational simplicity, we will from now refer to n_k as n, $\boldsymbol{X}_k$ as $\boldsymbol{X}$ and so on.

Using an expression for the Bayesian bootstrap sample of the covariance matrix (Rodriguez and Williams [2022]), we can obtain a weighted scatter matrix estimate for each bootstrap sample b, using the sampled weights $\boldsymbol{w}^{(b)}$

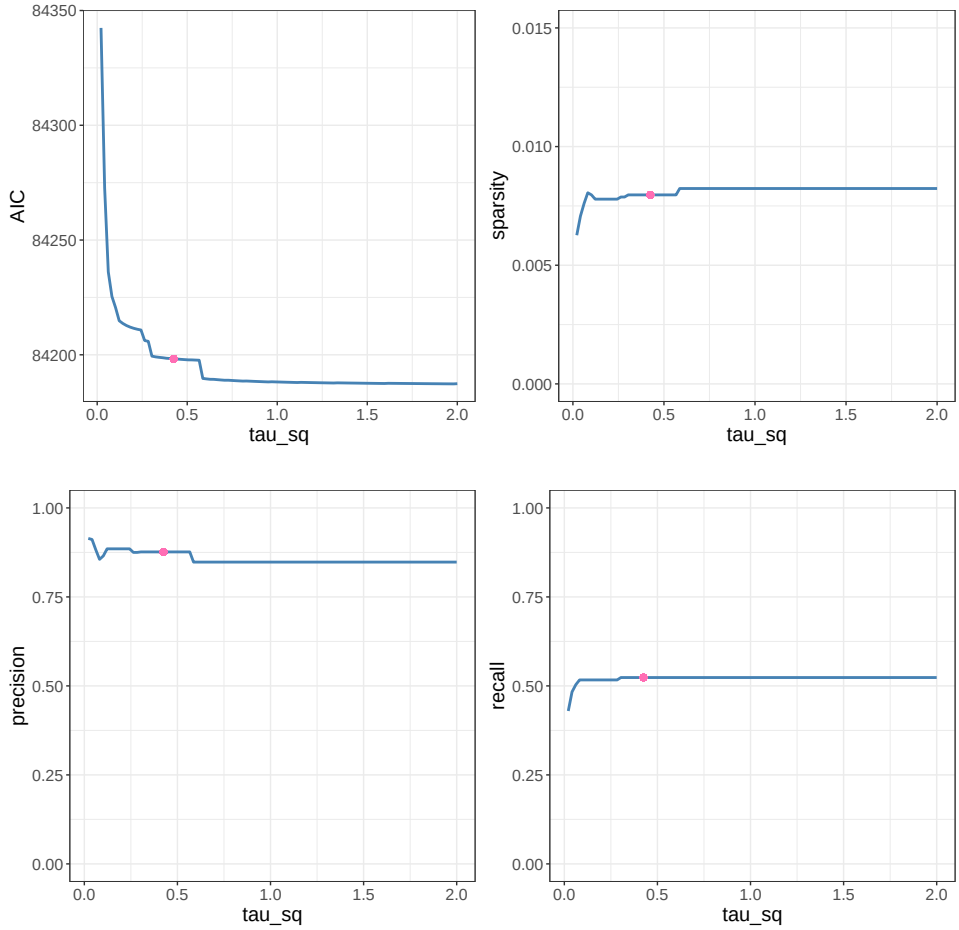

Figure A.10: Sparsity, precision and recall of ECM graphical horseshoe network estimates for different values of the fixed global shrinkage parameter τ^2 for a data set with $p = 150$ and $n = 200$ observations, and a true graph sparsity of 0.013. The values corresponding to the τ^2_{AIC} found by our AIC selection approach are marked as points.

$$\boldsymbol{S}_w^{(b)} = (n-1)\left[1 - \sum_{i=1}^{n} (w_i^{(b)})^2\right]^{-1} \boldsymbol{X}_w^{(b)^T} \boldsymbol{X}_w^{(b)}, \tag{16}$$

where $\boldsymbol{X}_w^{(b)}$ is the weighted observation matrix $\boldsymbol{X}_w^{(b)} = \boldsymbol{X} \circ \boldsymbol{w}^{*(b)} \mathbf{1}_p^T$. Here $\circ$ denotes the Hadamard product, $\boldsymbol{w}^{*(b)}$ is the length n vector with elements $\sqrt{w_i^{(b)}}$ and $\mathbf{1}_p$ is a length p vector of ones.

Replacing the unweighted scatter matrix $\boldsymbol{S}$ in the ECM graphical horseshoe algorithm by (16), we can obtain Bayesian bootstrap samples of θ_{ijk}. Repeating this B times, we get a set of Bayesian bootstrap samples $\{{\theta_{ijk}}^{(b)}\}_{b=1}^{B}$ that we can use to describe the posterior of the precision matrix elements in the single network model.

Using this sampled posterior, we can investigate whether the estimated non-zero θ_{ijk} from the joint network conflicts with its single network distribution. Since we ultimately are interested in the partial correlations, we perform comparisons regardless of scale by looking at the distribution of the scaled precision matrix elements $\theta_{ijk}/\sqrt{\theta_{iik}\theta_{jjk}}$. If a scaled precision matrix element exceeds the empirical 95^{th} percentile of its bootstrap distribution in absolute value, this suggests that it may have been overestimated and thus that the edge (i, j) may have been wrongly included in the joint model.

One can expect a few edges from the joint approach to be in conflict with their single network posterior as more information is available in the joint approach, but if this occurs for a large portion of edges this could suggest that the

network is being forced towards networks it bears little similarity to. This could indicate that a joint approach is not suitable.

In our R package `jointGHS` (`github.com/Camiling/jointGHS`), we provide a functionality for performing the Bayesian bootstrap as part of the joint graphical horseshoe, allowing assessment of all or specific edges. This also includes a plot and a print function. This optional functionality allows users to assess the suitability of a joint analysis. Notably, this tool is meant to guide users to make a decision, not to make the decision for them. For more details, we refer to the package documentation.

A.6.2 Examples

Two similar networks. The package has a plotting function which enables visualisation of the results of the Bayesian bootstrap. Figure A.11 shows the output for $K = 2$ data sets with the same underlying network structure. The Gaussian graphical data is generated using the procedure described in Section A.2, with $p = 50$ nodes and $n = 100$ observations in each data set. In this case we see that no joint graphical horseshoe estimate exceeds its Bayesian bootstrap empirical 95^{th} percentile in absolute value, implying that the joint estimates are not in conflict with the single network estimates.

Two different networks. On the contrary, Figure A.12 shows the output of the plot function for $K = 2$ data sets with completely unrelated network structures. The networks have $p = 50$ nodes and the data sets $n = 100$ and $n = 200$ observations respectively. Once again, we see that there is no conflict between the joint graphical horseshoe estimates and the single-network bootstrap distributions. While this might be surprising given that the two networks are completely unrelated, it is important to note that we have a "symmetrical" (dis)similarity pattern and so the common latent ν_{ij}^{-1} will not be large for edges present in only one network. For this reason, a Bayesian bootstrap check is not necessary for $K = 2$ networks.

Six networks with one unrelated. Figure A.13 shows the output of the plotting function for the last of $K = 6$ data sets, where all but the last data set have the same underlying network structure. All networks have $p = 20$ nodes and the data sets have $n = 150$ observations each. It is very clear from the plot that most of the scaled precision matrix elements of the inferred edges from the joint approach strongly contradict their single-network bootstrap distributions, exceeding their empirical 95^{th} percentiles in absolute value. Such an output indicates that the user should reconsider using a joint analysis.

For a summary of the Bayesian bootstrap procedure, the user can also use the print function implemented in the package. The function provides a summary of the findings, including how many edges whose corresponding scaled precision matrix element from the joint analysis exceeds its Bayesian bootstrap empirical 95^{th} percentile in absolute value. Figures A.14 and A.15 show the output of the print function for the 1^{st} and the 6^{th} data sets from the previous example respectively. This output indicates that 5.3% of the edges in network 1 exceeds this percentile, in contrast to 73.7% for network 6. When using the 95^{th} percentile, we can expect 5% of edges to exceed this threshold. Thus, the output suggests that the joint estimate strongly contradicts the single network estimate, which agrees with the simulated truth, namely that the 6^{th} data set is unrelated to the others.

A.7 Additional results

A.7.1 Scalability

Figure A.16 shows the CPU time used to infer a network for various numbers of nodes, for fastGHS.

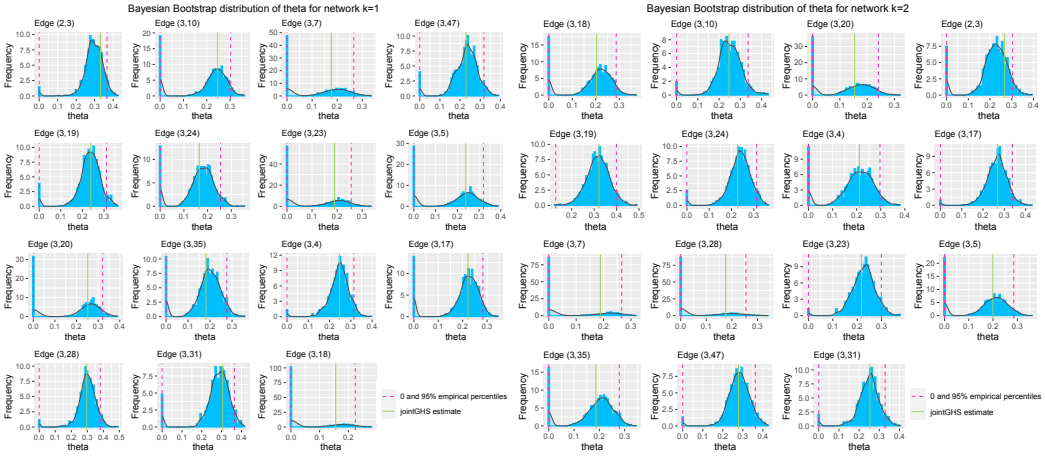

Figure A.11: Output of plot function implemented in the jointGHS package, showing results of the Bayesian bootstrap procedure for assessing the suitability of a joint approach for $K = 2$ networks with $p = 50$ nodes. The two data sets used for inference have the same true network structure, and $n = 100$ observations each. The output of the plot function is shown for both networks. For each inferred edge in the joint graphical horseshoe network of a given data set, the plot shows the Bayesian bootstrap distribution of the corresponding scaled precision matrix element of the single-network graphical horseshoe model, with the corresponding empirical 0% and 95% percentiles shown as dashed lines. The joint graphical horseshoe estimate is shown as a solid line.

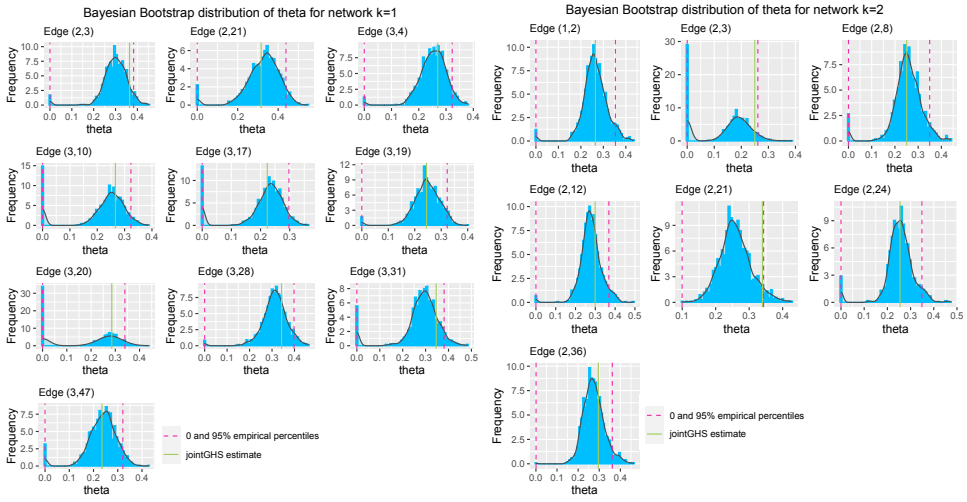

Figure A.12: Output of plot function implemented in the jointGHS package, showing results of the Bayesian bootstrap procedure for assessing the suitability of a joint approach for $K = 2$ networks with $p = 50$ nodes. The two data sets used for the inference have unrelated true network structures, and $n = 100$ and $n = 200$ observations respectively. The output of the plot function is shown for both networks. For each inferred edge in the joint graphical horseshoe network of a given data set, the plot shows the Bayesian bootstrap distribution of the corresponding scaled precision matrix element of the single-network graphical horseshoe model, with the corresponding empirical 0 and 95% percentiles shown as dashed lines. The joint graphical horseshoe estimate is shown as a solid line.

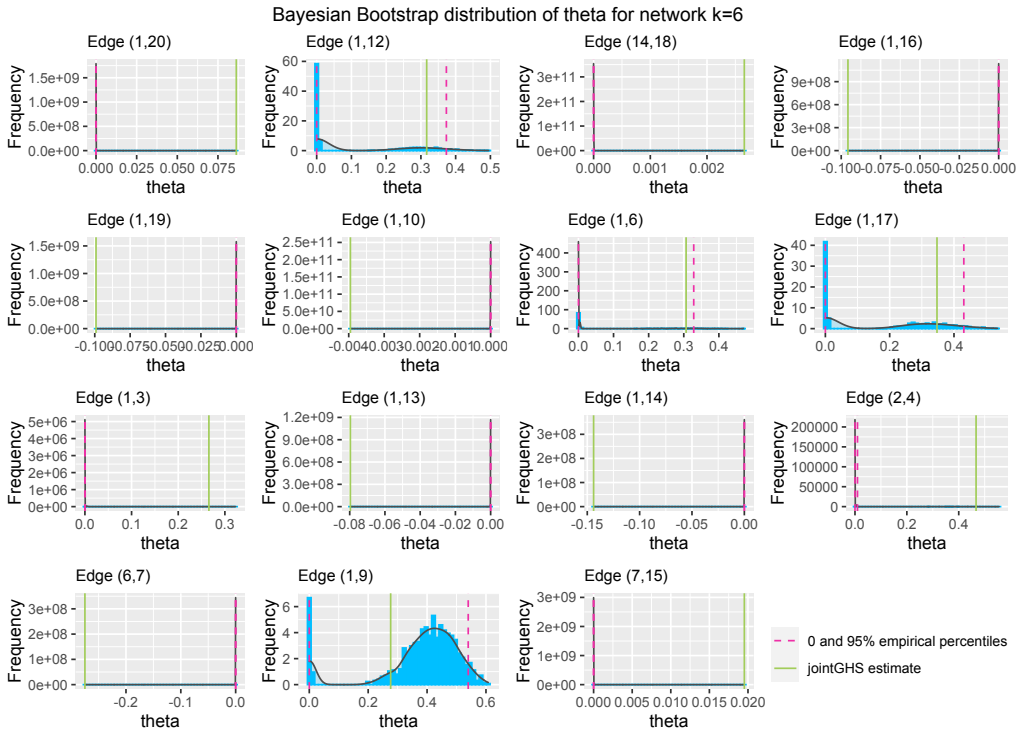

Figure A.13: Output of plot function implemented in the jointGHS package, showing results of the Bayesian bootstrap procedure for assessing the suitability of a joint approach for $K = 6$ networks with $p = 20$ nodes. The first five data sets used for the inference have the same true network structure, and the network structure of the sixth data set in question is completely unrelated to them. All data sets have $n = 150$ observations each. The output of the plot function is shown for data set 6. For each inferred edge in the joint graphical horseshoe network of a given data set, the plot shows the Bayesian bootstrap distribution of the corresponding scaled precision matrix element of the single-network graphical horseshoe model, with the corresponding empirical 0 and 95% percentiles shown as dashed lines. The joint graphical horseshoe estimate is shown as a solid line.

```
CHECK PARAMETERS OF IDENTIFIED EDGES IN NETWORK k=1
================================================================================
theta checked against its single-network Bayesian bootstrap distribution
================================================================================
   Edge                                jointGHS estimate    95% empirical percentile
1  (1,3)                                          0.3450                     0.3563
2  (1,5)                                         -0.2111                    -0.2357
3  (1,6)                                          0.2370                     0.2694
4  (1,8)                                          0.3422                     0.3991
5  (1,9)                                         -0.1180                    -0.1467
6  (1,10)                                        -0.1248                    -0.1425
7  (1,11)                                         0.2763                     0.2850
8  (1,12)                                         0.2633                     0.2765
9  (1,13)                                        -0.1441                    -0.1895
10 (1,14)                                         0.2349                     0.2865
11 (1,16)                                        -0.1567                    -0.2030
12 (1,17)                                         0.3208                     0.4001
13 (1,19)                                        -0.2696                    -0.3422
14 (1,20)                                         0.3317                     0.4196
15 (2,4)                                          0.8775                     0.8757 *
16 (4,5)                                         -0.3450                    -0.3983
17 (6,7)                                          0.5586                     0.6263
18 (7,15)                                         0.3064                     0.3932
19 (14,18)                                        0.5138                     0.5746
================================================================================
'*' Larger than 95% empirical percentile in absolute value

Edge parameters larger than 95% empirical percentile in absolute value: 1 (5.3%)
Expected number of edge parameters larger than  95 % empirical percentile in absolute value: 1  ( 5 %)
```

Figure A.14: Output of print function implemented in the jointGHS package, showing results of the Bayesian bootstrap procedure for assessing the suitability of a joint approach for $K = 6$ networks with $p = 20$ nodes. The first five data sets used for the inference have the same true network structure, and the network structure of the sixth data set in question is completely unrelated to them. All data sets have $n = 150$ observations each. The bootstrap results of the first data set is shown, and for each inferred edge in the joint graphical horseshoe network its scaled precision matrix estimate as well as the empirical 95% percentile of the corresponding single-network Bayesian bootstrap distribution is shown. If the estimate from the joint approach exceeds this percentile in absolute value, it is marked by a star.

```
CHECK PARAMETERS OF IDENTIFIED EDGES IN NETWORK k=6
================================================================================
theta checked against its single-network Bayesian bootstrap distribution
================================================================================
   Edge                                jointGHS estimate    95% empirical percentile
1  (1,3)                                          0.2657                     0.0003 *
2  (1,5)                                          0.1132                     0.0000 *
3  (1,6)                                          0.3067                     0.3284
4  (1,8)                                          0.2812                     0.0067 *
5  (1,9)                                          0.2763                     0.5397
6  (1,10)                                        -0.0040                     0.0000 *
7  (1,11)                                         0.3372                     0.4057
8  (1,12)                                         0.3166                     0.3733
9  (1,13)                                        -0.0796                     0.0000 *
10 (1,14)                                        -0.1439                     0.0000 *
11 (1,16)                                        -0.0957                     0.0000 *
12 (1,17)                                         0.3472                     0.4306
13 (1,19)                                        -0.0991                     0.0000 *
14 (1,20)                                         0.0863                     0.0000 *
15 (2,4)                                          0.4669                     0.0096 *
16 (4,5)                                          0.2052                     0.0000 *
17 (6,7)                                         -0.2749                     0.0000 *
18 (7,15)                                         0.0195                     0.0000 *
19 (14,18)                                        0.0027                     0.0000 *
================================================================================
'*' Larger than 95% empirical percentile in absolute value

Edge parameters larger than 95% empirical percentile in absolute value: 14 (73.7%)
Expected number of edge parameters larger than  95 % empirical percentile in absolute value: 1  ( 5 %)
```

Figure A.15: Output of print function implemented in the jointGHS package, showing results of the Bayesian bootstrap procedure for assessing the suitability of a joint approach for $K = 6$ networks with $p = 20$ nodes. The first five data sets used for the inference have the same true network structure, and the network structure of the sixth data set in question is completely unrelated to them. It is the bootstrap results of this data set that are printed. All data sets have $n = 150$ observations each. For each inferred edge in the joint graphical horseshoe network, its scaled precision matrix estimate as well as the empirical 95% percentile of the corresponding single-network Bayesian bootstrap distribution is shown. If the estimate from the joint approach exceeds this percentile in absolute value, it is marked by a star.

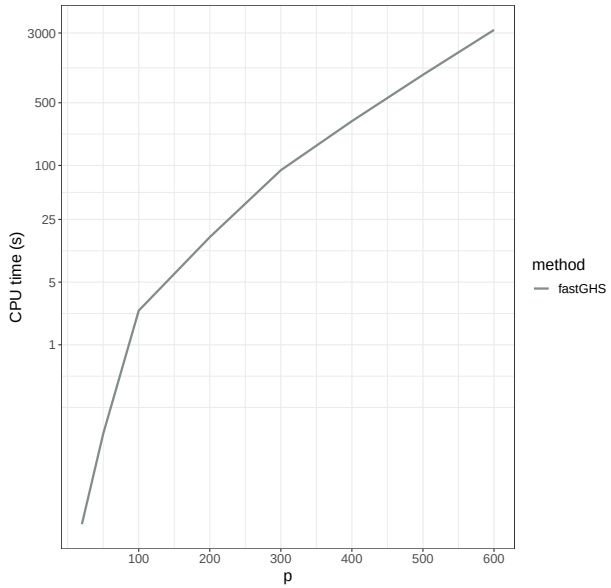

Figure A.16: CPU time in seconds on a logarithmic scale used to infer a network for various numbers of nodes p with $n = 500$ observations, for our fast ECM implementation of the graphical horseshoe (fastGHS).