The Annals of Applied Statistics
2024, Vol. 18, No. 3, 1899–1923
https://doi.org/10.1214/23-AOAS1863

SCALABLE MULTIPLE NETWORK INFERENCE WITH THE JOINT GRAPHICAL HORSESHOE

BY CAMILLA LINGJÆRDE[1,a], BENJAMIN P. FAIRFAX[2,d], SYLVIA RICHARDSON[1,b] AND HÉLÈNE RUFFIEUX[1,c]

[1]*MRC Biostatistics Unit, University of Cambridge,* [a]*camilla.lingjaerde@mrc-bsu.cam.ac.uk,* [b]*sylvia.richardson@mrc-bsu.cam.ac.uk,* [c]*helene.ruffieux@mrc-bsu.cam.ac.uk*

[2]*Department of Oncology, MRC Weatherall Institute for Molecular Medicine, University of Oxford,* [d]*benjamin.fairfax@oncology.ox.ac.uk*

Network models are useful tools for modelling complex associations. In statistical omics such models are increasingly popular for identifying and assessing functional relationships and pathways. If a Gaussian graphical model is assumed, conditional independence is determined by the nonzero 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 nonzero off-diagonal elements. However, its applicability is currently limited in statistical omics settings, which often involve high-dimensional data from multiple conditions that might share common structures. We propose: (i) a scalable expectation conditional maximisation (ECM) algorithm for the original graphical horseshoe and (ii) a novel joint graphical horseshoe estimator, which borrows information across multiple related networks to improve estimation. We show numerically 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. Finally, we leverage our approach to clarify gene regulation activity within and across immune stimulation conditions in monocytes, and formulate hypotheses on the pathogenesis of immune-mediated diseases.

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 nonzero entries of the inverse covariance (precision) matrix of the data. There is significant literature on this problem, both frequentist and Bayesian. Notable frequentist methods include the neighbourhood selection (Meinshausen and Bühlmann (2006)), the graphical lasso (Friedman, Hastie and Tibshirani (2008)) and the graphical SCAD (Fan, Feng and Wu (2009)). In later years Bayesian methods, such as the Bayesian graphical lasso (Wang (2012)), Bayesian spike-and-slab approaches (Wang

Received March 2023; revised October 2023.

Key words and phrases. Bayesian graphical models, cancer genomics, expectation conditional maximisation, gene networks, genomics, graphical horseshoe, horseshoe prior, high-dimensional inference, integrative analysis, multiomics, network models.

(2015)) and the graphical horseshoe (Li, Craig and Bhadra (2019)), have gained popularity. The Bayesian model formulation of the graphical horseshoe leads to many desirable properties, particularly for identifying weak edges. Indeed, the global-local horseshoe prior it relies on permits the introduction of edge-specific parameters that prevent overshrinkage of nonzero off-diagonal elements, resulting in a highly flexible framework. However, in the high-dimensional settings commonly needed for investigating biological networks, the Gibbs sampling implementation, proposed by Li, Craig and Bhadra (2019), 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. In biomedical applications 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 the field of multiple Gaussian graphical models, notable frequentist methods include variations of the joint graphical lasso (Danaher, Wang and Witten (2014), Lingjærde and Richardson (2023)), which enforce similar graphical structures by solving a penalised likelihood problem, and a group extension of the graphical lasso to multiple networks (Guo et al. (2011)). In a Bayesian framework, Peterson, Stingo and Vannucci (2015) propose a Gibbs sampling approach which uses a Markov random field prior on multiple graphs to learn the similarity of graphical structures. Very recently, Busatto and Stingo (2023) have proposed an extension of the graphical horseshoe which does not build upon our method, but models pairwise network similarity. Implemented with a Metropolis-within-Gibbs sampler, it is fully Bayesian and hence less scalable. Other notable Bayesian approaches include two expectation-maximisation (EM) approaches, namely, the Bayesian spike-and-slab joint graphical lasso (Li, Mccormick and Clark (2019)), which builds on seminal work by Wang (2015) and formulates a multiple-network model based on a Gaussian spike-and-slab prior, and GemBag (Yang et al. (2021)), which relies on a spike-and-slab prior with Laplace distributions. For a more thorough review on existing Bayesian graphical methods, we refer to Ni et al. (2022).

The method presented in this paper is, to our knowledge, the first to adapt graphical modelling based on global-local priors to the multiple network setting. We first propose an expectation conditional maximisation (ECM) algorithm for the graphical horseshoe, which allows us to tackle network inference in biological problems of realistic sizes. Building on this efficient implementation, we then formulate a joint model that permits borrowing information between multiple networks using the graphical horseshoe prior. We provide the two R packages `fastGHS` and `jointGHS`, available on Github and in Supplementary Material D (Lingjærde et al. (2024a)), which implement the single and joint methods, respectively.

The paper is organised as follows. In Section 2 we recall the classical graphical horseshoe estimator of Li, Craig and Bhadra (2019) and discuss the advantages of the horseshoe prior in graphical settings. In Section 3 we motivate the need for further development to enable practical and meaningful inference at scale by introducing a multicondition gene regulation study in monocytes. In Section 4 we describe our new inference procedure for the single-network graphical horseshoe, and in Section 5 we present our joint graphical horseshoe model formulation. 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. 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 $\boldsymbol{x_1}, \ldots, \boldsymbol{x_n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, 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 (Lauritzen (1996)). For Gaussian variables, correlation equal to zero is equivalent to independence, which implies that a conditional independence graph can be constructed by determining the nonzero entries of the precision matrix. The graph is assumed to be sparse; that is, the number of edges in the edge set E relative to the number of potential edges in the graph, $2|E|/(p^2 - p)$, is small. The precision matrix must also necessarily be positive definite, $\boldsymbol{\Theta} \succ 0$.

Li, Craig and Bhadra (2019) have recently proposed the *graphical horseshoe* to obtain a sparse estimate of the precision matrix $\boldsymbol{\Theta}$, repurposing the horseshoe prior initially introduced by Carvalho, Polson and Scott (2010) in the normal means setting 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:

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

with $\boldsymbol{\Theta} \succ 0$ and where $\mathrm{C}^+(0, 1)$ is the half-Cauchy distribution with density $p(x) \propto (1 + x^2)^{-1}$, $x > 0$. A key feature of the horseshoe prior in (1) is the presence of local shrinkage parameters, λ_{ij}, which are meant to flexibly capture edge-specific effects with no or very limited overshrinkage, while the global parameter τ is set to ensure overall sparsity. In the next section, we motivate the extension of the graphical horseshoe to a multiple network setting, with a scalable implementation.

3. Data and motivating example. A wide array of prevalent diseases, such as inflammatory bowel disease, rheumatoid arthritis and cancer, are believed to result from an inappropriate immune activity and consequent inflammation. It is now established that exposing monocyte cell cultures to specific stimuli can create conditions that resemble certain immune-mediated disease states (Biswas and Mantovani (2010)). Indeed, different stimuli may activate different cellular or molecular pathways that contribute to disease development. Hence, systematically investigating the impact of different types of immune stimulation on gene regulatory activity—and pinpointing the mechanisms that are common to several stimuli or stimulus specific—can provide valuable insights into the pathophysiology of such diseases.

We propose to contribute to this research by analysing graphical structures from a detailed gene expression dataset where primary $\mathrm{CD14}^+$ monocytes in 432 healthy European individuals were exposed to different types of immune stimulation (hereafter "conditions"). Specifically, monocyte expression was quantified using Illumina HumanHT-12 v4 BeadChip arrays 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). The

number of samples available in each condition is $n_{\text{unstim}} = 413$, $n_{\text{IFN-}\gamma} = 366$, $n_{\text{LPS2h}} = 260$, $n_{\text{LPS24h}} = 321$ for unstimulated cells, and IFN-γ-, LPS 2h- and LPS 24h-stimulated cells, respectively (Fairfax et al. (2014)).

Examining the effect of genetic variation on gene regulatory activity after stimulation can help pinpointing gene sets implicated in disease risk and development. Indeed, previous studies suggest that gene stimulation triggers regulatory activity that leads to a beneficial environment for *hotspots* to establish (Fairfax et al. (2014), Lee et al. (2014), Kim et al. (2014), Ruffieux et al. (2020)); hotspots are genetic variants regulating large numbers of genes, thereby potentially representing important players in disease mechanisms (Yao et al. (2017)). We will, therefore, focus our study to networks of genes under hotspot control in the different conditions.

While Fairfax et al. (2014) mainly report condition-specific gene 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, they found that the genetic variant rs6581889 was 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 it controls, namely, *LYZ* and *YEATS4*, which are thought to play a central role in the pathogenesis of immune disorders (Fairfax et al. (2012)).

Given the shared hotspot control in all four immune stimulation conditions, borrowing information with a joint network approach on the controlled genes across these conditions seems particularly appropriate to further investigate the gene regulation mechanisms triggered by rs6581889. At the same time, Ruffieux et al. (2020) found that there is only a partial overlap between the genes associated with the hotspot across the different conditions, which calls for a modelling approach that can also effectively detect stimulus-specific effects—such effects being highly relevant to understand how distinct pathways may be activated in different disease states. Such an analysis should help characterising the complex interplays among the genes controlled, that is, the direct effects on the distal genes or the indirect effects, mediated via other genes controlled by the hotspot (typically via proximal genes, such as *LYZ* and *YEATS4*). Although graphical modelling approaches seem particularly appropriate to disentangle direct and mediated effects, they have not been employed thus far.

With its desirable theoretical properties as well as its high performance in numerical studies, the graphical horseshoe estimator (Li, Craig and Bhadra (2019)) would be a natural choice for graph inference. However, its applicability in the above monocyte setting is hampered by two limitations: first, its Gibbs sampler implementation does not scale to the problem dimensions ($p = 381$ genes for $n \leq 413$ observations in each condition). Second, its model is formulated for the analysis of a single network, meaning that it can only be applied separately to each of the four conditions. Although relevant for identifying common structures across the conditions, the Bayesian spike-and-slab joint graphical lasso (Li, Mccormick and Clark (2019)) and the joint graphical lasso (Danaher, Wang and Witten (2014)) do not enjoy the flexibility granted by the horseshoe prior's local scales for identifying network-specific effects (whose detection is key to disentangle disease-specific mechanisms, as explained above). Moreover, these methods did not reach convergence within 48 hours, as jointly modelling all four conditions requires inferring a total of 289,560 edges. This severe high dimensionality makes it essential to develop approaches that are specifically designed to scale to the current statistical omics problem sizes.

Motivated by the monocyte problem, this work is concerned with proposing a new framework that addresses the two shortcomings outlined above to enable effective network inference in realistic practical settings. Namely, we aim to: (i) develop an expectation conditional

maximisation (ECM) algorithm for the graphical horseshoe as a fast yet accurate alternative to the Gibbs sampling procedure proposed by Li, Craig and Bhadra (2019) and (ii) formulate a joint model for multiple networks, leveraging the global-local horseshoe feature to borrow strength across shared patterns *while* preserving differences across networks.

Equipped with this framework, we will return to the monocyte problem in Section 7 to demonstrate the computational feasibility of joint network modelling on these data and exploit the advantages of the global-local formulation for inferring and interpreting condition-specific and shared gene regulation structures.

4. An ECM algorithm for estimating the graphical horseshoe. As a first step in detailing our proposal, we outline the expectation conditional maximisation (ECM) procedure for estimating the joint posterior of the precision matrix and its local scale parameters, which adapts the spike-and-slab EM approach of Ročková and George (2014) to the graphical horseshoe setting. Similar to the latter approach, we propose to identify the joint posterior of $(\boldsymbol{\Theta}, \boldsymbol{\Lambda})$ instead of $\boldsymbol{\Theta}$ marginally, to ensure analytical tractability of the objective function and computational effectiveness. Note that this is in contrast to the usual application of ECM, where one would integrate out the local scale parameters. We first detail the updates for the single-network graphical horseshoe prior (1) based on a classical inverse Gamma prior reparametrisation of the horseshoe local scales, which guarantees parameter conjugacy. The ECM approach, described first by Meng and Rubin (1993), is a generalised EM algorithm (Dempster, Laird and Rubin (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, Craig and Bhadra (2019), the full conditional posteriors of the local λ_{ij}'s can be derived by introducing the augmented variables ν_{ij}. We next employ the following reparameterisation, introducing the latent ν_{ij} and writing

$$\begin{aligned}\lambda_{ij}^2|\nu_{ij} &\sim \text{InvGamma}(1/2, 1/\nu_{ij}), \quad 1 \le i < j \le p,\\ \nu_{ij} &\sim \text{InvGamma}(1/2, 1).\end{aligned}$$

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

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

where $\cdot$ denotes all other variables. The latent variables can be collected in the latent matrix $\boldsymbol{N} = (\nu_{ij})$. The global shrinkage parameter τ in (1) is for now treated as a known fixed hyperparameter; its specification will be detailed in Section 5.3. 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 (2012), we write

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

$$\boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{(-p)(-p)} & \boldsymbol{\lambda}_{(-p)p} \\ \boldsymbol{\lambda}_{(-p)p}^T & 1 \end{pmatrix},$$

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

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

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

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

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 columnwise.

4.2. *ECM algorithm.* Given the estimates from the previous iteration l, the objective function is obtained as

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

where $\text{E}_{|\cdot|}(\cdot)$ denotes $\text{E}_{\boldsymbol{N}|\boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)}, \boldsymbol{S}}(\cdot)$ and const. is a constant not depending on $\boldsymbol{\Theta}$ or $\boldsymbol{\Lambda}$. Note that the objective function accounts for the full priors of $\boldsymbol{\Theta}$ and $\boldsymbol{\Lambda}$.

In the E-step of the algorithm, the conditional expectations in (4) are computed over the latent parameter $\boldsymbol{N}$, while the CM-step performs the maximisation with respect to $(\boldsymbol{\Theta}, \boldsymbol{\Lambda})$. Similar to the Bayesian spike-and-slab joint graphical lasso of Li, Mccormick and Clark (2019) where the objective function is maximised over both the precision matrix and sparsity parameters, this approach finds a posterior mode of $(\boldsymbol{\Theta}, \boldsymbol{\Lambda})$, accounting for prior distributions on all other parameters.

From (2) the full conditional distributions of the ν_{ij}'s are inverse Gamma. Therefore, the E-step updates are

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

where $\psi(\cdot)$ is the digamma function.

Next, the CM-step maximises (4) with respect to $(\boldsymbol{\Theta}, \boldsymbol{\Lambda})$ in a coordinate ascent fashion, with the expectations replaced with the expressions found in (5). The following closed-form updates are obtained for the λ_{ij}^2's:

$$\lambda_{ij}^{2(l+1)} = \frac{\lambda_{ij}^{*(l)} + \theta_{ij}^2/(2\tau^2)}{2}. \tag{6}$$

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

$$
\begin{aligned}
\theta_{pp}^{(l+1)} &= \boldsymbol{\theta}_{(-p)p}^{(l+1)^T} \big(\boldsymbol{\Theta}_{(-p)(-p)}^{(l+1)}\big)^{-1} \boldsymbol{\theta}_{(-p)p}^{(l+1)} + \frac{n}{s_{pp}}, \\
\boldsymbol{\theta}_{(-p)p}^{(l+1)} &= -\left\{ s_{pp} \big(\boldsymbol{\Theta}_{(-p)(-p)}^{(l+1)}\big)^{-1} + \frac{1}{\tau^2} \big(\boldsymbol{\Lambda}^{*^{(l+1)}}\big)^{-1} \right\}^{-1} \boldsymbol{s}_{(-p)p},
\end{aligned}
\tag{7}
$$

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 columnwise. With these updates the positive definiteness constraint for $\boldsymbol{\Theta}$ is maintained at each iteration as long as the initial value is positive definite. This can be shown with an argument equivalent to that of Wang (2012): assume that the update $\boldsymbol{\Theta}^{(l)}$ is positive definite, then all its p leading principal minors are positive. After updating the last row and column as in (7), the updated precision matrix $\boldsymbol{\Theta}^{(l+1)}$ has the same leading principal minors as in $\boldsymbol{\Theta}^{(l)}$, except for the last one, which is of order p. The last leading principal minor is clearly equal to $\det(\boldsymbol{\Theta}^{(l+1)}) = \gamma \det(\boldsymbol{\Theta}_{(-p)(-p)}^{(l)})$, where $\det(\boldsymbol{\Theta}_{(-p)(-p)}^{(l)})$ is the $(p-1)$th leading principal minor of $\boldsymbol{\Theta}^{(l)}$ and thus positive, and we have, from (3), that $\gamma = \theta_{pp} - \boldsymbol{\theta}_{(-p)p}^T \boldsymbol{\Theta}_{(-p)(-p)}^{-1} \boldsymbol{\theta}_{(-p)p} > 0$. Consequently, $\det(\boldsymbol{\Theta}^{(l+1)}) > 0$, and so the updated $\boldsymbol{\Theta}^{(l+1)}$ is positive definite.

With this CM-step update, it is ensured that $Q(\boldsymbol{\Theta}^{(l+1)}, \boldsymbol{\Lambda}^{(l+1)} | \boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)}) \geq Q(\boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)} | \boldsymbol{\Theta}^{(l)}, \boldsymbol{\Lambda}^{(l)})$ (Meng and Rubin (1993)). By iterating between the E-step and the CM-step until convergence, we obtain an estimator of a posterior mode of $(\boldsymbol{\Theta}, \boldsymbol{\Lambda})$. It is important to note that we are finding posterior modes of $p(\boldsymbol{\Theta}, \boldsymbol{\Lambda} | \boldsymbol{S})$ rather than $p(\boldsymbol{\Theta} | \boldsymbol{S})$. As previously discussed, this enables closed-form updates and hence substantial computational savings; our simulation studies in Section 6 indicate that this approach still permits identifying high posterior regions of $p(\boldsymbol{\Theta} | \boldsymbol{S})$, with our algorithm achieving a very good recovery of the precision matrix. The full derivations for this section are given in Section S.1 of Supplementary Material A (Lingjærde et al. (2024b)).

One of the main computational advantages of the ECM approach over stochastic search is that a posterior mode is fast to obtain. The estimates are computed directly, and a full stochastic search is not necessary. Further, while the updates for the local scales never become exactly zero, as they are half Cauchy, they can be very small in the absence of an edge. As a result, the precision matrix entries corresponding to unidentified edges tend to converge to values close to zero whose inclusion is not meaningful, 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)). To ensure numerical stability and avoid under- and overflow in the implementation of the algorithm, the updates are reformulated using the exponential and logarithmic functions to avoid direct divisions. We hereafter refer to this ECM implementation as "fastGHS."

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, and $n_k \times p$ observation matrices $\boldsymbol{X}_k$ for $k = 1, \ldots, K$, we are interested in estimating the precision matrices $\{\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_K\}$. We let the kth precision matrix follow the hierarchical model

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

with $\boldsymbol{\Theta_k} \succ 0$, $k = 1, \ldots, K$. This is the standard graphical horseshoe model for each network separately. To share information across networks, we introduce the latent variables ν_{ij} and write

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

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

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

similarly to the standard graphical horseshoe. Hence, information is now shared across networks through the common latent variable ν_{ij}, and the full conditional posterior of the ν_{ij}'s now depends on the λ_{ijk}'s of all K networks,

$$p(\nu_{ij} | \cdot) \propto \mathrm{InvGamma}\left(\frac{K+1}{2}, 1 + \sum_{k=1}^{K} \frac{1}{\lambda_{ijk}^2}\right). \tag{8}$$

The derivation of (8) is given in Section S.1.4 of Supplementary Material A (Lingjærde et al. (2024b)).

The global scales τ_k are network-specific to allow for different sparsity levels across networks; their specification is detailed in Section 5.3. Alternative approaches that directly model structured and shared sparsity, for example, using a Markov random field prior as in the spike-and-slab graphical approach of Peterson, Stingo and Vannucci (2015), could also be considered. In practice, however, the use of local scales results in considerable flexibility in adapting to the overall sparsity levels of the different networks, as our numerical experiments from Sections 6 and 7 below suggest.

5.2. *ECM approach.* The E-step and CM-step of the multiple-network ECM algorithm are 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 distribution of ν_{ij} now depends on the local shrinkage parameters of all K networks.

Using the full conditional distribution (8), the E-step updates are

$$\begin{aligned} \mathrm{E}_{|\cdot}\{\log(\nu_{ij})\} &= \log\left(1 + \sum_{k=1}^{K} \frac{1}{\lambda_{ijk}^{2(l)}}\right) - \psi\left(\frac{K+1}{2}\right), \\ \mathrm{E}_{|\cdot}\left(\frac{1}{\nu_{ij}}\right) &= \frac{K}{2(1 + \sum_{k=1}^{K} 1/\lambda_{ijk}^{2(l)})} =: \lambda_{ij\cdot}^{*(l)}. \end{aligned} \tag{9}$$

The CM-step updates for the λ_{ijk}^2's are obtained by replacing the expectation $\mathrm{E}_{|\cdot}(\nu_{ij}^{-1}) = \lambda_{ij}^{*(l)}$ in (6) by the $\lambda_{ij\cdot}^{*(l)}$ update of the E-step in (9),

$$\lambda_{ijk}^{2(l+1)} = \frac{\lambda_{ij\cdot}^{*(l)} + \theta_{ijk}^2/(2\tau_k^2)}{2}. \tag{10}$$

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, Craig and Bhadra (2019)) given by

$$\theta_{ppk}^{(l+1)} = \boldsymbol{\theta}_{(-p)pk}^{(l+1)^T} \big(\boldsymbol{\Theta}_{(-p)(-p)k}^{(l+1)}\big)^{-1} \boldsymbol{\theta}_{(-p)pk}^{(l+1)} + \frac{n_k}{s_{ppk}},$$

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

where the matrix partitioning is analogous to (7). 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.* The specification of the horseshoe global scale parameter τ (or τ_k in the multiple network case) has been a subject of active debate over the past years (see, e.g., Carvalho, Polson and Scott (2009), Carvalho, Polson and Scott (2010), Piironen and Vehtari (2017)). The different proposals may be grouped into three strategies: (i) use a prior on τ, typically a half-Cauchy prior, (ii) fix it or (iii) use a selection criterion. Previous work has found that strategy (i) can result in degenerate solutions when using deterministic inference algorithms (such as our ECM algorithm) or other empirical Bayes procedures, in very sparse settings (Scott and Berger (2010), Polson and Scott (2010), Bhadra et al. (2019), van de Wiel, Te Beest and Münch (2019)). As Li, Craig and Bhadra (2019) indicate, strategy (ii) can be employed to control the sparsity level of the graphical horseshoe estimates and avoid overshrinkage. This is a common approach in nongraphical horseshoe settings. For instance, van der Pas, Kleijn and van der Vaart (2014) fix τ based on theoretical justification, namely, τ should be of the order of the proportion of nonnull effects to guarantee asymptotic minimaxity, an argument that Bhadra et al. (2017) also follow in the context of the horseshoe+ estimator. Piironen and Vehtari (2017) instead proposed to make assumptions on the "effective model size;" their approach is, however, not transferable to our graphical setting due to the iterative nature of our updates (7). We instead implement a procedure based on strategy (iii), as detailed hereafter.

Assuming the multiple network setting, we propose to select each τ_k^2 separately for each network using the AIC criterion for Gaussian graphical models (Akaike (1973)), before running the joint analysis. As demonstrated in Section S.5 of Supplementary Material A (Lingjærde et al. (2024b)), fastGHS and jointGHS typically do not overselect edges, and using more stricter criteria, such as the BIC, would result in severe underselection of edges. For a given global shrinkage parameter τ_k^2 and corresponding precision matrix estimate $\widehat{\boldsymbol{\Theta}}_{k,\tau_k^2}$ found with fastGHS, the AIC score is given by

$$\mathrm{AIC}(\tau_k^2) = \frac{n_k}{n_k - 1} \operatorname{tr}(\boldsymbol{S}_k \widehat{\boldsymbol{\Theta}}_{k,\tau_k^2}) - n_k \log\{\det(\widehat{\boldsymbol{\Theta}}_{k,\tau_k^2})\} + 2|E_{\tau_k^2}|,$$

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

For small τ_k^2, small increases lead to large changes in the AIC score (see Section S.5 of Supplementary Material A (Lingjærde et al. (2024b))). However, for sufficiently large values, the AIC score stabilises as the global shrinkage parameter increases. This can be attributed to the flexibility of the local scale parameters, which compensate for the larger global scale parameter, thus still effectively capturing the magnitude of local effects. Hence, instead of attempting to identify the globally AIC minimising value of τ_k^2, which is computationally expensive, 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, using a suitable grid of M increasing values $\{\tau_{k,1}^2, \ldots, \tau_{k,M}^2\}$, we set τ_k^2 to be

$$\tau_{k,\mathrm{AIC}}^2 = \min\{\tau_{k,m}^2 : \left|\mathrm{AIC}(\tau_{k,m}^2) - \mathrm{AIC}(\tau_{k,m-1}^2)\right| < \epsilon\}$$

for some convergence tolerance ϵ.

By selecting the τ_k's separately, we allow for different sparsity levels across networks. In practice, our jointGHS implementation runs the single-network approach on each network separately (optionally in parallel for computational efficiency), sets τ_k using the above procedure and then uses them in the final joint network run.

5.4. *More on the heavy horseshoe tail.* In the joint graphical horseshoe, information is shared through the common latent parameter ν_{ij}. In practice, only the full conditional expectation of $1/\nu_{ij}$ is used in the ECM algorithm. The larger $1/\nu_{ij}$ in (9), the larger the CM-updates (10) 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 (10) that a conditional expectation of $1/\nu_{ij}$ close to zero 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 suggests no edge. This is illustrated by Figure 1, which shows the precision elements estimated by the joint graphical horseshoe for $K = 2$ networks, plotted against the posterior expectations of the corresponding shared latent parameters $1/\nu_{ij}$. The true networks have 99 edges each, of which 39 in common. 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, that is, shared information is not found, posterior output still captures edges (i.e., nonzero θ_{ijk}) specific to each network. This illustrates how the joint graphical horseshoe estimator can simultaneously share information between networks and capture their differences.

6. Simulations. To evaluate the performance of our approach, we have performed comprehensive simulation studies in R (R Core Team (2013)). We have generated data as close as possible to our omics application of interest, with nonzero 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 common assumption for omics data (Chen and Sharp (2004)). We assess graph accuracy by the precision, that is, the fraction of the inferred edges that are actually present in the true graph (also known as positive predictive value or complement of the false discovery rate), and by the recall, that is, the fraction of edges in the true graph that are present in the inferred one (also known as sensitivity or true positive rate). Because the networks inferred by the different methods may result in different sparsity estimates, some consideration is needed when comparing their precision and recall. For example, the recall tends to increase as the number of edges increases, favouring methods that overselect edges. In omics applications, one rather wishes to identify few but highly reliable associations than a large number of associations, many of which will be spurious. In the discussion of the results,

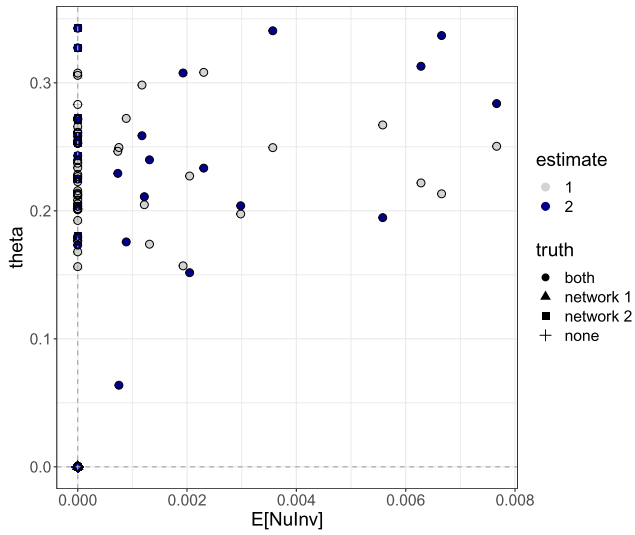

FIG. 1. *Off-diagonal precision matrix elements estimated by the joint graphical horseshoe, plotted against the expectation of the corresponding inverse shared latent parameters for local scales. The data sets consist of* $K = 2$ *networks with* 40% *edge agreement, with* $p = 100$ *nodes, and* $n_1 = 100$ *and* $n_2 = 150$ *observations, respectively. Both networks have simulated sparsity of* 0.02. *The shade of a point indicates which network it is estimated for, and the shape indicates whether the corresponding edge truly is present in both, one or neither networks. The details of the data generation and analysis are given in Section S.3 of Supplementary Material A (Lingjærde et al. (2024b)).*

we, therefore, put more emphasis on the precision and 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 Section 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, Craig and Bhadra (2019) and to the graphical lasso (Friedman, Hastie and Tibshirani (2008)). Additional simulations demonstrating that fastGHS is robust to random initialisations can be found in Section S.7.4 of Supplementary Material A (Lingjærde et al. (2024b)). In Section 6.2 we demonstrate that, thanks to joint modelling, the accuracy of the jointGHS increases with the number of related networks. Finally, in Section 6.3 we compare the performance of jointGHS to the Bayesian spike-and-slab joint graphical lasso (Li, Mccormick and Clark (2019)), the joint graphical lasso (Danaher, Wang and Witten (2014)) and GemBag (Yang et al. (2021)). Details on all simulation studies are given in Section S.3 of Supplementary Material A (Lingjærde et al. (2024b)), and the corresponding code is available on Github (https://github.com/Camiling/jointGHS_simulations) and in Supplementary Material D (Lingjærde et al. (2024a)).

6.1. *Comparison of fastGHS with the Gibbs sampling scheme for single networks.* In this section we compare the performance of our ECM implementation of the graphical horseshoe to that of the Gibbs sampler by Li, Craig and Bhadra (2019). As a baseline reference, we also provide the results of the widely used graphical lasso algorithm (Friedman, Hastie and Tibshirani (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.

6.1.1. *Runtime profiling.* All above graphical settings give rise to high-dimensional problems: for instance, with $n = 200$ and $p = 100$, 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); running examples for this can be found at https://github.com/Camiling/jointGHS_simulations or in Supplementary Material D (Lingjærde et al. (2024a)). In this comparison we, therefore, only consider up to $p = 100$ nodes. We emphasise that the limitation $p < 200$ for the Gibbs sampler applies to our particular simulation settings. In their simulations, Li, Craig and Bhadra (2019) apply the Gibbs sampler to networks with as many as $p = 400$ nodes, but these networks are sparser than ours and have larger partial correlations of magnitude $0.25 - -0.75$ (ours are of magnitude 0.1–0.2). This makes the networks strongly identifiable from data, and thus fewer singularity and convergence issues are encountered.

6.1.2. *Edge-selection performance*. Table 1 indicates the edge-selection performance of fastGHS is comparable to the Gibbs sampler. The two graphical horseshoe implementations perform better than the graphical lasso in terms of both precision and recall in all but one case. For this exception the graphical lasso has the best performance in terms of precision, likely because it has the sparsest estimate, and, therefore, its inferred edges are more accurate, yet at the expense of a lower recall. The horseshoe-based methods have the best overall performance for a wider range of scenarios.

As anticipated, our runtime profiling for the Gibbs sampler and our ECM implementation of the graphical horseshoe indicate striking differences. Comparing the two types of inference is not straightforward, as they rely on different stopping rules and convergence diagnostics. To alleviate the risk of unfair comparison, we run the Gibbs sampler for a relatively small number of MCMC samples, namely, 1000, after 100 burn-in iterations, while for the ECM algorithm, we use a maximum of 10,000 iterations. Figure 2 shows on a logarithmic scale the CPU time used to infer a network for different numbers of nodes p, with $n = 100$ observations. For $p = 90$ nodes, fastGHS is 30 times faster than the Gibbs sampler. For larger p, only the ECM

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 $N = 20$ replicates and show sparsity, precision and recall as well as their standard errors in parentheses. For each case the highest value of the precision (resp., recall) is marked in bold (resp., italic) and so is the precision (resp. recall) 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)

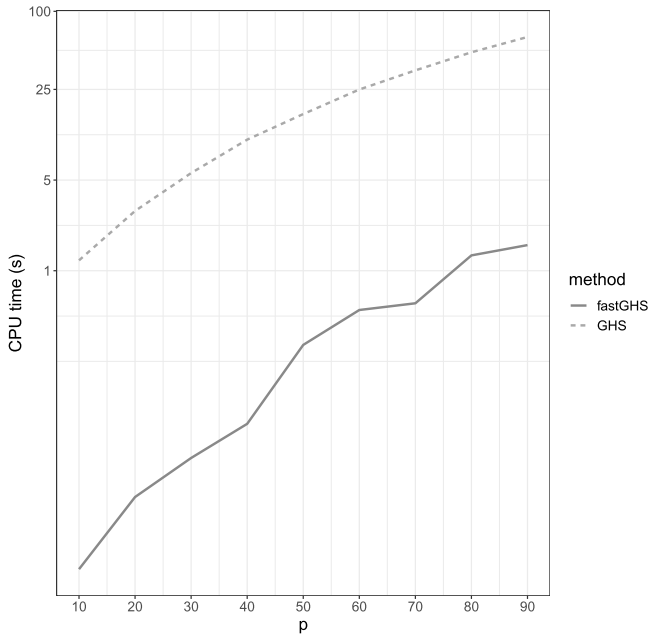

FIG. 2. *CPU time in seconds on a logarithmic scale to infer a network for a grid of node numbers* p *and* $n = 100$ *observations, for our ECM implementation (fastGHS) and the Gibbs sampling implementation (GHS) of the graphical horseshoe. Computations were performed on a* 16*-core Intel Xeon CPU*, 2.60 *GHz.*

estimator can be used, which, in practice, is limited only by the available memory to store the $p \times p$ matrix updates for $\boldsymbol{\Theta}$, $\boldsymbol{\Lambda}$ and $\boldsymbol{N}$.

6.2. *Increased accuracy with joint modelling.* Now that we have established that the performance of our ECM implementation of the graphical horseshoe is comparable to that of the Gibbs sampler, we aim to investigate the gain in statistical power when applying our joint graphical horseshoe estimator, as a function of the number of networks modelled jointly. We use jointGHS to reconstruct $K \in \{2, 4, 10\}$ graphs with $p = 50$ nodes, also applying our single-network method fastGHS on each network separately to serve as baseline. While we simulate scenarios with different degrees of shared information across the K graphs (edge disagreement), for a given scenario the edge disagreement is the same for any pair of networks. Of note, neither the spike-and-slab joint graphical lasso (Li, Mccormick and Clark (2019)) nor the joint graphical lasso (Danaher, Wang and Witten (2014)) can run within reasonable time for the setting with $K = 10$ networks (<48 hours, see Section S.7.3 of Supplementary Material A (Lingjærde et al. (2024b))). The results are averaged over $N = 40$ replicates 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 3 shows the precision and recall for jointGHS and fastGHS as a function of the available information (total number of graphs K) and level of disagreement between them. Although the simulated graph structure remains the same in all settings, the sparsity of the inferred jointGHS graphs varies with total number of graphs and their level of similarity. Hence, to ensure a fair comparison, we obtained single-network estimates with the same sparsity as the joint estimates in each setting, making the fastGHS results vary with both K and the level of similarity; we refer to Section S.3 of Supplementary Material A (Lingjærde et al. (2024b)) for details.

As expected, the joint approach clearly outperforms the single network approach in terms of both precision and recall, and the improvement increases with the number of graphs K,

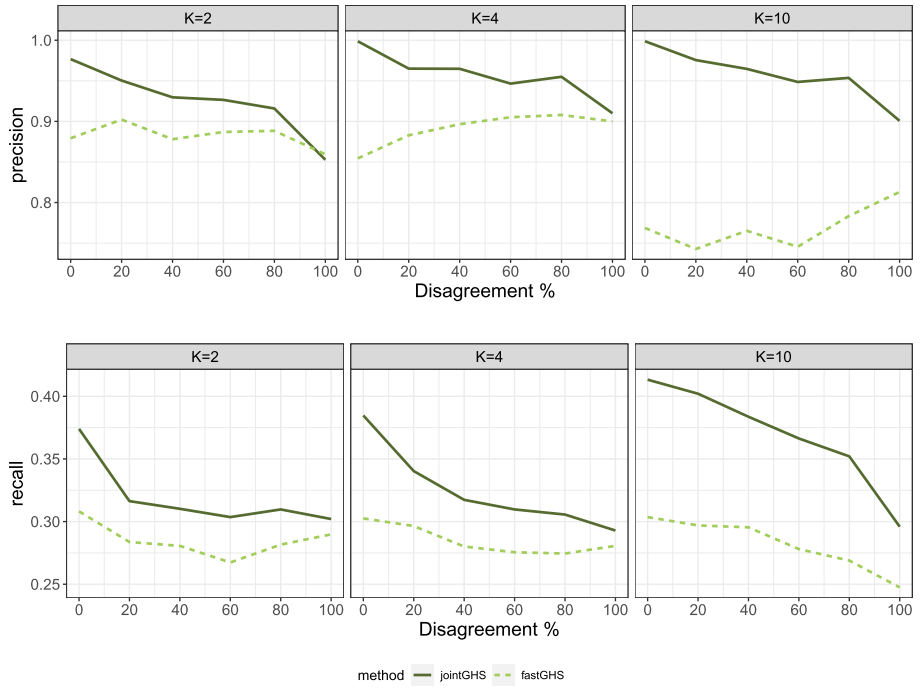

FIG. 3. *Performance of the joint graphical horseshoe (jointGHS) and single network graphical horseshoe (fastGHS), reconstructing $K \in \{2, 4, 10\}$ graphs with $p = 50$ nodes and various similarity of the true graph structures. The edge disagreement between the two graphs is shown as the percentage of edges in one network not present in the other. The results are averaged over $N = 40$ replicates 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.

since more shared information is available. This applies to all levels of edge disagreement, including when the graphs have no common edges. This may be explained by the sparsity of the graphs: while the networks do not share any edge, they do share the fact that no edge is present for a large number of pairs (i, j). Thus, information is still be shared through conditional expectations of the common ν_{ij}^{-1}'s being equal to zero. The larger K is, the stronger this information is, leading to a large improvement compared to a single network approach.

6.3. *Comparison of jointGHS with other joint network inference methods.* Now that we have demonstrated the benefit of joint modelling, we next assess the computational and statistical performance of our joint graphical horseshoe estimator, jointGHS, through comparisons with the Bayesian spike-and-slab joint graphical lasso (SSJGL; Li, Mccormick and Clark (2019)), the joint graphical lasso (JGL; Danaher, Wang and Witten (2014)) and GemBag (Yang et al. (2021)).

6.3.1. *Runtime profiling.* Figure 4 compares the runtime of all methods for a grid of node numbers p, $K \in \{2, 3, 4\}$ networks each with $n_k \in \{100, 150\}$: jointGHS is the fastest of the four methods for all settings, followed by GemBag, JGL and finally SSJGL. The last two approaches become computationally prohibitive as the number of networks K increases. Indeed, Danaher, Wang and Witten (2014) highlight that the JGL algorithm scales well in problems with only two classes ($K = 2$), as a closed-form solution to the generalised fused lasso problem can be obtained in that case (this also holds for SSJGL). Both SSJGL and JGL use the same alternating direction method of multipliers (ADMM) algorithm to update the precision matrix estimates.

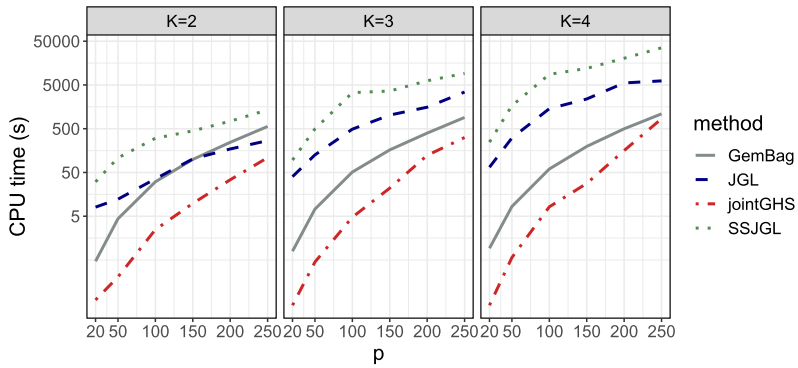

FIG. 4. *CPU time in seconds on a logarithmic scale to jointly infer networks for various numbers of nodes p and networks K, using the joint graphical horseshoe (jointGHS), the spike-and-slab joint graphical lasso (SSJGL), the joint graphical lasso (JGL) and GemBag. In all settings the networks agree on* 50% *of their edges. For $K = 2$, the networks are inferred from $n_1 = 100$ and $n_2 = 150$ observations, for $K = 3$, the networks are inferred from $n_1 = n_2 = 100$ and $n_3 = 150$ observations, and for $K = 4$ the networks are inferred from $n_1 = n_2 = 100$ and $n_3 = n_4 = 150$ observations. Computations were performed on a* 16*-core Intel Xeon CPU,* 2.60 *GHz.*

Notably, in their respective simulation studies, Li, Mccormick and Clark (2019) and Danaher, Wang and Witten (2014) apply their methods to as many as $p = 400$ and $p = 500$ nodes. However, in the SSJGL numerical experiments, precision matrix elements are sampled from the G-Wishart distribution with *three* degrees of freedom, giving very strong partial correlations ($0.5 - 0.9$ in absolute value). Similarly, following the precision matrix construction described in the JGL experiments gives partial correlations of ≈ 0.6 in absolute value. This renders the networks strongly identifiable from data, leading to faster convergence than in our simulations where partial correlations and between 0.1 and 0.2. Thus, motivated by realistic biological network strengths, we are considering a more challenging inference problem.

The lower runtime of jointGHS and GemBag can be explained by their EM/ECM implementations, although their higher scalability compared to SSJGL, which also implements an EM algorithm, could be due to their computationally efficient C++ subroutines. Because of these computational limitations, we use relatively small numbers of nodes p to make, so all methods can run within reasonable time (< 48 hours). Note, however, that jointGHS successfully completes within this timeframe on examples with $p > 1000$ nodes (see Section S.7.3 of Supplementary Material A (Lingjærde et al. (2024b))).

6.3.2. *Edge-selection performance.* We next compare the edge-selection performance of all four methods on problems with $K = 2$ graphs of $p = 50$ nodes each, and $n_1 = 50$ and $n_2 = 80$ observations, respectively. We consider six settings, with different levels of graph similarity, that is, proportion of edges present in both graphs. Namely, we simulate data with similarity varying between 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 we sample $N = 100$ data sets from each of the two corresponding multivariate Gaussian distributions. In all settings both graphs have true sparsity 0.04, corresponding to 49 edges. We report the precision and recall of the final estimate of each method. A threshold-free comparison, based on precision-recall curves and corresponding areas under the curves, is given in Section S.7.2 of Supplementary Material A (Lingjærde et al. (2024b)).

Table 2 shows the performance of the joint network approaches. The joint graphical lasso JGL, applied with its default AIC-based selection criteria for sparsity- and similarity-selection, has low precision in all settings. The method tends to severely overselect edges,

TABLE 2

Performance of the joint graphical horseshoe (jointGHS), the spike-and-slab joint graphical lasso (SSJGL), the joint graphical lasso (JGL) and GemBag for reconstructing $K = 2$ graphs with various similarity of the true graph structures. Results are averaged over $N = 100$ replicates. The edge disagreement between the two simulated graphs is shown as the percentage of edges in one network not present in the other, along with the average edge disagreement of the graphs estimated by each method. The graphs are simulated from a multivariate Gaussian distribution with $p = 50$ variables and with $n_1 = 50$ and $n_2 = 80$ observations; both graphs have simulated sparsity of 0.04. The estimated sparsity, precision and recall for both graphs is reported, with 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

			$n_1 = 50$			$n_2 = 80$		
Disagr. %	Method	$\widehat{\text{Disagr.}}$ %	Sparsity	Precision	Recall	Sparsity	Precision	Recall
0	JGL	63	0.312 (0.011)	0.11 (0.01)	0.83 (0.05)	0.269 (0.011)	0.13 (0.01)	0.91 (0.04)
	GemBag	41	0.039 (0.006)	0.65 (0.07)	0.62 (0.06)	0.064 (0.008)	0.49 (0.05)	0.77 (0.06)
	SSJGL	0	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	17	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	65	0.310 (0.02)	0.11 (0.01)	0.83 (0.05)	0.266 (0.017)	0.14 (0.01)	0.90 (0.04)
	GemBag	42	0.041 (0.007)	0.61 (0.07)	0.61 (0.06)	0.064 (0.01)	0.47 (0.06)	0.73 (0.06)
	SSJGL	1	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	25	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	67	0.291 (0.048)	0.12 (0.02)	0.82 (0.06)	0.252 (0.059)	0.14 (0.05)	0.83 (0.08)
	GemBag	49	0.038 (0.005)	0.57 (0.08)	0.54 (0.05)	0.066 (0.007)	0.41 (0.04)	0.66 (0.05)
	SSJGL	1	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	34	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	69	0.291 (0.049)	0.12 (0.02)	0.82 (0.07)	0.247 (0.058)	0.15 (0.05)	0.84 (0.07)
	GemBag	48	0.039 (0.004)	0.52 (0.07)	0.50 (0.05)	0.064 (0.009)	0.43 (0.04)	0.67 (0.06)
	SSJGL	2	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	47	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	71	0.299 (0.040)	0.11 (0.02)	0.81 (0.06)	0.259 (0.046)	0.14 (0.04)	0.89 (0.06)
	GemBag	47	0.040 (0.005)	0.46 (0.07)	0.46 (0.06)	0.068 (0.008)	0.42 (0.05)	0.70 (0.06)
	SSJGL	2	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	59	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	72	0.313 (0.012)	0.11 (0.01)	0.83 (0.05)	0.251 (0.009)	0.16 (0.01)	0.98 (0.02)
	GemBag	43	0.050 (0.004)	0.34 (0.05)	0.43 (0.06)	0.074 (0.006)	0.48 (0.04)	0.89 (0.04)
	SSJGL	3	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	78	0.015 (0.002)	**0.65** (0.12)	0.24 (0.04)	0.013 (0.002)	**0.95** (0.06)	0.31 (0.05)

reporting nearly 10 times more edges as the number simulated edges: this leads to high recall values but very low precision.

For GemBag, overselection of edges is not as severe as for (JGL), but the method still reports more edges than the joint graphical horseshoe (jointGHS) and the spike-and-slab joint graphical lasso (SSJGL) in all settings, which results in a lower precision but a higher recall than the two methods. Interestingly, while the first network with $n_1 = 50$ seems to benefit from GemBag's joint modelling, it is not the case of the second network with the largest sample size $n_2 = 80$, as its estimation does not improve as the level of similarity between the two networks increases. Further, due to the larger sample size, GemBag selects more edges for the second network in all settings, which yields a higher recall yet a lower precision than for the first network. Remarkably, the disagreement level (percentage of edges present in only one of the two networks) of the estimated GemBag networks remains almost the same in all settings and hence does not reflect the different simulated similarity between the two graphs: the method does not appear to adapt to varying network similarity levels, with possibly too much information shared between unrelated networks, yet too little information shared between highly related ones.

The precision of jointGHS is either higher or comparable to that of SSJGL in all settings. In general, jointGHS is more conservative than the other methods and hence is most suitable for detecting edges with high confidence. This is further exemplified in our extended simulations, particularly in our threshold-free comparisons for sparse edge selection where jointGHS in all settings considered had the highest precision for a given recall level (Section S.7.1 of Supplementary Material A (Lingjærde et al. (2024b))). The graphs estimated by SSJGL are denser, which tends to result in a large recall values: this holds for very similar networks, where SSJGL has the highest recall, yet as network dissimilarity increases, the recall of SSJGL decreases and becomes similar to that of jointGHS. This happens because SSJGL shrinks all precision matrices toward a common structure, thereby overselecting edges that are absent in some networks while being present in others, as exemplified further in Section 6.3.3. Table 2 indicates that the two networks are estimated by jointGHS as being increasingly different from each other as the simulated level of disagreement increases, while they are invariably estimated as almost identical by SSJGL, even when the true networks are completely unrelated. Of note, the precision of SSJGL deteriorates in settings where the simulated networks share little information, while jointGHS effectively adapts to this setting, maintaining relatively high values of both the precision and recall for completely unrelated networks. As further demonstrated in Section 6.3.3 hereafter, this clear advantage can be attributed to the local scales λ_{ijk} of the graphical horseshoe, which flexibly capture isolated effects thanks to their heavy Cauchy tails.

6.3.3. *Ability to capture edges on the individual-graph level.* We next provide a more detailed illustration of the benefits of the horseshoe heavy-tailed local scales for capturing graph-specific edges by comparing precision matrix estimates obtained with our joint graphical horseshoe, jointGHS and the spike-and-slab joint graphical lasso, SSJGL. We use the same data generation procedure as in Section 5.4, with networks reconstructed from two data sets corresponding to $K = 2$ graphs with 40% edge agreement, $p = 100$ nodes, and $n_1 = 100$ and $n_2 = 150$ observations, respectively. Both graphs have a true sparsity of 0.02.

Figure 5 indicates that jointGHS effectively identifies both common and graph-specific edges. Thanks to its network-specific local scales, when an edge is present in only one of the graphs, the corresponding precision matrix element in the other graph is correctly estimated as null; hence, no false positive is reported due to excess shrinkage toward a common graph. As a result, jointGHS is more inclined to false negatives than false positives; in a few instances, it reports edges in only one of the two networks, while they actually present in

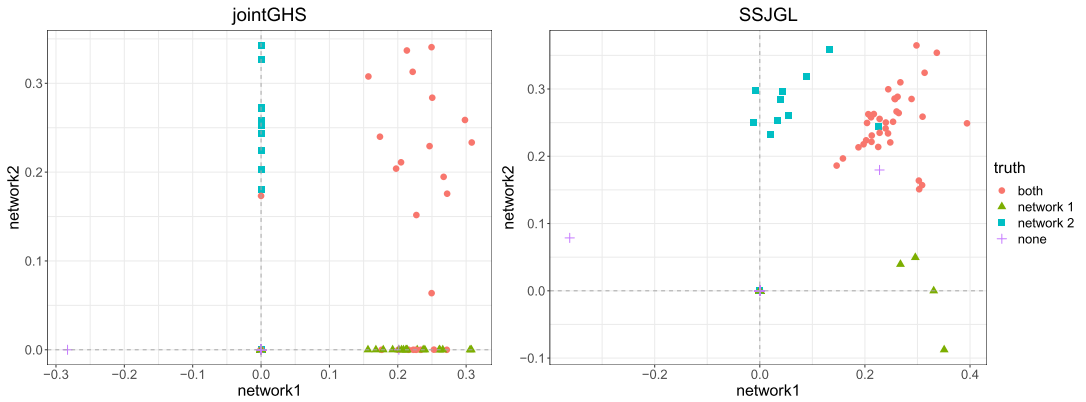

FIG. 5. *Comparison of estimated precision matrix elements (scaled by the diagonal as when finding partial correlations) of two networks with 40% edge agreement and $p = 100$ nodes, for the joint graphical horseshoe and for the spike-and-slab joint graphical lasso separately. The points' shape indicates whether the corresponding edge is simulated as present is both, one or neither network.*

both. SSJGL displays the opposite behaviour: it shrinks excessively toward a common graph and, therefore, largely fails to identify the network-specific edges. While the SSJGL does well in capturing edges common to both graphs, an edge in one graph tends to be reported as present in both, resulting in 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, as discussed in Table 2. Regardless of how similar the networks may be, jointGHS effectively borrows shared information across them, while successfully avoiding overshrinkage toward a common structure to preserve graph-specific information.

While our simulations have illustrated the flexibility of the joint graphical horseshoe, performing joint modelling of K data sets consisting of many highly similar networks and a few unrelated or a priori more loosely similar networks would make little sense. While, thanks to the horseshoe local scales, the risk of the many similar networks dominating the analysis is lower with jointGHS compared to other joint graphical methods, including the spike-and-slab joint graphical lasso (Li, Mccormick and Clark (2019)) and the joint graphical lasso (Danaher, Wang and Witten (2014)); it may be helpful to investigators to rule out this scenario. To this end, we outline a Bayesian bootstrapping procedure (Rubin (1981)) in Section S.6 of Supplementary Material A (Lingjærde et al. (2024b)) to check whether the joint network estimates are in strong contradiction with each of the single network estimates; this optional routine is implemented in our R package `jointGHS`.

7. Application to a study of hotspot activity with stimulated monocyte expression. Returning to the monocyte data set from Section 3, 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 genetic variant (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 hotspot *in each condition* was 294, 88, 16 and 215, respectively (permutation-based FDR < 0.05); hereafter, we focus on the $p = 381$ genes associated with the hotspot in at least one condition. Further information on the data and preprocessing steps is available in Ruffieux et al. (2020), and details on the analysis presented below can be found in Section S.4 of Supplementary Material A (Lingjærde et al. (2024b)). Complete lists of the node degree of all genes in the estimated networks and all edges identified by the method are given in Supplementary Material B and C (Lingjærde et al. (2024c, 2024d)), respectively.

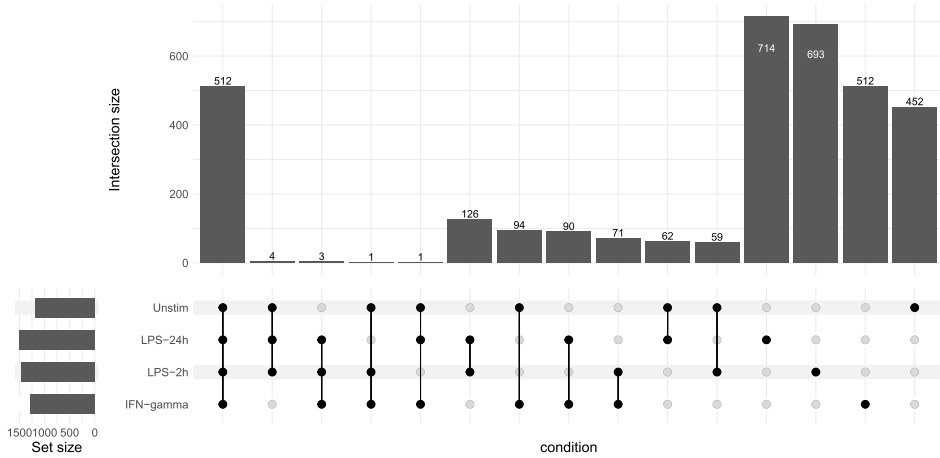

FIG. 6. *Upset plot of the joint graphical horseshoe graphs of the monocyte data, an alternative to a Venn diagram, showing the number of edges shared strictly between given conditions (Conway, Lex and Gehlenborg (2017)). For each intersection the number of edges shared only by the corresponding conditions is shown. The total number of edges for each condition is represented on the left bar plot. Every possible intersection is represented by the bottom plot, and their occurrence is shown in the top bar plot.*

We first present and interpret the results obtained by applying jointGHS to jointly estimate the precision matrices and hence network structures of the genes in the four conditions. We then describe a comparative study with: (i) the classical graphical horseshoe applied separately to each network (using our fastGHS ECM implementation, which scales to this problem) and (ii) a competing joint modelling approach. For the latter comparison, we use GemBag (Yang et al. (2021)), as neither the spike-and-slab joint graphical lasso (Li, Mccormick and Clark (2019)) nor the joint graphical lasso (Danaher, Wang and Witten (2014)) runs within 48 hours on the data; as previously discussed in Section 6.3.1, both algorithms become substantially slower in problems with $K > 2$ classes, due to the absence of closed-form solution for $K \neq 2$.

Figure 6 shows the number of edges shared between the different conditions, as inferred by jointGHS. Many edges are common to all four networks, suggesting a high degree of similarity across all monocytic conditions, likely reflecting the effect of the shared hotspot control. Very few edges are shared across three conditions only, but many pairs of conditions have edges that are shared strictly between them. In particular, LPS 2h and LPS 24h have the largest number of shared edges that are not present in the other two conditions, which is expected as they correspond to an exposition to differing durations of a same lipopolysaccharide activation. While LPS is a component of gram-negative bacterial cell walls, IFN-γ is a cytokine important in myobacterial and viral infections (Fairfax et al. (2014)). In addition and in line with the results of our simulation studies, jointGHS is able to capture many condition-specific edges, with LPS 24h having the highest number of unique edges, possibly because it corresponds to the densest graph across all conditions. These observations call for further biological investigations, which may motivate new mechanistic studies, such as whether groups of edges shared by two or more conditions pertain to known pathway activation or whether pathways of genes involved in edges unique to one stimulated condition are indicative of some condition-specific functional mechanisms. We explore such questions in the next sections.

Network-specific activity. A number of the network-specific structures identified by jointGHS warrant close inspection. For example, the Cytochrome C Oxidase Subunit 6A1 (*COX6A1*) gene has large degree in both LPS 2h and LPS 24h, moderately large degree in

IFN-γ but low degree in the unstimulated condition. 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. Another example pertains to the PHD-finger 1 protein encoding gene (*PHF1*), which has high degree only in the IFN-γ condition, where it is also found to be controlled by the top hotspot rs6581889. 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 (Baker, Allis and Wang (2008), Liu et al. (2018)). The centrality of *PHF1* in the network of the IFN-γ stimulated monocytes suggests a potential role in the immune reaction and provides a relevant alley for further studies.

Hub genes. Investigating hub genes in the jointGHS networks for the different conditions can help gain better understanding of the immune response driving disease mechanisms. 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 (Section S.4 of Supplementary Material A (Lingjærde et al. (2024b))). 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, Org and Rebane (2008)). Mutations in this gene have been associated with autoimmune polyendocrinopathy-candidiasis-ectodermal dystrophy (*APECED*), distinguished by multiorgan autoimmunity (Mathis and Benoist (2007), Akirav, Ruddle and Herold (2011)). Similarly, the arylformamidase (*AFMID*) gene, also known as kynurenine formamidase, is found to have among the higher degrees in all four conditions, in addition to being associated with the top hotspot in all; it also has a link to *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)). Additional biological results and discussion can be found in Section S.4 of Supplementary Material A (Lingjærde et al. (2024b)).

Hotspot control. We next explore the extent to which the top genetic hotspot rs6581889 influences the conditional independence structure of the genes it controls. Using permutation testing to derive empirical p-values, we find that the subnetwork of genes associated with rs6581889 in each condition 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 *LYZ* gene. As *LYZ* is located a few Kb away from rs6581889, this may suggest a mediation of the hotspot effect on other genes via *LYZ*—a hypothesis already examined in different studies (Fairfax et al. (2012), Ruffieux et al. (2021)) but which would require experimental validation or dedicated inspection, for example, with Mendelian randomisation analysis. The findings are summarised in Table 3.

Comparison to the single-network analysis. We next aim to assess the possible added value of joint modelling for increasing biological insight by comparing the jointGHS results to those of our single-network ECM implementation of the graphical horseshoe, fastGHS. To obtain comparable networks, we use for fastGHS the same sparsity levels as in the jointGHS estimates for each condition separately (see Section S.4 of Supplementary Material A (Lingjærde et al. (2024b)) for details). The subgraph of hotspot-controlled genes inferred by jointGHS is denser as that inferred by fastGHS in all conditions; a dense graph agrees with the expectation that the hotspot triggers substantial activity among the controlled genes (Ruffieux

TABLE 3

Sparsity and hotspot control with jointGHS. Sparsity of the jointGHS subnetwork of top-hotspot-controlled genes as well as the overall sparsity of the full network in each condition. Proportion of neighbour genes of the two genes proximal to the hotspot (LYZ and YEATS4) as well as the overall proportion in the different conditions. For each condition a proportion higher than the overall proportion is marked in bold with statistical significance (compared to randomly sampled subsets of genes of the same size) 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

et al. (2020)). Moreover, both *LYZ* and *YEATS4* have a more central role in the joint network, with more edges to other genes as well as more edges among their neighbours. This lends further support to the mediation hypothesis formulated above. Finally, many genes directly associated with *LYZ* and *YEATS4* have very high degree, suggesting that their interplay with other genes could be relevant for disease-driving mechanisms. All these observations highlight the biological insight gained by sharing information across networks with jointGHS.

Comparison to the GemBag analysis. The comparison of jointGHS with the other joint modelling approach GemBag (Yang et al. (2021)) is also informative. Strikingly, GemBag identifies a strict subset of the edges identified by jointGHS. Moreover, almost all edges are estimated as shared by all four conditions, and very few are condition-specific edges; this stands in strong contrast to jointGHS, which finds many condition-specific edges. Similarly, the four conditions have an almost complete overlap of top hubs (genes with node degree larger than the 90th percentile) in the GemBag networks, again contrasting the jointGHS networks where the top hubs are mainly network-specific. The biological plausibility of the network specificities discussed above lends support to the argument that jointGHS succeeds at capturing network-specific effects thanks to its horseshoe local scales. Further, *LYZ* and *YEATS4* have very low node degrees in all conditions in the GemBag networks, whereas the two genes are central in the jointGHS networks, with a large number of neighbours, many central neighbours (i.e., hubs) and many top hotspot controlled neighbours, which align with evidence from previous studies (Fairfax et al. (2012), Ruffieux et al. (2021)). Further results and details on comparison of GemBag and jointGHS are given in Section S.4 of Supplementary Material A (Lingjærde et al. (2024b)).

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 coregulation and coexpression 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. Conclusion. We have introduced an efficient ECM algorithm for jointly estimating the precision matrix in the graphical horseshoe and their associated local scale, fastGHS, and a novel joint graphical horseshoe estimator for multiple-network inference, jointGHS. Through simulations we have shown that fastGHS achieves equivalent performance to the

fully Bayes graphical horseshoe Gibbs sampler for selecting edges while being substantially more scalable. In the multiple-network setting, we have also shown that jointGHS successfully shares information between networks while capturing their differences, outperforming competing methods, such as the joint graphical lasso, GemBag and the spike-and-slab joint graphical lasso, which can be very anticonservative. 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 jointGHS can be attributed to the horseshoe heavy-tailed local scales, which are able to adapt even in the absence of shared information, favouring the detection of isolated network-specific edges. To date, 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. Finally, while our ECM implementation does not provide a fully Bayesian solution, we have demonstrated that this does not affect performance for our primary inference goal, namely, edge selection, but now enables estimation for problems of realistic dimensions, which was a key ambition of our work. If desired, parameter uncertainty could still be quantified using an additional bootstrapping procedure.

We have taken advantage of jointGHS to study the gene regulation mechanisms underpinning immune-mediated diseases, using monocyte expression from four immune stimulation conditions. Joint inference on these data identified biologically-supported links and allowed us to formulate sound mechanistic hypotheses.

As our interest lies in the computational aspect of the algorithm, our ECM algorithm is formulated to find modes of the joint posterior of the precision matrices and the local scales, which ensures analytical tractability of the objective function and computational effectiveness. This leads to a different posterior landscape compared to that induced by the marginal posterior distribution of the precision matrices alone; multimodality may be exacerbated in the former case. While our simulations suggest that the performance of our algorithm is barely affected by this problem, there are multiple ways to alleviate the risk of entrapment into local modes. As the objective function in an ECM algorithm is typically not concave, a common recommendation is to perform multiple runs using different starting values (McLachlan and Basford (1988)). In practice, this could be done by implementing multiple restarts and performing Bayesian averaging, for example, by weighing estimates by their objective function value. Another approach to improve the chances of finding global modes would be to implement annealing schemes (Kirkpatrick, Gelatt and Vecchi (1983)). While out of scope for this paper, these are relevant alleys of future work.

There are many possible extensions. For example, given the increasing prevalence of longitudinal studies, a natural continuation would be to propose a time-variant version of the model. Such an extension could be particularly profitable for studies aimed at understanding disease progression and may also be relevant for the omics application of this paper, where two of the monocyte conditions involved exposure to differing durations of lipopolysaccharide (LPS 2h and LPS 24h). In settings with large numbers of timepoints, autoregression-like approach could be developed, where information would be shared between successive time points.

To conclude, our approach is, to our knowledge, the first to extend graphical models based on global-local priors to the multiple network setting. Additionally, thanks to its remarkable scalability, jointGHS effectively bridges the gap between Bayesian joint network modelling and large-scale inference for real-world studies such as encountered in statistical omics.

Acknowledgments. C. Lingjærde, H. Ruffieux and S. Richardson developed the methodology. C. Lingjærde developed the software, performed the data analysis and drafted the manuscript. B. P. Fairfax contributed to the data anlysis. All authors have contributed to the interpretation of results. All authors have contributed to the manuscript, and read and approved the final version.

Software. The ECM graphical horseshoe approach for single or multiple network inference has been implemented in the R packages `fastGHS` and `jointGHS`, with all subroutines implemented in C++ for computational efficiency.

Data Availability. The $CD14^+$ monocyte gene expression data used in this study has been generated with HumanHT-12 v4 arrays and freely available for downloading from ArrayExpress45 (accession E-MTAB-2232, Fairfax et al. (2012, 2014))

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.).

SUPPLEMENTARY MATERIAL

Supplement A (DOI: 10.1214/23-AOAS1863SUPPA; .pdf). Additional details on the implementation of the method, simulations and real omics data analysis.

Supplement B (DOI: 10.1214/23-AOAS1863SUPPB; .zip). List of the degree of all genes in the jointGHS network, for all conditions of the monocyte gene expression study.

Supplement C (DOI: 10.1214/23-AOAS1863SUPPC; .zip). List of all edges in the jointGHS network, for all conditions of the monocyte gene expression study. The first two columns contain the names of the genes in each edge, and the third column contains the value of their partial correlation. The fourth column indicates the condition.

Supplement D (DOI: 10.1214/23-AOAS1863SUPPD; .zip). R packages `fastGHS` and `jointGHS` for fast single and multiple network inference with the graphical horseshoe, and R code for all simulations and data analyses.

REFERENCES

AKAIKE, H. (1973). Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory* (*Tsahkadsor*, 1971) 267–281. Akad. Kiadó, Budapest. MR0483125

AKIRAV, E. M., RUDDLE, N. H. and HEROLD, K. C. (2011). The role of AIRE in human autoimmune disease. *Nat. Rev. Endocrinol.* **7** 25–33. https://doi.org/10.1038/nrendo.2010.200

BAKER, L. A., ALLIS, C. D. and WANG, G. G. (2008). PHD fingers in human diseases: Disorders arising from misinterpreting epigenetic marks. *Mutat. Res.* **647** 3–12.

BHADRA, A., DATTA, J., POLSON, N. G. and WILLARD, B. (2017). The horseshoe+ estimator of ultra-sparse signals. *Bayesian Anal.* **12** 1105–1131. MR3724980 https://doi.org/10.1214/16-BA1028

BHADRA, A., DATTA, J., POLSON, N. G. and WILLARD, B. (2019). Lasso meets horseshoe: A survey. *Statist. Sci.* **34** 405–427. MR4017521 https://doi.org/10.1214/19-STS700

BISWAS, S. K. and MANTOVANI, A. (2010). Macrophage plasticity and interaction with lymphocyte subsets: Cancer as a paradigm. *Nat. Immunol.* **11** 889–896. https://doi.org/10.1038/ni.1937

BUSATTO, C. and STINGO, F. C. (2023). Inference of multiple high-dimensional networks with the graphical horseshoe prior. arXiv preprint. Available at arXiv:2302.06423.

CARVALHO, C. M., POLSON, N. G. and SCOTT, J. G. (2009). Handling sparsity via the horseshoe. In *Artificial Intelligence and Statistics* 73–80. PMLR.

CARVALHO, C. M., POLSON, N. G. and SCOTT, J. G. (2010). The horseshoe estimator for sparse signals. *Biometrika* **97** 465–480. MR2650751 https://doi.org/10.1093/biomet/asq017

CHEN, H. and SHARP, B. M. (2004). Content-rich biological network constructed by mining PubMed abstracts. *BMC Bioinform.* **5** 1–13.

CONWAY, J. R., LEX, A. and GEHLENBORG, N. (2017). UpSetR: An R package for the visualization of intersecting sets and their properties. *Bioinformatics*.

DANAHER, P., WANG, P. and WITTEN, D. M. (2014). The joint graphical lasso for inverse covariance estimation across multiple classes. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **76** 373–397. MR3164871 https://doi.org/10.1111/rssb.12033

DEMPSTER, A. P., LAIRD, N. M. and RUBIN, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B* **39** 1–38. MR0501537

DOBROVOLSKY, V. N., BOWYER, J. F., PABARCUS, M. K., HEFLICH, R. H., WILLIAMS, L. D., DOERGE, D. R., ARVIDSSON, B., BERGQUIST, J. and CASIDA, J. E. (2005). Effect of arylformamidase (kynurenine formamidase) gene inactivation in mice on enzymatic activity, kynurenine pathway metabolites and phenotype. *Biochim. Biophys. Acta, Gen. Subj.* **1724** 163–172.

FAIRFAX, B. P., HUMBURG, P., MAKINO, S., NARANBHAI, V., WONG, D., LAU, E., JOSTINS, L., PLANT, K., ANDREWS, R. et al. (2014). Innate immune activity conditions the effect of regulatory variants upon monocyte gene expression. *Science* **343** 1246949.

FAIRFAX, B. P., MAKINO, S., RADHAKRISHNAN, J., PLANT, K., LESLIE, S., DILTHEY, A., ELLIS, P., LANGFORD, C., VANNBERG, F. O. et al. (2012). Genetics of gene expression in primary immune cells identifies cell type-specific master regulators and roles of HLA alleles. *Nat. Genet.* **44** 502–510. https://doi.org/10.1038/ng.2205

FAN, J., FENG, Y. and WU, Y. (2009). Network exploration via the adaptive lasso and SCAD penalties. *Ann. Appl. Stat.* **3** 521–541. MR2750671 https://doi.org/10.1214/08-AOAS215

FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9** 432–441. https://doi.org/10.1093/biostatistics/kxm045

GUO, J., LEVINA, E., MICHAILIDIS, G. and ZHU, J. (2011). Joint estimation of multiple graphical models. *Biometrika* **98** 1–15. MR2804206 https://doi.org/10.1093/biomet/asq060

HAO, L., SAKURAI, A., WATANABE, T., SORENSEN, E., NIDOM, C. A., NEWTON, M. A., AHLQUIST, P. and KAWAOKA, Y. (2008). Drosophila RNAi screen identifies host genes important for influenza virus replication. *Nature* **454** 890–893. https://doi.org/10.1038/nature07151

HUGILL, A. J., STEWART, M. E., YON, M. A., PROBERT, F., COX, I. J., HOUGH, T. A., SCUDAMORE, C. L., BENTLEY, L., WALL, G. et al. (2015). Loss of arylformamidase with reduced thymidine kinase expression leads to impaired glucose tolerance. *Biol. Open* **4** 1367–1375.

KARCZEWSKI, K. J. and SNYDER, M. P. (2018). Integrative omics for health and disease. *Nat. Rev. Genet.* **19** 299–310.

KIM, S., BECKER, J., BECHHEIM, M., KAISER, V., NOURSADEGHI, M., FRICKER, N., BEIER, E., KLASCHIK, S., BOOR, P. et al. (2014). Characterizing the genetic basis of innate immune response in TLR4-activated human monocytes. *Nat. Commun.* **5** 1–7.

KIRKPATRICK, S., GELATT, C. D. JR. and VECCHI, M. P. (1983). Optimization by simulated annealing. *Science* **220** 671–680. MR0702485 https://doi.org/10.1126/science.220.4598.671

KOOK, J. H., VAUGHN, K. A., DEMASTER, D. M., EWING-COBBS, L. and VANNUCCI, M. (2021). BVAR-connect: A variational Bayes approach to multi-subject vector autoregressive models for inference on brain connectivity networks. *Neuroinformatics* **19** 39–56.

KYEWSKI, B. and KLEIN, L. (2006). A central role for central tolerance. *Annu. Rev. Immunol.* **24** 571–606. https://doi.org/10.1146/annurev.immunol.23.021704.115601

LAURITZEN, S. L. (1996). *Graphical Models. Oxford Statistical Science Series* **17**. The Clarendon Press, Oxford University Press, New York. MR1419991

LEE, M. N., YE, C., VILLANI, A.-C., RAJ, T., LI, W., EISENHAURE, T. M., IMBOYWA, S. H., CHIPENDO, P. I., RAN, F. A. et al. (2014). Common genetic variants modulate pathogen-sensing responses in human dendritic cells. *Science* **343** 1246980.

LI, Y., CRAIG, B. A. and BHADRA, A. (2019). The graphical horseshoe estimator for inverse covariance matrices. *J. Comput. Graph. Statist.* **28** 747–757. MR4007755 https://doi.org/10.1080/10618600.2019.1575744

LI, Z., MCCORMICK, T. and CLARK, S. (2019). Bayesian joint spike-and-slab graphical lasso. In *International Conference on Machine Learning* 3877–3885. PMLR.

LINGJÆRDE, C., FAIRFAX, B. P., RICHARDSON, S. and RUFFIEUX, H. (2024a). Supplement D to "Scalable multiple network inference with the joint graphical horseshoe." https://doi.org/10.1214/23-AOAS1863SUPPD

LINGJÆRDE, C., FAIRFAX, B. P., RICHARDSON, S. and RUFFIEUX, H. (2024b). Supplement A to "Scalable multiple network inference with the joint graphical horseshoe." https://doi.org/10.1214/23-AOAS1863SUPPA

LINGJÆRDE, C., FAIRFAX, B. P., RICHARDSON, S. and RUFFIEUX, H. (2024c). Supplement B to "Scalable multiple network inference with the joint graphical horseshoe." https://doi.org/10.1214/23-AOAS1863SUPPB

LINGJÆRDE, C., FAIRFAX, B. P., RICHARDSON, S. and RUFFIEUX, H. (2024d). Supplement C to "Scalable multiple network inference with the joint graphical horseshoe." https://doi.org/10.1214/23-AOAS1863SUPPC

LINGJÆRDE, C. and RICHARDSON, S. (2023). StabJGL: A stability approach to sparsity and similarity selection in multiple network reconstruction. arXiv preprint. Available at arXiv:2306.03212.

LISTON, A., LESAGE, S., WILSON, J., PELTONEN, L. and GOODNOW, C. C. (2003). Aire regulates negative selection of organ-specific T cells. *Nat. Immunol.* **4** 350–354.

LIU, R., GAO, J., YANG, Y., QIU, R., ZHENG, Y., HUANG, W., ZENG, Y., HOU, Y., WANG, S. et al. (2018). 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 Res*. **46** 6608–6626.

MAKALIC, E. and SCHMIDT, D. F. (2015). A simple sampler for the horseshoe estimator. *IEEE Signal Process. Lett*. **23** 179–182.

MATHIS, D. and BENOIST, C. (2007). A decade of AIRE. *Nat. Rev., Immunol*. **7** 645–650. https://doi.org/10.1038/nri2136

MCLACHLAN, G. J. and BASFORD, K. E. (1988). *Mixture Models*: *Inference and Applications to Clustering*. *Statistics*: *Textbooks and Monographs* **84**. Dekker, New York. MR0926484

MEINSHAUSEN, N. and BÜHLMANN, P. (2006). High-dimensional graphs and variable selection with the lasso. *Ann. Statist*. **34** 1436–1462. MR2278363 https://doi.org/10.1214/009053606000000281

MENG, X.-L. and RUBIN, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika* **80** 267–278. MR1243503 https://doi.org/10.1093/biomet/80.2.267

NI, Y., BALADANDAYUTHAPANI, V., VANNUCCI, M. and STINGO, F. C. (2022). Bayesian graphical models for modern biological applications. *Stat. Methods Appl*. **31** 197–225. MR4426829 https://doi.org/10.1007/s10260-021-00572-8

PETERSON, C., STINGO, F. C. and VANNUCCI, M. (2015). Bayesian inference of multiple Gaussian graphical models. *J. Amer. Statist. Assoc*. **110** 159–174. MR3338494 https://doi.org/10.1080/01621459.2014.896806

PETERSON, P., ORG, T. and REBANE, A. (2008). Transcriptional regulation by AIRE: Molecular mechanisms of central tolerance. *Nat. Rev., Immunol*. **8** 948–957. https://doi.org/10.1038/nri2450

PIIRONEN, J. and VEHTARI, A. (2017). Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electron. J. Stat*. **11** 5018–5051. MR3738204 https://doi.org/10.1214/17-EJS1337SI

POLSON, N. G. and SCOTT, J. G. (2010). Shrink globally, act locally: Sparse Bayesian regularization and prediction. *Bayesian Stat*. **9** 105.

ROČKOVÁ, V. and GEORGE, E. I. (2014). EMVS: The EM approach to Bayesian variable selection. *J. Amer. Statist. Assoc*. **109** 828–846. MR3223753 https://doi.org/10.1080/01621459.2013.869223

ROČKOVÁ, V. and GEORGE, E. I. (2018). The spike-and-slab LASSO. *J. Amer. Statist. Assoc*. **113** 431–444. MR3803476 https://doi.org/10.1080/01621459.2016.1260469

RUBIN, D. B. (1981). The Bayesian bootstrap. *Ann. Statist*. **9** 130–134. MR0600538

RUFFIEUX, H., DAVISON, A. C., HAGER, J., INSHAW, J., FAIRFAX, B. P., RICHARDSON, S. and BOTTOLO, L. (2020). A global-local approach for detecting hotspots in multiple-response regression. *Ann. Appl. Stat*. **14** 905–928. MR4117834 https://doi.org/10.1214/20-AOAS1332

RUFFIEUX, H., FAIRFAX, B. P., NASSIRI, I., VIGORITO, E., WALLACE, C., RICHARDSON, S. and BOTTOLO, L. (2021). EPISPOT: An epigenome-driven approach for detecting and interpreting hotspots in molecular QTL studies. *Am. J. Hum. Genet*. **108** 983–1000. https://doi.org/10.1016/j.ajhg.2021.04.010

SCOTT, J. G. and BERGER, J. O. (2010). Bayes and empirical-Bayes multiplicity adjustment in the variable-selection problem. *Ann. Statist*. **38** 2587–2619. MR2722450 https://doi.org/10.1214/10-AOS792

SOMEREN, E. V., WESSELS, L. F., BACKER, E. and REINDERS, M. J. (2002). Genetic network modeling. *Pharmacogenomics J*. **3** 507–525.

R CORE TEAM (2013). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

VAN DE WIEL, M. A., TE BEEST, D. E. and MÜNCH, M. M. (2019). Learning from a lot: Empirical Bayes for high-dimensional model-based prediction. *Scand. J. Stat*. **46** 2–25. MR3915265 https://doi.org/10.1111/sjos.12335

VAN DER PAS, S. L., KLEIJN, B. J. K. and VAN DER VAART, A. W. (2014). The horseshoe estimator: Posterior concentration around nearly black vectors. *Electron. J. Stat*. **8** 2585–2618. MR3285877 https://doi.org/10.1214/14-EJS962

WANG, H. (2012). Bayesian graphical lasso models and efficient posterior computation. *Bayesian Anal*. **7** 867–886. MR3000017 https://doi.org/10.1214/12-BA729

WANG, H. (2015). Scaling it up: Stochastic search structure learning in graphical models. *Bayesian Anal*. **10** 351–377. MR3420886 https://doi.org/10.1214/14-BA916

WANG, L., HUANG, Y., WANG, X. and CHEN, Y. (2019). Label-free LC-MS/MS proteomics analyses reveal proteomic changes accompanying MSTN KO in C2C12 cells. *BioMed Res. Int*. **2019**.

YANG, X., GAN, L., NARISETTY, N. N. and LIANG, F. (2021). GemBag: Group estimation of multiple Bayesian graphical models. *J. Mach. Learn. Res*. **22** 54. MR4253747

YAO, C., JOEHANES, R., JOHNSON, A. D., HUAN, T., LIU, C., FREEDMAN, J. E., MUNSON, P. J., HILL, D. E., VIDAL, M. et al. (2017). Dynamic role of trans regulation of gene expression in relation to complex traits. *Am. J. Hum. Genet*. **100** 571–580.