Statistics and Computing (2025) 35:179
https://doi.org/10.1007/s11222-025-10709-4

ORIGINAL PAPER

Outcome-guided spike-and-slab Lasso Biclustering: A Novel Approach for Enhancing Biclustering Techniques for Gene Expression Analysis

Luis A. Vargas-Mieles[1] · Paul D. W. Kirk[1,2] · Chris Wallace[1,2]

Received: 10 December 2024 / Accepted: 10 August 2025

Abstract
Biclustering has gained interest in gene expression data analysis due to its ability to identify groups of samples that exhibit similar behaviour in specific subsets of genes (or vice versa), in contrast to traditional clustering methods that classify samples based on all genes. Despite advances, biclustering remains a challenging problem, even with cutting-edge methodologies. This paper introduces an extension of the recently proposed Spike-and-Slab Lasso Biclustering (SSLB) algorithm, termed Outcome-Guided SSLB (OG-SSLB), aimed at enhancing the identification of biclusters in gene expression analysis. Our proposed approach integrates disease outcomes into the biclustering framework through Bayesian profile regression. By leveraging additional clinical information, OG-SSLB improves the interpretability and relevance of the resulting biclusters. Comprehensive simulations and numerical experiments demonstrate that OG-SSLB achieves superior performance, with improved accuracy in estimating the number of clusters and higher consensus scores compared to the original SSLB method. Furthermore, OG-SSLB effectively identifies meaningful patterns and associations between gene expression profiles and disease states. These promising results demonstrate the effectiveness of OG-SSLB in advancing biclustering techniques, providing a powerful tool for uncovering biologically relevant insights. The OGSSLB software can be found as an R/C++ package at https://github.com/luisvargasmieles/OGSSLB.

Keywords Biclustering · Factor analysis · Profile regression · Spike-and-Slab Lasso

1 Introduction

Over the last few decades, the identification of groups that share interesting common characteristics has been a key objective in various real-world applications. Clustering has proven to be a crucial method for discovering these groups that exhibit patterns within high-dimensional data, particularly in the context of large omics datasets (Chauvel et al. 2019). This technique enables the detection of associations between related entities based on shared features or attributes.

One domain of omics in which clustering techniques have been widely employed has been the examination of transcriptomic data, which captures patterns of gene expression levels within biological entities such as tissues or cells (Oyelade et al. 2016; Saelens et al. 2018). The need to understand shared transcriptional patterns embedded in gene expression data has led to extensive development of clustering methodologies.

Although clustering has been beneficial in revealing hidden patterns within these large-scale datasets, it is not without drawbacks. Among its disadvantages, traditional clustering models assume that samples within a cluster behave similarly across all genes and vice versa. Additionally, clustering often results in a partition of the samples or genes into disjoint subsets. These assumptions may oversimplify the biological system under analysis.

Owing to these limitations, biclustering, a methodology that clusters genes and samples simultaneously, has gained

P. D. W. Kirk, C. Wallace.: These authors contributed equally to this work.

✉ Luis A. Vargas-Mieles
lv375@cam.ac.uk

Paul D. W. Kirk
paul.kirk@mrc-bsu.cam.ac.uk

Chris Wallace
cew54@cam.ac.uk

[1] Cambridge Institute of Therapeutic Immunology and Infectious Disease (CITIID), University of Cambridge, Cambridge, UK

[2] MRC Biostatistics Unit (BSU), University of Cambridge, Cambridge, UK

Published online: 28 August 2025

more attention in recent years (Xie et al. 2018, 2019; Wang et al. 2021; Gong et al. 2024). This approach allows flexibility in capturing subsets of genes that may behave differently across conditions or subsets of samples that differ according to specific sets of features. Furthermore, biclustering permits overlapping patterns, where genes or samples may belong to multiple biclusters, which more closely matches biological systems. For instance, samples may cluster by sex, disease age and disease state simultaneously, and a single gene may be a member of two or more biological pathways.

Several approaches have been proposed to estimate these subgroups of genes and samples, and various reviews of the biclustering methods developed over the past decades exist in the literature (see, e.g., Eren et al. (2012); Padilha and Campello (2017)). Building upon the results of Nicholls and Wallace (2021), this work focuses on the adoption of a multiplicative model, which has proven advantageous in this context. Such models have demonstrated effective capture of diverse sources of variability in gene expression data, including the presence of outlier genes or genes with fluctuating expression levels (Hochreiter et al. 2010).

Among the existing algorithms using this methodology, we highlight three: Factor analysis for bicluster acquisition (FABIA) (Hochreiter et al. 2010), the BicMix biclustering method (Gao et al. 2016) and Spike-and-Slab Lasso Biclustering (SSLB) (Moran and George 2021). All three are based on a Bayesian factor analysis model with sparsity-inducing priors that have proven to possess a notable ability to recover latent structures in gene expression data. However, a comparative study by Nicholls and Wallace (2021) highlights that SSLB has the advantage of allowing different sparsity levels on each bicluster, in comparison to BicMix, which allows only two levels of sparsity (sparse or dense) for each bicluster, and FABIA, which uses the same sparsity level for all biclusters. Furthermore, while FABIA requires setting the number of biclusters in advance, SSLB (and BicMix) automatically estimates the number of biclusters.

Although recent approaches, such as the ones mentioned above, have been shown to be capable of revealing these latent structures within gene expression data, biclustering, in general, is recognised as an NP-hard problem (Tanay et al. 2002; Peeters 2003). The NP-hardness arises from the challenge of simultaneously grouping rows and columns of a matrix to identify coherent submatrices while considering various constraints and optimisation criteria. Added to the fact that biclusters can also overlap, these difficulties pose a substantial challenge to even the most state-of-the-art methods, further complicating the accurate identification of samples and gene groups that share a common characteristic.

One promising approach to mitigate this complexity is the integration of informative outcome data into the clustering process, thereby guiding the inference towards biologically relevant clustering structures. Several outcome-guided clustering methods have been developed in recent decades, with applications in K-Means clustering (Meng et al. 2022) and gene selection based on survival data (Koestler et al. 2010), to mention a few. For additional insights, see Bair (2013).

In light of these developments, Bayesian profile regression has emerged as another outcome-guided, semi-supervised method for clustering that leverages an outcome variable to inform cluster allocations (Molitor et al. 2010; Liverani et al. 2015). Unlike some of the previously mentioned approaches, it offers a fully model-based framework that can handle a variety of outcome types, making it more versatile. This approach has already shown success in handling binary covariate data (Beall et al. 2024) as well as longitudinal or multivariate continuous outcomes (Rouanet et al. 2023), making it a valuable tool in the context of gene expression analysis.

Building on this success, we explore whether such outcome-guided strategies can also enhance biclustering, where the goal is to simultaneously group genes and samples. Since most gene expression studies also include phenotype information such as age, sex, and disease status, we investigate in this work whether integrating disease outcomes would enhance cluster consistency within specific disease groups. To achieve this, we introduce an outcome-guided version of SSLB by incorporating the disease outcome of the samples into the model via Bayesian profile regression, aiming to better guide the biclustering membership of genes and samples. As we show in numerical and real-data experiments, this integration enhances the accuracy of the SSLB model and refines the biological relevance of the estimated biclusters.

The remainder of the paper is organised as follows. Section 2 discusses the current state of biclustering models, particularly within the context of factor analysis models, and explains the SSLB model that we aim to improve. Section 3 highlights the potential contributions of additional data available in most gene expression studies and presents the methodology used in our work, which integrates Bayesian profile regression into the SSLB model, detailing the computations added to implement this new approach. Section 4 details the results of our experiments and provides a comprehensive analysis of the findings. Finally, Section 5 concludes by summarising the key contributions of our research and suggesting directions for future work.

2 Factor Analysis Models and Current Biclustering Techniques

Before presenting the full mathematical formulation, we briefly summarise the modelling approach underlying our work. Our goal is to detect biclusters (groups of genes and patients that co-vary) by leveraging a sparse factor analysis model. In this framework, gene expression data is decom-

posed into two low-rank matrices capturing patient and gene contributions to each bicluster. We then focus on the Spike-and-Slab Lasso Biclustering (SSLB) method, which uses sparsity-inducing priors to identify interpretable biclusters and forms the basis for the methodological extension we propose in this work.

We assume that gene expression data is represented in a matrix $\mathbf{X} \in \mathbb{R}^{N\times G}$ with N samples and G features or genes. Additionally, we assume that the data $\mathbf{X}$ contains K non-disjoint latent groups of genes and samples that potentially may be linked due to some common biological characteristics. The problem of determining the number of subgroups K, as well as identifying the genes and samples belonging to each of these groups, is defined as "biclustering".

Following the results of Moran and George (2021), we adopt a factor analysis model to identify these latent groups or biclusters, where $\mathbf{X}$ can be represented as

$$\mathbf{X} = \mathbf{Z}\Lambda^{\mathsf{T}} + \mathbf{E}, \tag{1}$$

where

- $\mathbf{Z} \in \mathbb{R}^{N\times K}$, which will be called the sample loading matrix,
- $\Lambda \in \mathbb{R}^{G\times K}$, which will be called the gene loading matrix, and
- $\mathbf{E} = [\varepsilon_1, \ldots, \varepsilon_N]^{\mathsf{T}} \in \mathbb{R}^{N\times G}$ is noise, where each $\varepsilon_i \sim N_G(\mathbf{0}, \Sigma)$, and $\Sigma = \text{diag}\{\sigma_j^2\}_{j=1}^G$.

Current biclustering algorithms that use the model given in (1) implement an unsupervised approach to identify sparse groups of relevant genes and samples and thus infer $\mathbf{Z}$ and $\mathbf{\Lambda}$. Their main input is $\mathbf{X}$ without additional information (apart from the necessary model parameters) included in the model. For instance, FABIA (Hochreiter et al. 2010) uses a Laplacian prior on all $\mathbf{Z}$ and $\mathbf{\Lambda}$ entries to induce sparsity, applying the same prior to every entry in these matrices. BicMix (Gao et al. 2016), on the other hand, allows the columns of $\mathbf{Z}$ and $\mathbf{\Lambda}$ to be sparse or dense. For the sparse components, it utilises three levels of shrinkage, each employing a three-parameter beta (TPB) prior (Armagan et al. 2011), to promote sparsity.

Finally, SSLB employs the Spike-and-Slab Lasso prior (Ročková and George 2018) for both $\mathbf{Z}$ and $\mathbf{\Lambda}$. This prior enables stronger regularisation on near-zero coefficients (in the spike) to achieve sparsity, while applying weaker regularisation on larger coefficients (in the slab) to maintain accuracy. A key advantage of the SSLB prior over FABIA and BicMix is its ability to allow varying levels of sparsity for each bicluster. As mentioned earlier, FABIA uses the same prior for all biclusters, and BicMix permits only two sparsity levels ('sparse' or 'dense'). SSLB, however, assigns a distinct sparsity parameter to each bicluster.

2.1 The SSLB model

For completeness, we provide a brief explanation of each component within this Bayesian model. See Moran and George (2021) for more details.

2.1.1 SSLB likelihood

Since (1) can also be written as

$$\mathbf{X} = \sum_{k=1}^{K} \mathbf{z}^k \boldsymbol{\lambda}^{k\mathsf{T}} + \mathbf{E},$$

where the superscript $\mathbf{z}^k$ represents the kth column of $\mathbf{Z}$, the likelihood is defined as

$$p(\mathbf{X} \mid \mathbf{Z}, \mathbf{\Lambda}) \propto \prod_{i=1}^{N} \left\{ \exp\left[-0.5 \left(\mathbf{x}_i - \mathbf{Z}_i \boldsymbol{\lambda}^{\mathsf{T}}\right)^{\mathsf{T}} \mathbf{\Sigma}^{-1} \left(\mathbf{x}_i - \mathbf{Z}_i \boldsymbol{\lambda}^{\mathsf{T}}\right) \right] \left(\prod_{j=1}^{G} \sigma_j^2 \right)^{-1/2} \right\},$$

where the subscript $\mathbf{Z}_i$ refers to the ith row of $\mathbf{Z}$[1].

2.1.2 SSLB priors

Prior on the elements of $\mathbf{\Lambda}$For the elements of the gene loading matrix, we have a spike-and-slab prior (Ročková and George 2018), defined by

$$p(\mathbf{\Lambda} \mid \mathbf{\Gamma}, \omega_0, \omega_1) \propto \prod_{j=1}^{G} \prod_{k=1}^{K} \left[(1-\gamma_{jk})\, \omega_0 \exp\left(-\omega_0 \left|\lambda_{jk}\right|\right) + \gamma_{jk} \omega_1 \exp\left(-\omega_1 \left|\lambda_{jk}\right|\right) \right],$$

where $\mathbf{\Gamma} = \{\gamma_{jk}\}_{j,k=1}^{G,K}$ are binary indicator variables that specify if feature j is active in bicluster k. Depending on γ_{jk}, each λ_{jk} can be drawn from either a Laplacian "spike" characterised by a large parameter value ω_0 and is consequently negligible, or from a Laplacian "slab" with a small parameter ω_1 and, consequently, can be large. Refer to Section 2.1.4 for detailed information on the values of ω_0 and ω_1.

Prior on the gene binary indicator variable $\mathbf{\Gamma}$o estimate each $\{\gamma_{jk}\}_{j,k=1}^{G,K}$, the authors use the Beta-Bernoulli prior

[1] Throughout, this superscript and subscript notation is utilised to denote the respective column and row vector of a matrix.

$$p(\boldsymbol{\Gamma} \mid \boldsymbol{\Theta}, \alpha) \propto \prod_{j=1}^{G} \prod_{k=1}^{K} \theta_k^{\gamma_{jk}+\alpha-1} (1-\theta_k)^{1-\gamma_{jk}},$$

where

- $\boldsymbol{\Theta} = \{\theta_1, \ldots, \theta_K\}$,
- $\gamma_{jk} \mid \theta_k \sim \text{Bernoulli}(\theta_k)$,
- $\theta_k \sim \text{Beta}(\alpha, 1)$.

For this prior, Moran and George (2021) recommends a finite approximation of the Indian buffet process (IBP) prior using $\alpha = 1/K$. When $K \to \infty$, this prior is the IBP prior. See Ghahramani and Griffiths (2005) for details.

Prior on the elements of $\mathbf{Z}$ For the elements of the sample loading matrix, the authors proposed an alternate formulation of the Spike-and-Slab Lasso prior previously defined for the gene loading matrix, for computational purposes. Firstly, an auxiliary variable $\{\tau_{ik}\}_{i,k=1}^{N,K}$ is introduced in the model, such as

$$z_{ik} \mid \tau_{ik} \sim N(0, \tau_{ik}), \tag{2}$$

and then, for each τ_{ik}, a mixture of exponentials is defined as

$$p(\mathbf{T} \mid \tilde{\boldsymbol{\Gamma}}, \tilde{\omega}_0, \tilde{\omega}_1) \propto \prod_{i=1}^{N} \prod_{k=1}^{K} \Big[(1-\tilde{\gamma}_{ik})\, \tilde{\omega}_0 \exp(-0.5\tilde{\omega}_0 \tau_{ik}) + \tilde{\gamma}_{ik} \tilde{\omega}_1 \exp(-0.5\tilde{\omega}_1 \tau_{ik}) \Big], \tag{3}$$

where $\tilde{\boldsymbol{\Gamma}} = \{\tilde{\gamma}_{ik}\}_{i,k=1}^{N,K}$ are binary indicator variables indicating bicluster membership on the elements of z_{ik}, and $\mathbf{T} = \{\tau_{ik}\}_{i,k=1}^{N,K}$ are the covariances of z_{ik}. In summary, the authors represent the Laplace distribution as a scale mixture of a normal with an exponential mixing density: a spike-and-slab Lasso prior on each z_{ik} by introducing auxiliary variables τ_{ik} for the variance of every z_{ik}, and then each τ_{ik} is assigned a mixture of exponentials (spike-and-slab) priors. Marginalising over the τ_{ik} yields the usual spike-and-slab Lasso prior.

Prior on the sample indicator variable $\tilde{\boldsymbol{\Gamma}}$

For this variable, the authors proposed an Indian Buffet Process (IBP) prior with an optional Pitman-Yor (PY) extension prior (Teh et al. 2007), defined as

$$\begin{aligned} \tilde{\gamma}_{ik} &\sim \text{Bernoulli}(\tilde{\theta}_{(k)}) \\ \tilde{\theta}_{(k)} &= \prod_{l=1}^{k} \nu_{(l)} \\ \nu_{(l)} &\sim \text{Beta}(\tilde{\alpha} + ld, 1-d), \quad \text{where } d \in [0,1),\ \tilde{\alpha} > -d. \end{aligned} \tag{4}$$

When $0 < d < 1$, the above formulation corresponds to the Pitman-Yor IBP prior. In the case where $d = 0$, it represents the standard IBP prior. For the simulations carried out in the SSLB paper and for consistency in this work, the finite approximation to the IBP is also used for comparison, which involves a Beta prior on the sparsity weights, $\tilde{\theta}_k \sim \text{Beta}(\tilde{a}, \tilde{b})$ where $\tilde{a} \propto 1/K$ and $\tilde{b} = 1$. See Teh et al. (2007) for further details.

Prior on the covariance matrix $\boldsymbol{\Sigma}$ *of* ε_i

For the covariance, $\boldsymbol{\Sigma}$, of the vectors ε_i that define $\mathbf{E}$ in (1), an inverse gamma prior was assumed. That is

$$p(\boldsymbol{\Sigma} \mid \eta, \xi) \propto \prod_{j=1}^{G} \left[\left(\sigma_j^2\right)^{-(\eta/2+1)} \exp\left(\frac{-\eta\xi}{2\sigma_j^2}\right) \right],$$

where the SSLB authors suggest setting $\eta = 3$ and choosing ξ such that the 95% quantile of the prior on $\{\sigma_j^2\}_{j=1}^{G}$ matches the sample column variance $\{s_j^2\}_{j=1}^{G}$, i.e., $p(\sigma_j < s_j) = 0.95$. Refer to (Chipman et al. 2010, Section 2.2.4) and (Moran and George 2021, Section 2.5) for further information.

After explaining the whole hierarchical structure of the SSLB model, we first provide a schematic overview of the SSLB model in Figure 1, highlighting its main components and structure. Then, for a more detailed representation of the variable dependencies, we present the corresponding Directed Acyclic Graph (DAG) in Figure 2.

2.1.3 Estimation of biclusters: EM algorithm

To proceed with the estimation of the parameters of interest in the SSLB model, the authors implemented an Expectation Maximisation (EM) algorithm. For completeness, we are going to briefly describe the important parts of this procedure for the IBP prior case for $\tilde{\boldsymbol{\Gamma}}$. See (Moran and George 2021, Section 2.3) and its supplementary material for more details.

E step At iteration $t+1$ of the EM algorithm, the E step involves the computation of the following expectation

$$Q(\boldsymbol{\Delta} \mid \boldsymbol{\Delta}^{(t)}) = \mathbb{E}_{\mathbf{Z}, \tilde{\Gamma} \mid \Delta^{(t)}, \mathbf{X}}[\log p(\boldsymbol{\Delta}, \mathbf{Z}, \tilde{\boldsymbol{\Gamma}} \mid \mathbf{X})],$$

where $\boldsymbol{\Delta} = \{\boldsymbol{\Lambda}, \boldsymbol{\Sigma}, \mathbf{T}, \nu\}$ are the variables at which $Q(\boldsymbol{\Delta} \mid \boldsymbol{\Delta}^{(t)})$ will be maximised in the M step. See (A1) and (A2) for more details.

M step In this stage, the following is calculated

$$\boldsymbol{\Delta}^{(t+1)} = \arg\max_{\Delta} Q(\boldsymbol{\Delta} \mid \boldsymbol{\Delta}^{(t)}).$$

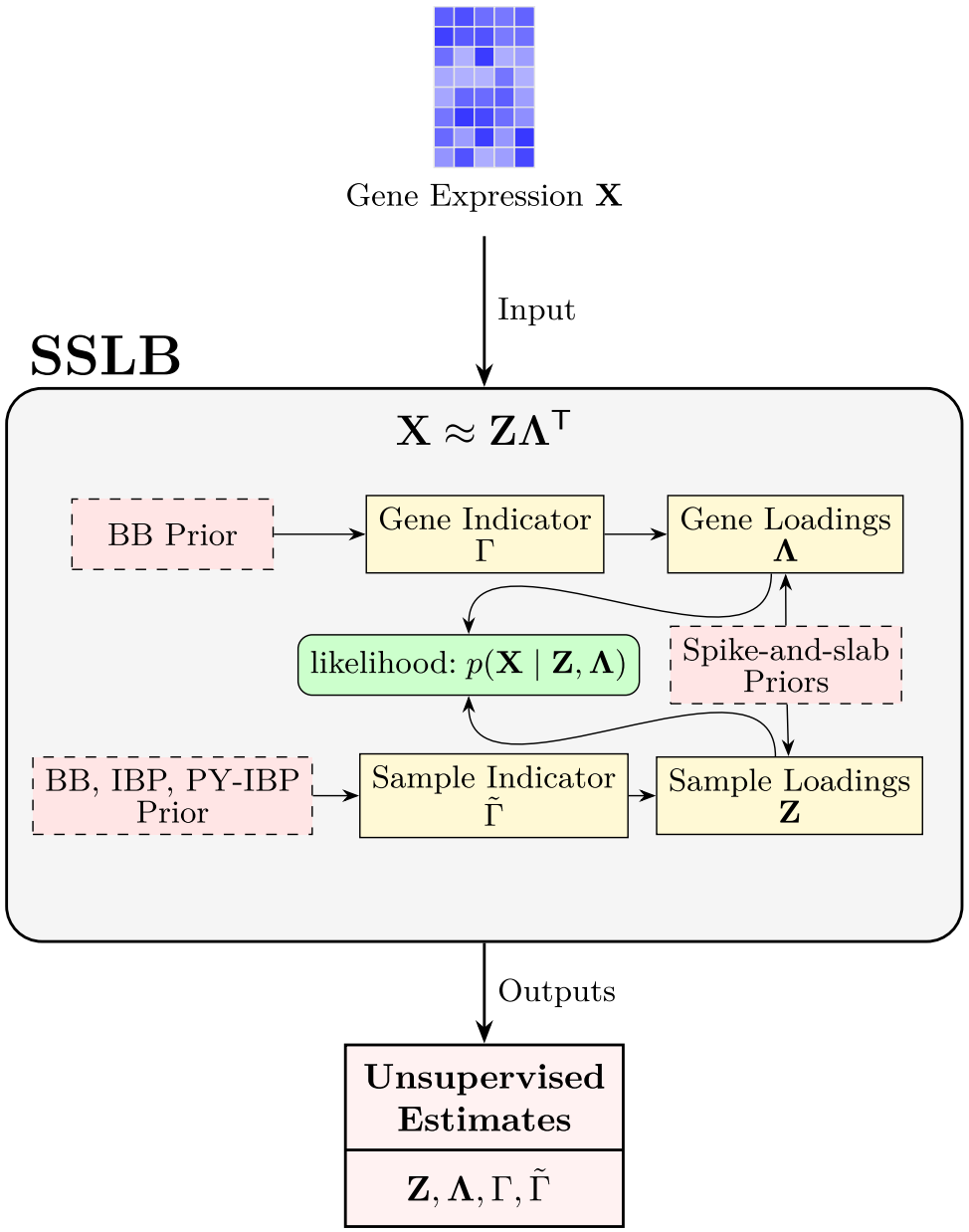

Fig. 1 Overview of SSLB. The gene expression matrix $\mathbf{X}$ is modelled using latent sample and gene loadings, $\mathbf{Z}$ and Λ, governed by binary indicators $\tilde{\Gamma}$ and Γ, respectively. Sparsity is induced via spike-and-slab priors, and nonparametric priors are placed on the indicator matrices. The main elements of the model are shown; some components (such as noise variance priors) are omitted for clarity.

2.1.4 Implementation of SSLB & initial conditions

The SSLB algorithm employs the previously described EM algorithm combined with a dynamic posterior exploration approach to estimate $\mathbf{\Lambda}$. This involves a gradual increase of the spike parameter ω_0 through a sequence of values, while the slab parameter ω_1 is kept fixed. This strategy helps stabilise large coefficients and progressively thresholds negligible coefficients to zero (see (Moran and George 2021, Section 2.3) for more details).

The SSLB algorithm is initialised with entries of $\mathbf{\Lambda}$ generated independently from a standard normal distribution. The entries of $\mathbf{T}$, the matrix of auxiliary variance parameters, are set to 100, representing an initial relatively non-informative prior to $\mathbf{Z}$. The sparsity weights, θ_k, are initialised at 0.5. The IBP parameters, $\boldsymbol{\nu}$, are generated independently from a Beta(1, 1) distribution and then ordered from largest to smallest. In real-world applications, the recommended initialisation for K, the number of biclusters, is set to $K_{init} = 50$. See (Moran and George 2021, Section 2.5) for more details.

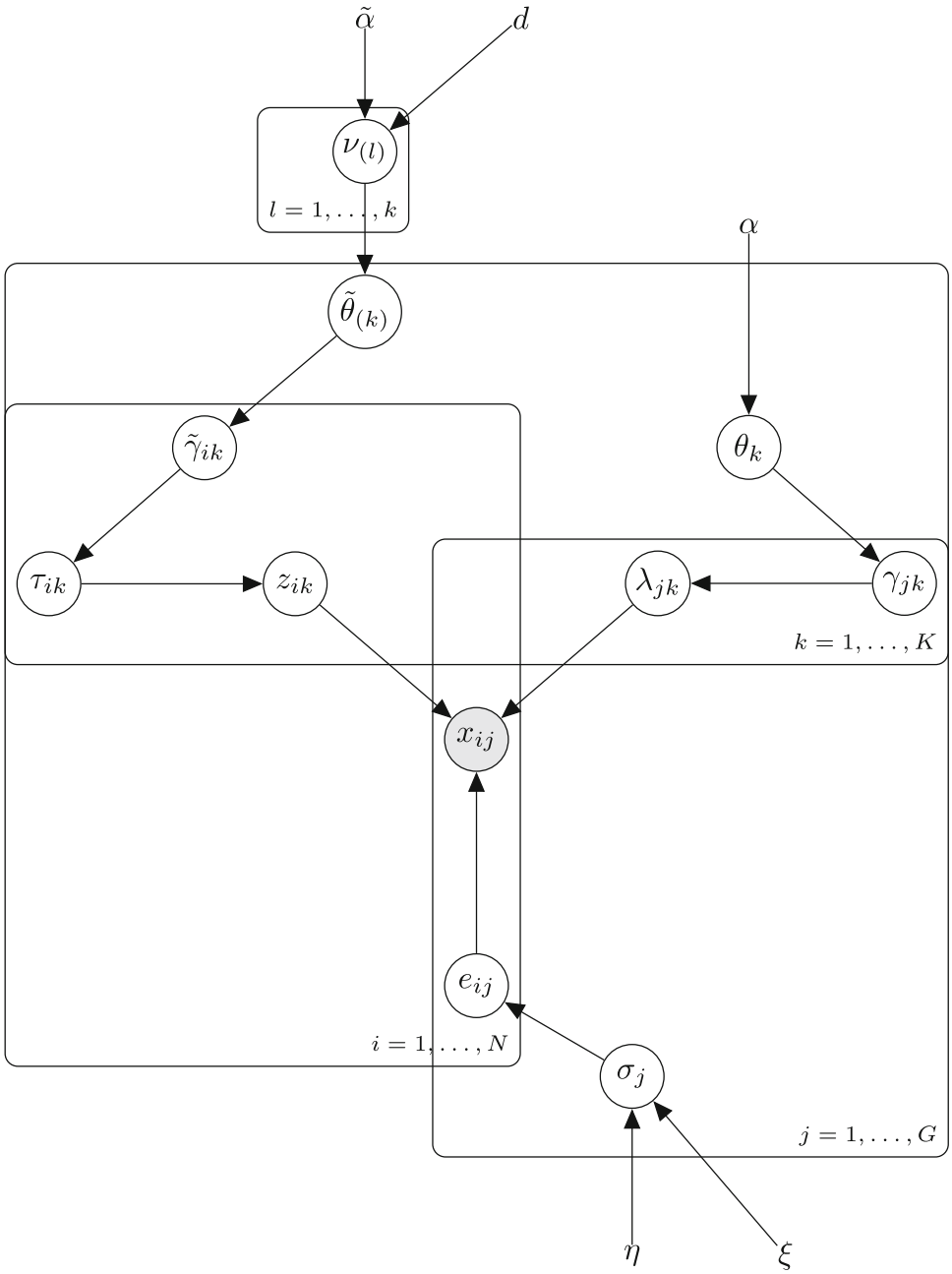

Fig. 2 DAG for the SSLB-IBP model, where the indices $i = 1, \ldots, N$ correspond to the N samples, the indices $j = 1, \ldots, G$ correspond to the G genes, and the indices $k = 1, \ldots, K$ represent the K biclusters. Variables x_{ij} are observed and correspond to the gene expression data.

3 Methodology

Considering the potential availability of additional data on sampled individuals, such as disease status, our objective is to investigate whether incorporating this information can enhance the complex task of biclustering. Notably, to the best of our knowledge, no existing biclustering method has explored the inclusion of the disease status of samples within the model.

We propose incorporating an outcome variable $\mathbf{Y} \in \{0, 1\}^{N \times C}$ into the current SSLB model, which will correspond to the presence or absence (i.e., 1 or 0, respectively) of disease $c \in \{1, \ldots, C\}$ in the sample $i \in \{1, \ldots, N\}$.

We assume we have gene expression data $\mathbf{X}$ also modelled as (1), and outcomes $\mathbf{Y}$. Since $\mathbf{Y}$ provides only sample-wise information, it will only affect the distribution related to the sample loading matrix $\mathbf{Z}$. The general model considered in Molitor et al. (2010) and adapted for the biclustering problem with respect to $\mathbf{\Lambda}$ is defined as:

$$p\left(\mathbf{Z}, \mathbf{Y} \mid \mathbf{\Theta}_{\mathbf{Z}}, \mathbf{\Theta}_{\mathbf{Y}}\right) = \prod_{i=1}^{N} p(\mathbf{y}_i \mid \boldsymbol{\theta}_{\mathbf{y}_i}, \mathbf{z}_i) p(\mathbf{z_i} \mid \boldsymbol{\theta}_{\mathbf{z}_i}),$$

where $\boldsymbol{\theta}_{\mathbf{z}_i}$ represents the parameters of the model for $\mathbf{z}_i$, and $\boldsymbol{\theta}_{\mathbf{y}_i}$ represents the parameters of the model for $\mathbf{y}_i$. Furthermore, in the profile regression setting, the factor loadings $\mathbf{Z}$ and the outcome $\mathbf{Y}$ are conditionally independent because their relationship is mediated by the binary indicator matrix $\tilde{\mathbf{\Gamma}}$, which governs the structure of the sampling clustering. $\tilde{\mathbf{\Gamma}}$ determines which latent factors contribute to $\mathbf{Z}$ and how they align with $\mathbf{Y}$. Therefore, the profile regression model becomes:

$$p\left(\mathbf{Z}, \mathbf{Y} \mid \mathbf{\Theta}_{\mathbf{Z}}, \mathbf{\Theta}_{\mathbf{Y}}\right) = \prod_{i=1}^{N} p(\mathbf{y}_i \mid \boldsymbol{\theta}_{\mathbf{y}_i}) p(\mathbf{z}_i \mid \boldsymbol{\theta}_{\mathbf{z}_i}),$$

where

- $\mathbf{\Theta}_{\mathbf{Z}} = \{\tilde{\mathbf{\Gamma}}, \mathbf{T}, \nu, \widetilde{\omega_0}, \widetilde{\omega_1}, \tilde{\boldsymbol{\theta}}, \tilde{\alpha}\}$.
- $p\left(\mathbf{Z} \mid \mathbf{\Theta}_{\mathbf{Z}}\right) \propto p(\mathbf{Z} \mid \mathbf{T}) p(\mathbf{T} \mid \tilde{\mathbf{\Gamma}}, \widetilde{\omega_0}, \widetilde{\omega_1}) p(\tilde{\mathbf{\Gamma}} \mid \tilde{\boldsymbol{\theta}}, \tilde{\alpha})$.

and $\mathbf{\Theta}_{\mathbf{Y}}$ is the set of bicluster-specific parameters of the model for $\mathbf{Y}$, which also includes $\tilde{\mathbf{\Gamma}}$. Note that each of the probability density functions given in $p\left(\mathbf{Z} \mid \mathbf{\Theta}_{\mathbf{Z}}\right)$ is already defined for the current SSLB model in (2), (3), and (4).

To model the disease outcome $\mathbf{Y}$, we adopt a multinomial logistic regression approach, where the latent bicluster memberships $\widetilde{\gamma}_{ik}$ serve as covariates. The model can be interpreted through the log-odds of a disease $c \in \{1, \ldots, C-1\}$ relative to a reference disease category C, given by:

$$\log \frac{P(y_{ic}=1)}{P(y_{iC}=1)} = \sum_{k=1}^{K} w_{ck} \widetilde{\gamma}_{ik},$$

where w_{ck} denotes the contribution of bicluster k to the log-odds of disease c. This formulation makes it explicit that if sample i belongs to bicluster k, it contributes additively to the risk of disease c. Unlike standard Bayesian profile regression, our model allows overlapping bicluster membership–that is, each sample can belong to multiple biclusters–providing greater flexibility in capturing complex disease-related structures in gene expression data.

We formalise this log-odds formulation in the full likelihood expression as:

$$p\left(\mathbf{Y} \mid \mathbf{\Theta}_{\mathbf{Y}}\right) \propto \prod_{i=1}^{N} \prod_{l=1}^{C} \left[\exp\left(\mathbf{w}^{(l)\,T} \tilde{\boldsymbol{\gamma}}_i'\right) / \sum_{l'=1}^{c} \exp\left(\mathbf{w}^{(l')\,T} \tilde{\boldsymbol{\gamma}}_i'\right) \right]^{y_{il}} + \exp\left(-\frac{1}{2}\zeta_{\mathrm{w}} \|\mathbf{W}\|_F^2\right), \tag{5}$$

where

- $\mathbf{\Theta}_{\mathbf{Y}} = \{\tilde{\mathbf{\Gamma}}', \mathbf{W}, \zeta_{\mathrm{w}}\}$.
- y_{il} corresponds to the presence/absence (i.e., 1 or 0) of the l disease in the i sample.
- C is the number of diseases (i.e., the number of categories in the multinomial logistic regression model) in the study.
- $\mathbf{W} \in \mathbb{R}^{(K+1)\times C}$ is the matrix of weights of the multinomial logistic regression, which includes the bias coefficients. It is the main element that allows $\mathbf{Y}$ to assist in the task of assigning samples (and genes) to biclusters.
- $\tilde{\boldsymbol{\gamma}}_i' = \left[1, \tilde{\gamma}_{i1}, \ldots, \tilde{\gamma}_{iK}\right]$.
- A reference category (e.g., class or disease C) needs to be chosen such that the column of the matrix $\mathbf{W}$ corresponding to the selected reference category has only zeros (e.g., $\mathbf{w}_C = [0, \ldots, 0]$).
- To avoid overfitting, we have introduced an ℓ_2 regularisation term for the weight matrix $\mathbf{W}$ with regularisation hyperparameter $\zeta_{\mathrm{w}} \in \mathbb{R}^{+}$.

By incorporating this approach into the model, the complete log posterior (A1) and the expression associated with the E step (A2) will undergo minor modifications. A schematic representation of the OG-SSLB model structure, including its outcome-guided component, is shown in Figure 3. See also (B3) and (B4) in the Appendix for details.

3.1 Adapted EM algorithm for OG-SSLB

To estimate the parameters of the OG-SSLB model, we adapt the Expectation-Maximisation (EM) procedure originally proposed for SSLB. The main difference lies in the incorporation of the disease outcome variable $\mathbf{Y}$, and the regression parameters $\mathbf{W}$ and ζ_{w}. Below we summarise the updated steps.

E step At iteration $t+1$, the E step involves computing:

$$Q(\mathbf{\Delta} \mid \mathbf{\Delta}^{(t)}) = \mathbb{E}_{\mathbf{Z}, \tilde{\mathbf{\Gamma}} \mid \mathbf{\Delta}^{(t)}, \mathbf{X}, \mathbf{Y}} \left[\log p(\mathbf{\Delta}, \mathbf{Z}, \tilde{\mathbf{\Gamma}} \mid \mathbf{X}, \mathbf{Y})\right],$$

where $\mathbf{\Delta} = \{\mathbf{\Lambda}, \mathbf{\Sigma}, \mathbf{T}, \nu, \mathbf{W}, \zeta_{\mathrm{w}}\}$. Compared to SSLB, this step now also includes:

- estimation of the expectations $\langle \widetilde{\gamma}_{ik} \rangle$ using the joint information from $\mathbf{X}$ and $\mathbf{Y}$,
- estimation of the expectation of the log-sum-exp term from the multinomial logistic regression likelihood.

M step In the M step, we maximise:

$$\mathbf{\Delta}^{(t+1)} = \arg\max_{\mathbf{\Delta}} Q(\mathbf{\Delta} \mid \mathbf{\Delta}^{(t)}),$$

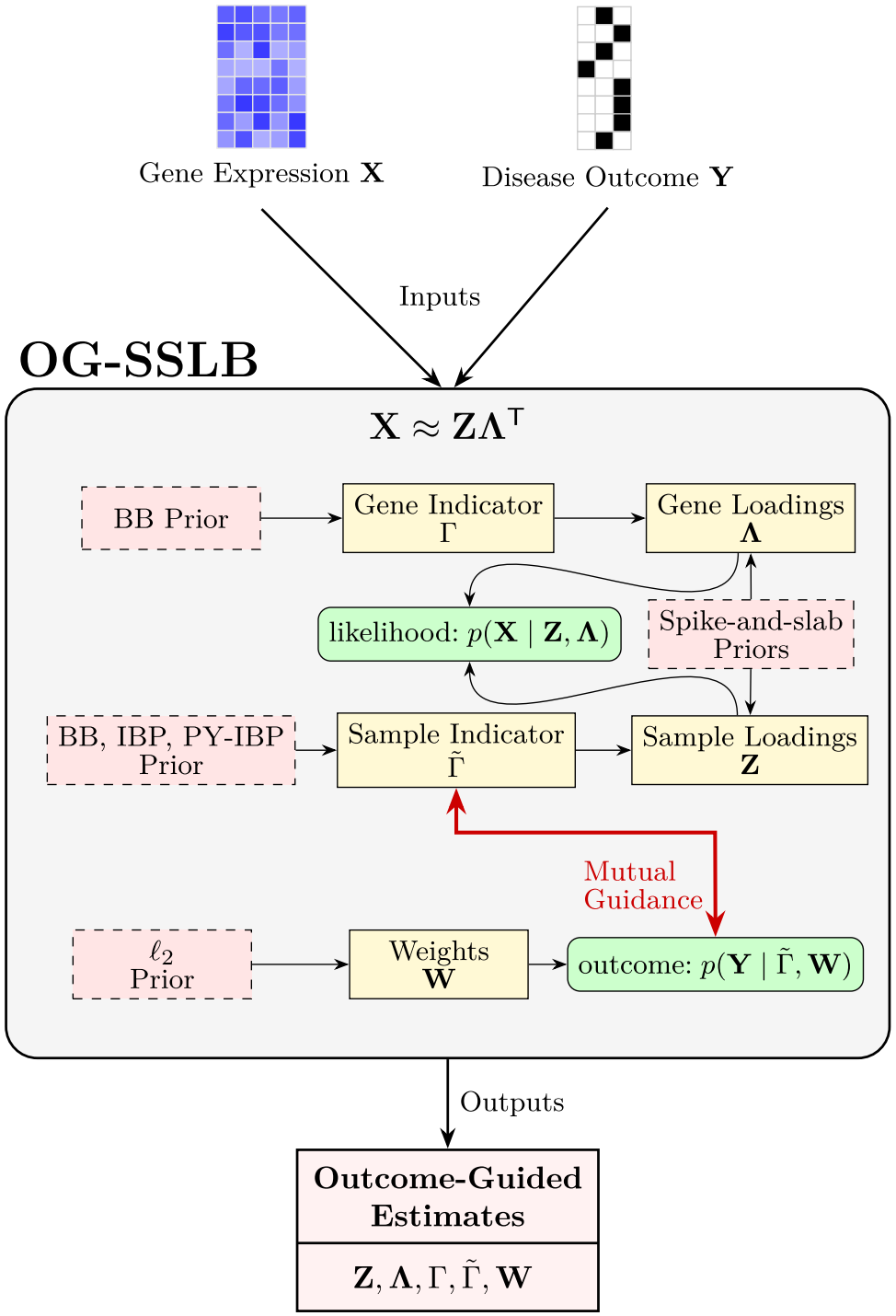

Fig. 3 Overview of OG-SSLB. The model extends SSLB by incorporating an outcome variable $\mathbf{Y}$, linked to the latent structure via the sample indicator matrix $\tilde{\Gamma}$ and regression weights $\mathbf{W}$. The outcome model $p(\mathbf{Y} \mid \tilde{\Gamma}, \mathbf{W})$ plays a dual role: it uses $\tilde{\Gamma}$ as input and, simultaneously, guides its inference. The red bidirectional arrow illustrates this mutual interaction. For clarity, auxiliary components such as noise priors are omitted.

which includes updating the standard SSLB parameters $\{\mathbf{\Lambda}, \mathbf{\Sigma}, \mathbf{T}, \nu\}$ and additionally estimating $\mathbf{W}$ and ζ_w via a regularised multinomial logistic regression.

Below we will explain the computation of the new expectation and maximisation steps introduced by the profile regression model adapted to the SSLB model, particularly the computation of $\langle \widetilde{\boldsymbol{\gamma}}_i' \rangle$, the second last term of (B4) that involves the computation of an expectation of a log-sum-exp expression, the estimation of ζ_w and the maximisation of $\mathbf{W}$. See (Moran and George 2021, Section 2.3) and its supplementary material for details on the computation of the remaining parameters that are not affected by the introduction of profile regression to the SSLB model.

3.2 Expectation of $\tilde{\gamma}_i$

For the expectation of the $\widetilde{\gamma}_{ik}$ variables (i.e., the binary membership indicator of sample i in bicluster k), we have

$$\begin{aligned}\langle\widetilde{\gamma}_{ik}\rangle &= p\left(\widetilde{\gamma}_{ik}=1 \mid \mathbf{Y},\mathbf{T},\widetilde{\boldsymbol{\theta}},\mathbf{W}\right)\\ &= \frac{1}{1+\frac{p(y_{ic}\mid\widetilde{\gamma}_{ik}=0,\mathbf{w}^c)\,p(\tau_{ik}\mid\widetilde{\gamma}_{ik}=0)\,p(\widetilde{\gamma}_{ik}=0\mid\widetilde{\theta}_k)}{p(y_{ic}\mid\widetilde{\gamma}_{ik}=1,\mathbf{w}^c)\,p(\tau_{ik}\mid\widetilde{\gamma}_{ik}=1)\,p(\widetilde{\gamma}_{ik}=1\mid\widetilde{\theta}_k)}}\end{aligned} \tag{6}$$

where c corresponds to the c-th disease presented in sample i, and $\mathbf{w}^c$ is the c-th column of the matrix of weights $\mathbf{W} \in \mathbb{R}^{(K+1)\times C}$.

From this, the only new expression left to compute is $p\left(y_{ic} \mid \widetilde{\gamma}_{ik}=1,\mathbf{w}^c\right)$, given by

$$p(y_{ic}\mid\widetilde{\gamma}_{ik}=1,\mathbf{w}^c)=\sum_{\widetilde{\gamma}_{i,\setminus k}} p(y_{ic}\mid\widetilde{\gamma}_{ik}=1,\widetilde{\gamma}_{i,\setminus k},\mathbf{w}^c)\,p(\widetilde{\gamma}_{i,\setminus k}\mid\widetilde{\gamma}_{ik}=1,\boldsymbol{T},\widetilde{\boldsymbol{\theta}}).$$

Since this is not available in closed form, we approximate it using Monte Carlo. Specifically, we

- Generate M samples from $p\left(\widetilde{\gamma}_{ik} \mid \mathbf{T},\widetilde{\boldsymbol{\theta}}\right)$, a probability to which we have access (see the supplementary material of Moran and George (2021)). This is done for every $k \in \{1,\ldots,K\}$. The results will be stored in a matrix $\mathbf{V} \in \mathbb{R}^{M\times K}$.
- For each column $k \in \{1,\ldots,K\}$ in $\mathbf{V}$:
  1. Extract samples of $\tilde{\boldsymbol{\gamma}}$ where $\widetilde{\gamma}_{ik}=1$; that is, extract only the rows of $\mathbf{V}$ whose k-th column is equal to 1. Note that these are samples from $p(\widetilde{\gamma}_{i,\setminus k} \mid \widetilde{\gamma}_{ik}=1,\boldsymbol{T},\widetilde{\boldsymbol{\theta}})$. This subset of $\mathbf{V}$ can be defined as $\mathbf{V}' \in \mathbb{R}^{M'\times K}$ where $M' \le M$.
  2. Compute $p(y_{ic} \mid \widetilde{\gamma}_{ik}^{(m)}=1,\widetilde{\gamma}_{i,\setminus k}^{m},\mathbf{w}^c)$ using (5) for the samples $m=1,\ldots,M'$.
  3. Finally, estimate $p\left(y_{ic} \mid \widetilde{\gamma}_{ik}=1,\mathbf{w}^c\right)$ as

$$p\left(y_{ic}\mid\widetilde{\gamma}_{ik}=1,\mathbf{w}^c\right) \approx \frac{1}{M'}\sum_{m=1}^{M'} p(y_{ic}\mid\widetilde{\gamma}_{ik}^{(m)}=1,\widetilde{\gamma}_{i,\setminus k}^{m},\mathbf{w}^c).$$

In our numerical experiments, we have empirically observed that using $M=50$ results in estimates of $\langle\widetilde{\gamma}_{ik}\rangle$ with sufficiently low variance and consistent results across multiple trials, indicating that this choice of M is sufficient for reliable estimation.

Connection to mean-field variational inference. The steps involved in the computation of $\langle\widetilde{\gamma}_{ik}\rangle$ in our model assume conditional independence across the binary latent indicators $\widetilde{\gamma}_{ik}$, thus approximating the full posterior $p(\widetilde{\boldsymbol{\Gamma}} \mid \mathbf{X},\mathbf{Y},\boldsymbol{\Delta}^{(t)})$ by the product of marginals:

$$p(\widetilde{\boldsymbol{\Gamma}}\mid\mathbf{X},\mathbf{Y},\boldsymbol{\Delta}^{(t)}) \approx \prod_{i=1}^{N}\prod_{k=1}^{K} p(\widetilde{\gamma}_{ik}\mid\mathbf{X},\mathbf{Y},\boldsymbol{\Delta}^{(t)}).$$

This corresponds to a standard *mean-field variational inference* (MFVI) approximation, where the variational posterior factorises as $q(\widetilde{\boldsymbol{\Gamma}})=\prod_{i,k} q(\widetilde{\gamma}_{ik})$, with each $q(\widetilde{\gamma}_{ik})$ taken to be a Bernoulli distribution parameterised by the current estimate of $\mathbb{E}[\widetilde{\gamma}_{ik}]$. Under this framework, the optimal update for $q(\widetilde{\gamma}_{ik})$ is given by Equation (6) (Blei et al. 2017).

In addition, the conditional likelihood $p(y_{ic} \mid \widetilde{\gamma}_{ik}=1,\mathbf{w}^c)$ depends formally on all other latent indicators $\widetilde{\gamma}_{i,\setminus k}$, that is

$$p(y_{ic}\mid\widetilde{\gamma}_{ik}=1,\mathbf{w}^c)=\sum_{\widetilde{\gamma}_{i,\setminus k}} p(y_{ic}\mid\widetilde{\gamma}_{ik}=1,\widetilde{\gamma}_{i,\setminus k},\mathbf{w}^c)\,p(\widetilde{\gamma}_{i,\setminus k}\mid\widetilde{\gamma}_{ik}=1,\mathbf{T},\widetilde{\boldsymbol{\theta}}).$$

While our Monte Carlo estimate does not explicitly model full dependence among all $\widetilde{\gamma}_i$, it is aligned with the mean-field variational approximation assumed. Empirically, this approach yields stable estimates and effective convergence in our experiments. For similar treatments under mean-field assumptions, see also Ročková and George (2014); Carbonetto and Stephens (2012).

3.3 Expectation of the log-sum-exp expression in $Q(\boldsymbol{\Delta} \mid \boldsymbol{\Delta}^{(t)})$

For the computation of the last term of (B4) which implies an expectation of a log-sum-exp expression, since we now have a way to estimate $\langle\widetilde{\gamma}_{ik}\rangle$, the computation of this expectation is a simple Monte Carlo estimate as follows:

- Generate $\widetilde{\boldsymbol{\gamma}}_i^{\prime(1)},\ldots,\widetilde{\boldsymbol{\gamma}}_i^{\prime(m)}$ samples from $p\left(\widetilde{\gamma}_{ik}=1 \mid \mathbf{Y},\mathbf{T},\widetilde{\boldsymbol{\theta}},\mathbf{W}\right)$ previously estimated in Section 3.2, for each $k \in \{1,\ldots,K\}$.
- Compute

$$\left\langle \log\left[\sum_{l=1}^{C}\exp\left(\mathbf{w}^{(l)\,T}\widetilde{\boldsymbol{\gamma}}_i^{\prime}\right)\right]\right\rangle \approx \frac{1}{m}\sum_{i=1}^{m}\log\left[\sum_{l=1}^{C}\exp\left(\mathbf{w}^{(l)\,T}\widetilde{\boldsymbol{\gamma}}_i^{\prime(m)}\right)\right]$$

3.4 Estimation of hyperparameter $\zeta_{\mathbf{w}}$

To estimate the regularisation hyperparameter ζ_{w} of the ℓ_2 penalisation term of the multinomial logistic regression weights matrix $\mathbf{W}$, we adopted an empirical Bayesian

approach by maximum marginal likelihood estimate. This can be done by solving the following

$$\begin{aligned}\zeta_{\mathrm{w}}^{*} &= \arg\max_{\zeta_{\mathrm{w}}} p\left(\mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}\right) \\ &= \arg\max_{\zeta_{\mathrm{w}}} \int_{\mathbb{R}^{(K+1)\times C}} \prod_{i=1}^{N}\prod_{l=1}^{C}\left[\exp\left(\mathbf{w}^{(l)\,T}\tilde{\boldsymbol{\gamma}}_i'\right) / \sum_{l'=1}^{c} \exp\left(\mathbf{w}^{(l')\,T}\tilde{\boldsymbol{\gamma}}_i'\right)\right]^{y_{il}} \\ &\quad + \exp\left(-\frac{1}{2}\zeta_{\mathrm{w}}\|\mathbf{W}\|_F^2\right) d\mathbf{W}. \end{aligned} \tag{7}$$

Since the latter integral, i.e., the resulting marginal likelihood of the multinomial logistic regression model, is computationally intractable, we will apply the Stochastic Optimisation via Unadjusted Langevin (SOUL) method (De Bortoli et al. 2021), which is specifically designed for this type of problem. We will explain this method in detail below.

We can solve (7) iteratively using the projected gradient algorithm (Levitin and Polyak 1966, Section 5)

$$\zeta_{\mathrm{w}}^{(n+1)} = \Pi_{\Theta_{\zeta_{\mathrm{w}}}}\left[\zeta_{\mathrm{w}}^{(n)} + \delta_{\mathrm{PGA}}^{(n)} \nabla_{\zeta_{\mathrm{w}}} p\left(\mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}^{(n)}\right)\right], \tag{8}$$

by computing a sequence $(\zeta_{\mathrm{w}}^{(n)})_{n\in\mathbb{N}}$ associated with the latter recursion, where $\Pi_{\Theta_{\zeta_{\mathrm{w}}}}$ denotes the projection onto the compact convex set $\Theta_{\zeta_{\mathrm{w}}} \subset (0, +\infty)$ and $(\delta_{\mathrm{PGA}}^{(n)})_{n\in\mathbb{N}}$ is a sequence of non-increasing step sizes[2]. However, the gradient in (8) is intractable, as we saw in (7). For this case, we can replace this gradient with a stochastic estimator by applying Fisher's identity (Douc et al. 2014, Section D.2)

$$\begin{aligned}&\nabla_{\zeta_{\mathrm{w}}} p\left(\mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}\right) \\ &= \int_{\mathbb{R}^{(K+1)\times C}} \frac{\nabla_{\zeta_{\mathrm{w}}} p\left(\mathbf{W}, \mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}\right)}{p\left(\mathbf{W}, \mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}\right)} p\left(\mathbf{W} \mid \mathbf{Y}, \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}\right) d\mathbf{W} \\ &= \int_{\mathbb{R}^{(K+1)\times C}} \nabla_{\zeta_{\mathrm{w}}} \\ &\log p\left(\mathbf{W}, \mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}\right) p\left(\mathbf{W} \mid \mathbf{Y}, \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}\right) d\mathbf{W}, \end{aligned}$$

5 where $p(\mathbf{W} \mid \mathbf{Y}, \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}})$ is the posterior distribution of $\mathbf{W}$, given by

$$p\left(\mathbf{W} \mid \mathbf{Y}, \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}\right) \propto p\left(\mathbf{Y} \mid \mathbf{W}, \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}\right) p\left(\mathbf{W} \mid \zeta_{\mathrm{w}}\right).$$

[2] When denoting sequences of values, we use superscripts in parentheses instead of subscript and superscript notation to avoid confusion with matrix notation. For example, $\mathbf{M}^{(i)}$ represents the i-th value in a sequence, rather than a power or column/row vector.

Given the fact that $p(\mathbf{W}, \mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}) = p(\mathbf{Y} \mid \mathbf{W}, \tilde{\mathbf{\Gamma}}') p(\mathbf{W} \mid \zeta_{\mathrm{w}})$ we have

$$\nabla_{\zeta_{\mathrm{w}}} p\left(\mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}\right) = \int_{\mathbb{R}^{(K+1)\times C}} \nabla_{\zeta_{\mathrm{w}}} \log p\left(\mathbf{W} \mid \zeta_{\mathrm{w}}\right) p\left(\mathbf{W} \mid \mathbf{Y}, \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}\right) d\mathbf{W}$$

where

$$\begin{aligned}\log p\left(\mathbf{W} \mid \zeta_{\mathrm{w}}\right) &= -\frac{1}{2}\zeta_{\mathrm{w}}\|\mathbf{W}\|_F^2 - \int_{\mathbb{R}^{(K+1)\times C}} \exp\left(-\frac{1}{2}\zeta_{\mathrm{w}}\|\mathbf{W}\|_F^2\right) d\mathbf{W} \\ &= -\frac{1}{2}\zeta_{\mathrm{w}}\|\mathbf{W}\|_F^2 - \log\left(\frac{2\pi}{\zeta_{\mathrm{w}}}\right)^{0.5\times(K+1)\times C}. \end{aligned}$$

Therefore

$$\nabla_{\zeta_{\mathrm{w}}} \log p\left(\mathbf{W} \mid \zeta_{\mathrm{w}}\right) = -0.5\|\mathbf{W}\|_F^2 + \frac{(K+1)\times C}{2\zeta_{\mathrm{w}}}$$

Having finally that

$$\begin{aligned}\nabla_{\zeta_{\mathrm{w}}} p\left(\mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}\right) &= \frac{(K+1)\times C}{2\zeta_{\mathrm{w}}} - 0.5\int_{\mathbb{R}^{(K+1)\times C}} \|\mathbf{W}\|_F^2 p\left(\mathbf{W} \mid \mathbf{Y}, \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}\right) d\mathbf{W} \\ &= \frac{(K+1)\times C}{2\zeta_{\mathrm{w}}} - \frac{1}{2}\mathbb{E}_{\mathbf{W}|\mathbf{Y},\tilde{\mathbf{\Gamma}}',\zeta_{\mathrm{w}}}\left[\|\mathbf{W}\|_F^2\right], \end{aligned}$$

that is, the gradient we need for the iterative scheme in (8) depends on the computation of an expectation that can be estimated using MCMC methods. To approximate samples from the posterior distribution $p(\mathbf{W} \mid \mathbf{Y}, \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}})$ and compute the latter expectation, the SOUL method uses the unadjusted Langevin algorithm (ULA) (Roberts and Tweedie 1996; Dalalyan 2017; Durmus and Moulines 2017), given by

$$\begin{aligned}W^{(k+1)} &= W^{(k)} - \delta_{\mathrm{ULA}} \nabla_W \log p(W^{(k)} \mid \mathbf{Y}, \tilde{\mathbf{\Gamma}}', \zeta_{\mathrm{w}}) \\ &\quad + \sqrt{2\delta_{\mathrm{ULA}}} Z^{(n+1)}, \end{aligned} \tag{9}$$

where $\delta_{\mathrm{ULA}} > 0$ is a given step size and $(Z^{(n+1)})_{n\geq 0}$ is an i.i.d. sequence of $(K+1)\times C$ - dimensional standard Gaussian random vectors. The SOUL method adapted for this problem, and details about its implementation can be found in Section C.

3.5 Maximisation step regarding the variable W

Finally, the last expression to compute in the new SSLB model is given by

$$\widehat{\mathbf{W}} = \underset{\mathbf{W}\in\mathbb{R}^{(K+1)\times C}}{\operatorname{argmin}} \sum_{i=1}^{N}\sum_{l=1}^{C} y_{il}\mathbf{w}_l^T \langle\widetilde{\boldsymbol{\gamma}}_i'\rangle + \sum_{i=1}^{N}\left\langle \log\left[\sum_{l=1}^{C}\exp\left(\mathbf{w}_l^T\widetilde{\boldsymbol{\gamma}}_i'\right)\right]\right\rangle + \frac{1}{2}\hat{\zeta}_{\mathrm{w}}\|\mathbf{W}\|_F^2.$$

Since there is no closed-form solution for the latter, we decided to implement an accelerated gradient descent (AGD) algorithm (Nesterov 1983; Güler 1992; Salzo and Villa 2012) to ensure rapid convergence to the minimum. To apply AGD, we need the gradient of the latter expression, which was given in Section D.1. This allows us to define the following iterative scheme

$$\mathbf{W}^{(0)} = \mathbf{W}^{(-1)} = \mathbf{0} \in \mathbb{R}^{(K+1)\times C};\ \mathbf{V}^{(0)} \in \mathbb{R}^{(K+1)\times C};\quad t_0 = 0 \in \mathbb{R},$$

$$t_{s+1} = \frac{1+\sqrt{1+4t_s^2}}{2},$$
$$\mathbf{V}^{(s)} = \mathbf{W}^{(s)} + \frac{t_s - 1}{t_{s+1}}\left(\mathbf{W}^{(s)} - \mathbf{W}^{(s-1)}\right),$$
$$\mathbf{W}^{(s+1)} = \mathbf{V}^{(s)} + \delta_{\mathrm{AGD}}\overline{\nabla}_{\mathbf{w}}\log p(\mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \mathbf{V}^{(s)}, \hat{\zeta}_{\mathrm{w}}),$$

$$s \in \{0, \ldots, S-1\} \subset \mathbb{N},$$

where δ_{AGD} is the step size or learning rate of the iterative AGD scheme, which must be carefully set to avoid divergence. Note that, on each AGD iteration, we need to generate a collection of $\tilde{\mathbf{\Gamma}}'^{(1)}, \ldots, \tilde{\mathbf{\Gamma}}'^{(J)}$ samples from $p(\widetilde{\gamma}_{ik} = 1 \mid \mathbf{Y}, \mathbf{T}, \widetilde{\boldsymbol{\theta}}, \mathbf{W})$ to compute the Monte Carlo estimate of the gradient (see Section D for details).

Note: While Section 3.4 approximates the posterior distribution of $\mathbf{W}$ using the SOUL algorithm to estimate the ℓ_2 regularisation hyperparameter, the current section focuses on computing a point estimate of $\mathbf{W}$ via MAP optimisation within the M-step of the EM algorithm. These two steps serve distinct purposes within our inference framework: SOUL is used for hyperparameter tuning by integrating over $\mathbf{W}$, while Nesterov's method is employed for parameter estimation given fixed hyperparameters.

With the methodology established, we will refer to this modified version of the SSLB model as **OG-SSLB**, which stands for **Outcome-Guided Spike-and-Slab Lasso Biclustering**.

4 Numerical Results

4.1 Simulation Study

In this section, we evaluate the performance of OG-SSLB compared to SSLB in a simulation setting. We use the consensus score metric (Hochreiter et al. 2010) to measure the accuracy of biclusters identified by each method relative to the true biclusters. The highest possible consensus score is 1, indicating identical sets of biclusters.

We reproduce the simulation described in (Moran and George 2021, Section 3.1), where a simulated dataset with $N = 300$, $G = 1000$, and $K = 15$ biclusters is examined. The data simulation follows settings closely aligned with those in the FABIA (Hochreiter et al. 2010) and SSLB studies. The data matrix $\mathbf{X}$ is generated as $\mathbf{Z}\Lambda^T + \mathbf{E}$, with each entry in the noise matrix $\mathbf{E}$ sampled from an independent standard normal distribution. For each column $\mathbf{z}^k$, the number of samples in bicluster k is drawn uniformly from $\{5, \ldots, 20\}$. The indices of these elements are randomly selected and assigned values from $N(\pm 2, 1)$, with the sign of the mean chosen randomly. The elements of $\mathbf{z}^k$ not in the biclusters have values drawn from $N(0, 0.2^2)$. The columns $\boldsymbol{\lambda}^k$ are generated similarly, except that the number of elements in each bicluster is drawn from $\{10, \ldots, 50\}$.

To construct the disease outcome matrix $\mathbf{Y}$ for the OG-SSLB algorithm, we first generate a matrix of weights $\mathbf{W}$ in the following way

1. **Intercepts (Baseline Weights for Healthy Control Group)**: The first row of $\mathbf{W}$, representing the intercepts, is initialised with small values

$$W_{1j} = \log(\epsilon) \quad \text{for all } j,$$

where $0 < \epsilon < 1$. These small values correspond to the reference class (i.e., the healthy control group, HC) in the multinomial logistic regression model. In logistic regression, the intercept term controls the baseline probability of a sample belonging to the reference class when no covariates (in this case, the bicluster assignments) are active. By assigning a small value to W_{1j}, we increase the baseline probability for healthy control samples when no bicluster is assigned to a sample. Since $\log(\epsilon)$ with $\epsilon < 1$ results in a negative value, this translates into a higher probability of belonging to the reference class (HC).

2. **Weight Assignment for Biclusters (Bias Towards Disease Samples)**: For samples belonging to a bicluster, we adjust the weights in the remaining rows of $\mathbf{W}$, correspond-

ing to the non-reference classes (i.e., the disease classes). Specifically, we set the weights for the non-reference class as:

$$W_{ij} = \log(1/\epsilon) \quad \text{for } i > 1, j.$$

Here, $\log(1/\epsilon)$, where $\epsilon < 1$, results in a positive value, increasing the likelihood that samples assigned to a bicluster are classified as disease samples. Importantly, only one column j is randomly selected for each row i in $\mathbf{W}$ to be assigned this value. This ensures that a specific bicluster is more strongly associated with a particular disease group, effectively biasing the model toward classifying samples in that bicluster as disease samples.
Finally, the assignment of disease for each sample is determined by applying the multinomial logistic regression model using the defined weights $\mathbf{W}$.

The rationale behind making samples belonging to a bicluster more likely to be classified as disease cases stems from the biological assumption that disease states are often driven by specific gene expression patterns (Ota et al. 2021; Mesko et al. 2011; Veer et al. 2002; Vijver et al. 2002). Biclustering aims to identify subsets of genes that co-vary together in certain subsets of samples, which may represent distinct biological processes or pathways that are activated in disease conditions.

We adopt similar hyperparameter configurations for SSLB and OG-SSLB, detailed in (Moran and George 2021, Section 2.5). In particular, the slab parameters for the loadings and the factors, $\mathbf{\Lambda}$ and $\mathbf{Z}$, are set to $\omega_1, \tilde{\omega}_1 = 1$. The spike parameters for $\mathbf{Z}$ follow an increasing sequence of $\omega_0 \in \{1, 5, 10, 50, 100, 500, 10^3, 10^4, 10^5, 10^6, 10^7\}$. The spike parameters for $\mathbf{Z}$ are chosen as $\tilde{\omega}_0 \in \{1, 5, \ldots, 5\}$ to correspond to the length of the sequence ω_0. Specifically, the values of $\tilde{\omega}_0$ are fixed at $\tilde{\omega}_0 = 5$. Furthermore, the initial overestimate of the number of biclusters is set to $K^* = 30$.

We compared 50 realisations of SSLB and OG-SSLB using the same simulated dataset across all runs while varying the algorithmic initial conditions for each of the fifty runs. This analysis was performed under three distinct implementations: SSLB/OG-SSLB with the Pitman–Yor extension (PY), where $\tilde{\alpha} = 1$ and $d = 0.5$, SSLB/OG-SSLB with the stick-breaking IBP prior (IBP) where $\tilde{\alpha} = 1$, and SSLB/OG-SSLB with the finite approximation to the IBP prior (Beta-Binomial, BB), where $\tilde{a} = 1/K^*$ and $\tilde{b} = 1$. For OG-SSLB, we implemented two variations: a non-informative approach, which assigns values around $\log(1)$ to all elements of the matrix $\mathbf{W}$, resulting in an imprecise simulated $\mathbf{Y}$ outcome, and an informative approach, which assigns $\log(1/2)$ to the intercepts and $\log(4)$ to specific bicluster-disease elements (i.e., one specific column in each row of $\mathbf{W}$), yielding a more informative simulated $\mathbf{Y}$ outcome. We aim to show the difference in adding more information to the model.

The distribution of the consensus score for each method can be seen in Figure 4. OG-SSLB consistently achieves higher consensus scores in all three prior versions of the binary indicators for the factors $\mathbf{Z}$, reaching even higher precision in the informative case (with a slight increase in the PY prior version for the informative case). We also illustrated in Figure 5 a comparison of the SSLB and OG-SSLB methods with the FABIA and BicMix algorithms. It is important to mention that FABIA requires the number of biclusters in advance, so we provided the true number of biclusters (i.e., $k = 15$) in all 50 runs. BicMix has its own method for estimating the number of biclusters in the data (Gao et al. 2016). As shown in the figure, the consensus scores achieved by FABIA and BicMix are considerably lower than those of SSLB and OG-SSLB. In addition, Table 1 presents the mean, over the 50 runs, of the estimated number of biclusters $\hat{K}$ for the methods that estimate K. All implementations of the informative OG-SSLB approach are closer to the true number of biclusters compared to SSLB. BicMix achieves a mean estimate of 12.56 biclusters. To further validate our results, we also evaluated biclustering accuracy using the Clustering Error metric (Horta and Campello 2014; Nicholls and Wallace 2021). These results, presented in Appendix E, confirm the same trend observed with the consensus score, with OG-SSLB (inf.) consistently achieving the best performance.

Regarding the corresponding run times, the SSLB implementations required between 35 and 60 seconds, whereas the OG-SSLB implementations took between 2600 and 3500 seconds. While OG-SSLB is computationally more intensive due to the additional stochastic optimisation steps for estimating regularisation parameters due to the SOUL algorithm, this increase is justified by the significant gains in performance, particularly in identifying more stable and biologically meaningful biclusters. These results demonstrate a clear trade-off between computational cost and modelling flexibility. We discuss potential strategies to mitigate this cost in the Conclusion.

4.2 Breast Cancer Microarray Dataset

The dataset used in this study consists of gene expression data from 337 breast cancer patients diagnosed with stage I or II breast cancer (Vijver et al. 2002; Veer et al. 2002). It comprises the expression levels of 24,158 genes, resulting in a large and high-dimensional data structure. These data have been widely used to study the heterogeneity of breast cancer, a disease known to comprise several molecular subtypes, including estrogen receptor-positive (ER+) and estrogen receptor-negative (ER-) subtypes. The patients' ER status is determined based on the expression of the ESR1

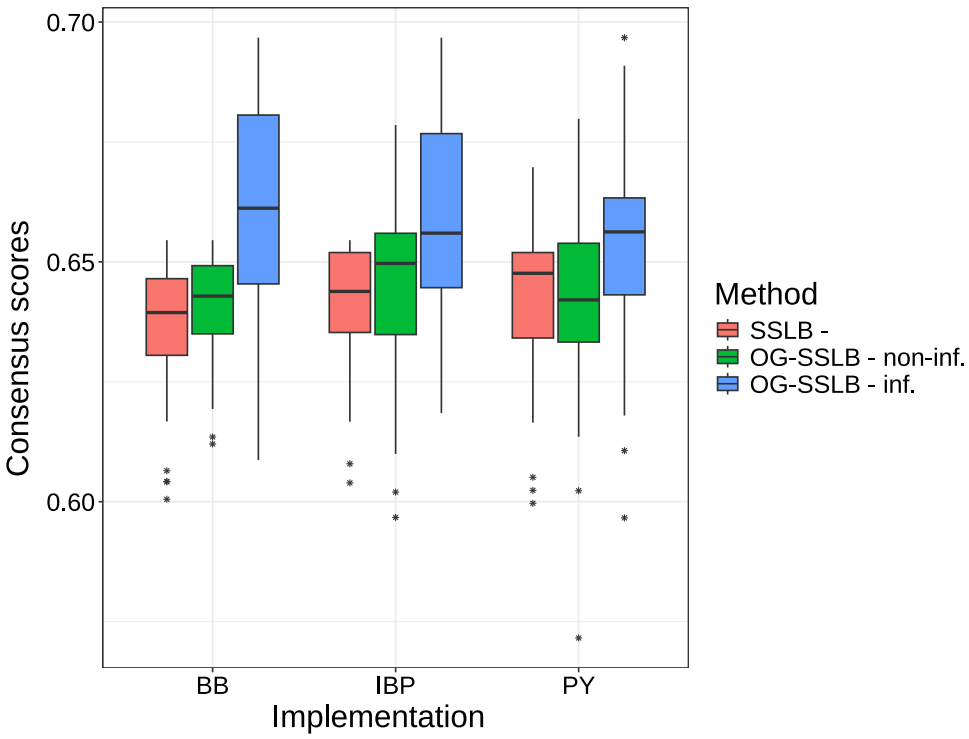

Fig. 4 Consensus scores of 50 replications of SSLB and OG-SSLB, both using three different prior implementations for $\tilde{\mathbf{\Gamma}}$: BB, PY and IBP (see Section 2.1.2 for details). For OG-SSLB, we implement a non-informative (i.e., non-inf.) approach, setting values for all elements of the matrix $\mathbf{W}$ around log(1), which produces a diffusive $\mathbf{Y}$ outcome, and an informative (i.e., inf.) approach, where we set $\log(1/2)$ for the intercepts and log(4) for specific bicluster-disease elements in the matrix $\mathbf{W}$, resulting in a more informative $\mathbf{Y}$ outcome.

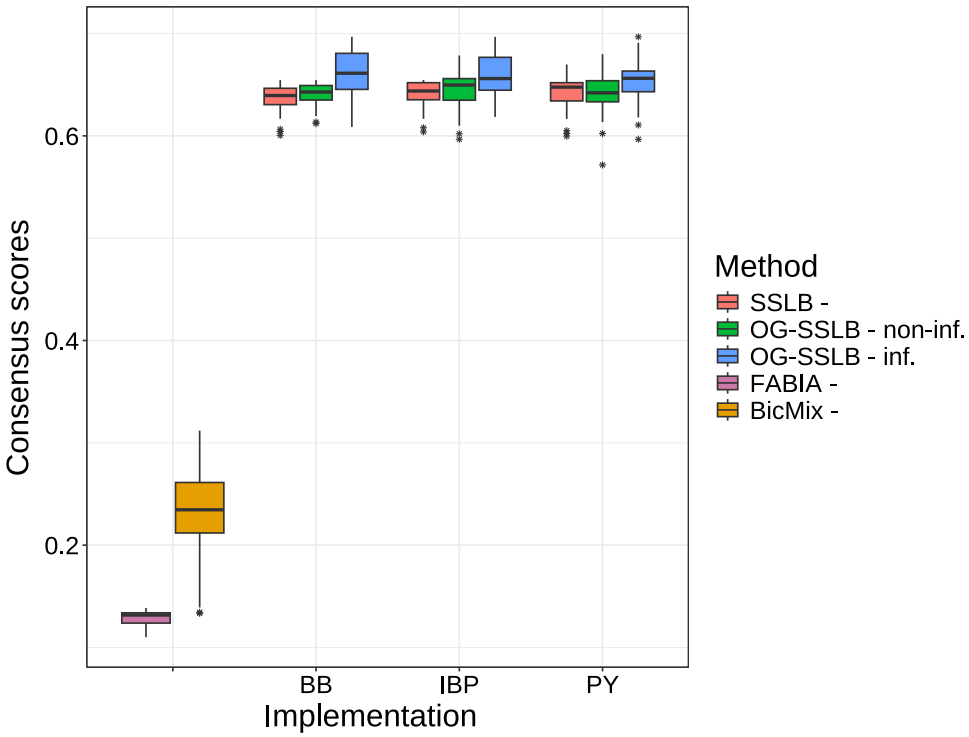

Fig. 5 Consensus scores of 50 replicates for FABIA, BicMix, SSLB, and OG-SSLB. For (OG-)SSLB, we include results under the three prior choices for $\tilde{\mathbf{\Gamma}}$: BB, PY, and IBP (see Section 2.1.2). OG-SSLB results are shown for both the non-informative and informative settings described in the caption of Figure 4.

Table 1 Mean estimated number of biclusters, $\hat{K}$, over 50 replications ($K_{\text{true}} = 15$). The three SSLB implementations (BB, IBP, PY) refer only to SSLB and OG-SSLB.

SSLB implementation	Method SSLB	OG-SSLB (non-inf.)	OG-SSLB (inf.)	BicMix
BB	14.04	14.02	14.50	|
IBP	14.10	14.36	14.50	12.56
PY	14.34	14.42	14.46	|

gene, which encodes the estrogen receptor, and serves as a key indicator of the disease subtype and treatment options.

The goal of this analysis is to identify meaningful biclusters of patients and genes that reflect the biological differences between these subtypes, particularly focusing on the estrogen receptor status (ER+ or ER-). Biclustering is especially well-suited for this task as it allows the simultaneous grouping of patients and relevant genes, which may highlight subtype-specific gene expression patterns. Consequently, it has also been used as a benchmark for biclustering methods such as FABIA (Hochreiter et al. 2010), BicMix (Gao et al. 2016), and SSLB.

To assess the performance of our proposed OG-SSLB method compared to the standard SSLB algorithm, we ran 50 replicates, each with different initial conditions, for both methods. Both SSLB and OG-SSLB were initialised with an overestimate of the number of biclusters $K^* = 50$. For the loadings, $\mathbf{\Lambda}$, we configure the Beta-Binomial hyperparameters as $a = 1/(GK^*)$ and $b = 1$. This normalisation by G enhances the focus on sparsity. For the factors, $\mathbf{Z}$, the IBP prior is used with hyperparameters set to $\tilde{\alpha} = 1/N$ and $d = 0$. The other parameters are assigned the default values specified in (Moran and George 2021, Section 2.5). For the OG-SSLB method, we incorporate the patients' ER status (1 for ER+ and 0 for ER-) as the outcome variable.

After obtaining the biclusters, we performed a Wilcoxon rank-sum test on the estimated factors (i.e., $\mathbf{Z}$ matrix) for each bicluster in every replicate, comparing the distributions of factor values between ER+ and ER– patients. For each run, we recorded the minimum p-value across all biclusters. The resulting $-\log_{10}(p)$ values are summarised in Figure 6. As shown, both SSLB and OG-SSLB achieve similarly high significance across replicates, suggesting that in this real-data scenario, both methods are equally able to recover subtype-relevant biclusters.

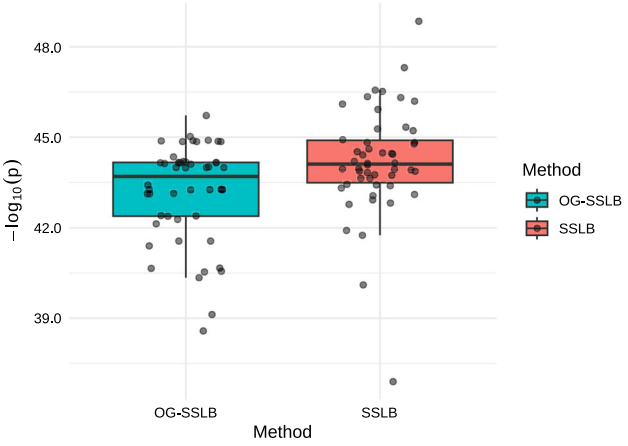

Fig. 6 Distribution of the smallest p-value from Wilcoxon rank-sum tests comparing ER+ vs ER– patients across estimated factors (the estimated $\mathbf{Z}$ matrix) for each estimated bicluster in each of the 50 SSLB and OG-SSLB replicates applied to the breast cancer microarray data.

4.3 Immune Cell Gene Expression Atlas, University of Tokyo

Detecting biclusters in transcriptomic data is one of the motivating applications for OG-SSLB, therefore, we also applied the SSLB and OG-SSLB methods to gene expression data from Ota et al. (2021). This study provides a comprehensive database of transcriptomic and genome sequencing data from a wide range of immune cells from patients with immune-mediated diseases (IMD). This collection of data, termed the "Immune Cell Gene Expression Atlas from the University of Tokyo (ImmuNexUT)", includes gene expression patterns consisting of healthy volunteers and patients diagnosed with systemic lupus erythematosus (SLE), idiopathic inflammatory myopathy (IIM), systemic sclerosis (SSc), mixed connective tissue disease (MCTD), Sjögren's syndrome (SjS), rheumatoid arthritis (RA), Behçet's disease (BD), adult-onset Still's disease (AOSD), ANCA-associated vasculitis (AAV), or Takayasu arteritis (TAK). The dataset encompasses 28 distinct immune cell types, nearly covering all peripheral immune cells. We anticipate that a subset of genes may cluster in a subset of patients who share some specific aetiology, and that this bicluster will be enriched in genes related to that aetiology and patients with related diseases.

In order to evaluate our method in a real-world dataset where we have some expectation of what to find, we focused on monocytes, which are known to express an interferon-response gene expression signature, found more often in patients with IMD, and particularly SLE (Nikolakis et al. 2023; Perez et al. 2022). The data were pre-processed as follows:

1. We perform batch normalisation using the ComBat-seq R package (Zhang et al. 2020).
2. We reduce low-count genes using the edgeR package (Robinson et al. 2009).
3. We calculated the Pearson correlation matrix between genes and setting a threshold at the 90th percentile, we focus on the most highly correlated gene pairs. Genes with fewer than five other genes correlating above this threshold are removed to eliminate those with weak or non-specific interactions, which could be noisy or less informative. This step ensures that only genes with strong co-expression relationships, potentially reflecting meaningful biological connections, are retained.
4. Finally, to correct for technical variation and differences in sequencing depth between samples, we applied the median of ratios normalisation method, as implemented in the DESeq2 R package (Love et al. 2014). This normalisation ensures that gene expression differences reflect true

biological variability rather than artefacts from varying read counts across samples.

Following preprocessing, we obtained a dataset comprising $N = 410$ and $G = 11215$. We ran 20 different replicates of both SSLB and OG-SSLB using the IBP prior for $\tilde{\mathbf{\Gamma}}$, with hyperparameters $\tilde{\alpha} = 1/N$ and $d = 0$, and the Beta-Bernoulli prior hyperparameters for $\mathbf{\Gamma}$ set to $a = 1/(GK^*)$ and $b = 1$. Our choice of using the IBP prior and the specified hyperparameters values for the factor and loading binary indicator matrices is in agreement with the real data experiments performed in the SSLB paper (see (Moran and George 2021, Sections 4 and 5) for details). The remaining hyperparameter settings were similarly aligned with those of the previous numerical experiment. Furthermore, the initial overestimate for the number of biclusters was set to $K^* = 50$.

From Nicholls et al. (2022), we obtained a list of 56 genes associated with the IFN signature, 51 of which were found present in the preprocessed dataset. To focus on sparse, IFN-related biclusters, we filtered the results from both methods to include only biclusters with less than 50% of the total number of samples and more than 6 IFN genes.

The results are first summarised in Figure 7, where a heatmap of the standardised gene expression data is shown. As can be seen, SSLB and OG-SSLB generally identify the same samples and genes forming the IFN biclusters, but OG-SSLB does so far more consistently: it finds IFN-related biclusters ($\geq$ 7 of the 51 known IFN genes) in 18 of 20 runs, whereas SSLB does so in only 7 of 20 runs.

Note that Figure 7 does not show a single bicluster, but instead summarises the frequency with which each sample and IFN gene is selected into any of the sparse IFN-related biclusters detected across 20 runs of each method. The bar annotations along the margins of the heatmap reflect how often each gene or sample appears in at least one such bicluster for OG-SSLB or SSLB, respectively. This aggregated view helps visualise consistent patterns across replicates rather than illustrating a single bicluster instance.

These results have also been summarised in Table 2. As can be seen, OG-SSLB produces a substantially larger number of replicates in which sparse IFN-related biclusters were detected, in comparison to SSLB.

To further examine disease associations, we analysed the matrix of weights $\mathbf{W}$ estimated by OG-SSLB, focused only on the biclusters corresponding to the sparse IFN-related condition. As shown in Figure 8, SLE consistently exhibited the strongest weights across these biclusters, indicating that these IFN-related profiles are more pronounced in SLE patients compared to other disease groups. Notably, idiopathic inflammatory myopathies (IIM) also showed strong weight values for the sparse IFN-related clusters. This agrees with the known role of type 1 interferon in IIM (Lundberg and Helmers 2010) and its association with disease activity (Kamperman et al. 2024).

Furthermore, Figures 9 and 10 show the distribution of samples and genes identified by the SSLB and OG-SSLB runs. Although neither method recovers all 51 IFN genes in a single bicluster, OG-SSLB identified biclusters under the specified conditions exhibit, in distribution, a higher percentage of SLE patients and a higher number of IFN gene signatures. Additionally, while SLE patients exhibited the highest fraction of patients in the IFN biclusters, IFN signatures have been found in other IMD, and both methods found a higher fraction of patients in IFN biclusters for IIM, MCTD, RA, SjS and SSc. Concerning the associated run times, the SSLB algorithm required about 1900 seconds to run, whereas the OG-SSLB algorithm took approximately 44700 seconds.

5 Conclusions and Discussion

In conclusion, our proposed algorithm, OG-SSLB, exhibits superior performance compared to the SSLB approach in both numerical and real-data experiments, particularly in its ability to estimate the number of biclusters more accurately and achieve higher consensus scores. The flexibility of OG-SSLB, particularly through its multinomial modelling framework, allows it to accommodate more complex clustering structures than the commonly used binomial models. While this improvement entails significantly higher computational costs due to iterative processes such as AGD and ULA, the enhanced precision and modelling capacity of OG-SSLB make it a valuable contribution to biclustering methodologies.

Our subsequent analyses will expand the OG-SSLB framework to the ImmuNexUT dataset, investigating other cell types and detecting new gene expression signatures rather than focusing solely on predefined ones. We aim to identify similarities between diseases, facilitated by the bicluster overlapping allowed by this method.

Additionally, we will explore other machine learning alternatives to multinomial logistic regression, such as support vector machines and naive Bayes, which may offer more robust solutions for integrating disease information into the biclustering framework. An especially promising approach could be the introduction of deep learning classifiers, which may better capture the potential non-linearity of boundary classes, thereby further enhancing the quality of the incorporated disease information. While deep learning classifiers can identify more complex patterns in the data, they would require significantly larger computational resources compared to the multinomial logistic regression model. Nonetheless, the shift from multinomial logistic regression to deep learning or other machine learning approaches presents exciting opportunities to improve classification accuracy

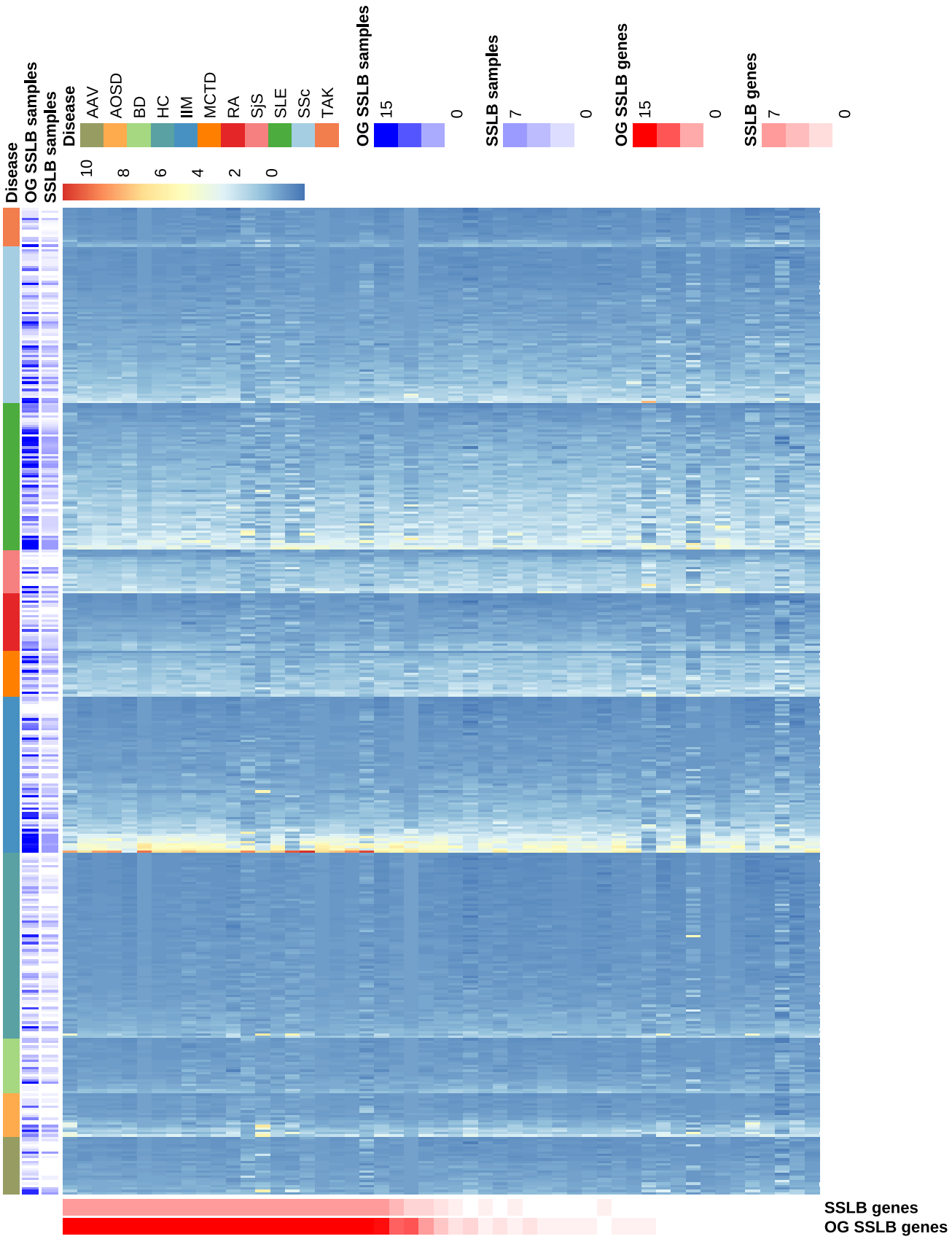

Fig. 7 Heatmap of standardised gene expression data from the ImmuNexUT study, where each column represents a sample and each row corresponds to one of the 51 interferon (IFN) signature genes. The data has been centred and scaled. The colour bars on the margins indicate, across 20 runs, how frequently each sample or gene was included in any sparse IFN-related bicluster identified by OG-SSLB or SSLB. This is not a single bicluster but an aggregate visualisation to summarise common inclusion patterns. Samples are grouped by disease.

Table 2 Results from 20 replicates of applying SSLB and OG-SSLB to the ImmuNexUT real data, focusing on sparse IFN-related biclusters (< 50% of samples, > 6 IFN genes).

Method	Replicates with at least one bicluster that meets sparse filtering cond.	Median % SLE patients in bicluster's replicates
SSLB	7	36
OG-SSLB	18	43.5

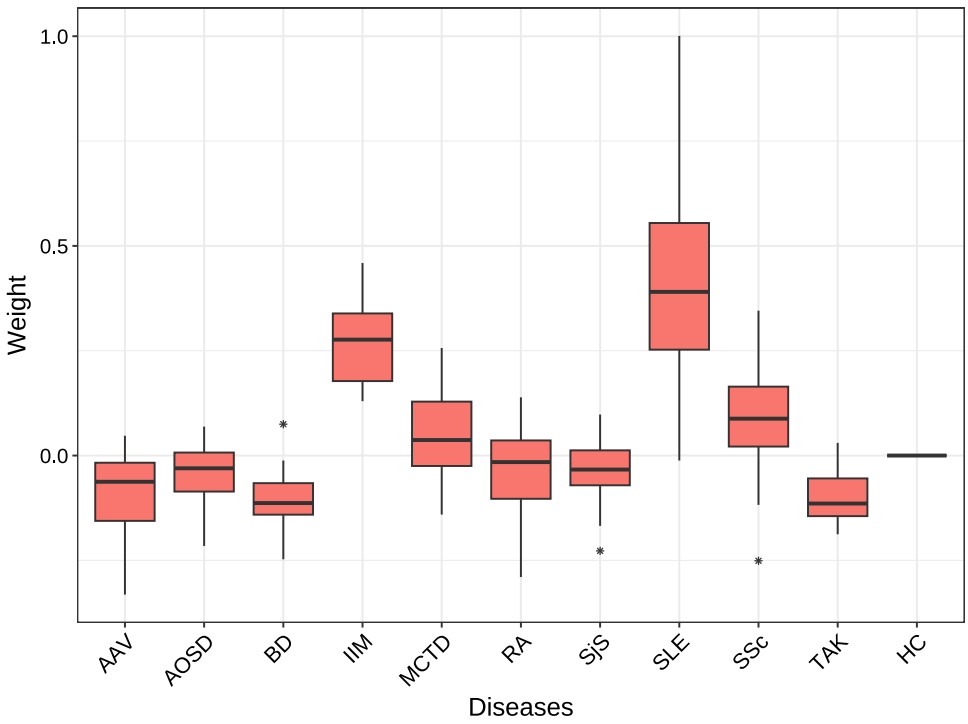

Fig. 8 Distribution of the estimated weight **W** values by OG-SSLB, for each disease, in sparse IFN-related biclusters (i.e., biclusters with less than 50% of the total samples and more than 6 IFN genes).

while addressing the challenges inherent in genomic data analysis.

We acknowledge that OG-SSLB incurs greater computational overhead compared to SSLB, due primarily to the iterative SOUL-based estimation of regularisation parameters in the multinomial regression framework. However, this added complexity enables more precise integration of outcome information and results in more informative biclusters. Future work will also explore strategies to reduce this execution time. For example, a practical approach might involve limiting the number of EM iterations during which the SOUL algorithm is applied and subsequently fixing the regularisation hyperparameters using the average of their recent estimates. This would avoid the need to rerun SOUL in later iterations. Although this is beyond the scope of the present article, this line of investigation may offer a promising trade-off between computational efficiency and model performance.

Regarding future applications, OG-SSLB could play a meaningful role in personalised medicine and biomarker discovery. By identifying sparse and interpretable biclusters that capture subgroups of patients and genes associated with specific disease outcomes, OG-SSLB may help uncover clinically relevant gene signatures. These could serve as potential diagnostic markers or guide stratification of patients for targeted therapies. In particular, its ability to incorporate disease labels makes OG-SSLB especially well-suited to reveal molecular patterns linked to disease heterogeneity, offering translational insights in clinical research.

The source code to reproduce the results in this paper is available online at https://github.com/luisvargasmieles/OGSSLB-examples.

Appendix A Current SSLB model: E step

For completeness, we present the log posterior and the related expression for the E-step of SSLB. For additional details, refer to Moran and George (2021). The complete log posterior is as follows

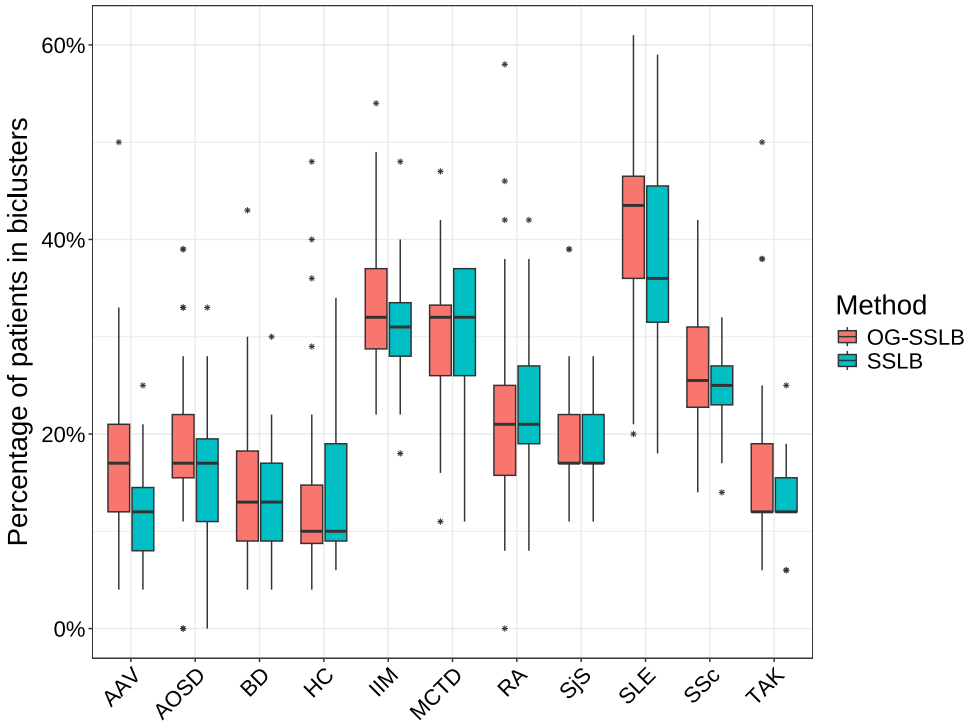

Fig. 9 Results from 20 replicates of applying SSLB and OG-SSLB to the ImmuNexUT real data, focusing on biclusters with less than 50% of the total samples and more than 6 IFN genes: Distribution of the percentage of samples per disease.

$$
\begin{aligned}
\log p(\boldsymbol{\Delta}, \mathbf{Z}, \tilde{\Gamma} \mid \mathbf{X}) \propto & -\frac{1}{2} \sum_{i=1}^{N} \left\{ \left(\mathbf{x_i} - \mathbf{z}_i \boldsymbol{\Lambda}^T \right) \right. \\
& \left. \boldsymbol{\Sigma}^{-1} \left(\mathbf{x_i} - \mathbf{z}_i \boldsymbol{\Lambda}^T \right)^T \right\} \\
& - \frac{N+\eta+2}{2} \sum_{j=1}^{G} \log \sigma_j^2 - \sum_{j=1}^{G} \frac{\eta \xi}{2\sigma_j^2} - \sum_{k=1}^{K^*} \log p\left(\boldsymbol{\lambda}^{\mathbf{k}}\right) \\
& - \frac{1}{2} \sum_{i=1}^{N} \mathbf{z_i} \mathbf{D_i} \mathbf{z_i}^T - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{K^*} \log \tau_{ik} \\
& - \frac{1}{2} \log \sum_{i=1}^{N} \sum_{k=1}^{K^*} \left[(1-\tilde{\gamma}_{ik}) \, \widetilde{\omega}_0 \exp\left(-\frac{\widetilde{\omega}_0 \tau_{ik}}{2} \right) \right. \\
& \left. + \tilde{\gamma}_{ik} \widetilde{\omega}_1 \exp\left(-\frac{\widetilde{\omega}_1 \tau_{ik}}{2} \right) \right] \\
& + \sum_{k=1}^{K^*} \left[\left(\sum_{i=1}^{N} \tilde{\gamma}_{ik} \right) \log \prod_{l=1}^{k} v_l + \left(N - \sum_{i=1}^{N} \tilde{\gamma}_{ik} \right) \right. \\
& \left. \log \left(1 - \prod_{l=1}^{k} v_l \right) \right] + (\tilde{\alpha} - 1) \sum_{k=1}^{K^*} \log v_k, \qquad \text{(A1)}
\end{aligned}
$$

where $\mathbf{D_i} = \operatorname{diag}\left\{ \tau_{i1}^{-1}, \ldots, \tau_{iK^*}^{-1} \right\}$ and $\boldsymbol{\Gamma}$, $\boldsymbol{\theta}$ have been margined out to have $p(\boldsymbol{\Lambda})$ (see Ročková and George (2018) for details). This allows us to define

$$
\begin{aligned}
Q(\boldsymbol{\Delta} \mid \boldsymbol{\Delta}^{(t)}) \propto & -\frac{1}{2} \sum_{i=1}^{N} \left\{ \left(\mathbf{x}_i - \langle \mathbf{z}_i \rangle \boldsymbol{\Lambda}^T \right) \boldsymbol{\Sigma}^{-1} \left(\mathbf{x}_i - \langle \mathbf{z}_i \rangle \boldsymbol{\Lambda}^T \right)^T \right. \\
& \left. + \operatorname{tr}\left[\boldsymbol{\Lambda}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Lambda} \left(\left\langle \mathbf{z}_i \mathbf{z}_i^T \right\rangle - \langle \mathbf{z}_i \rangle \langle \mathbf{z}_i \rangle^T \right) \right] \right\} \\
& - \frac{N+\eta+2}{2} \sum_{j=1}^{G} \log \sigma_j^2 - \sum_{j=1}^{G} \frac{\eta \xi}{2\sigma_j^2} - \sum_{k=1}^{K^*} \log p\left(\boldsymbol{\lambda}^k\right) \\
& - \frac{1}{2} \sum_{i=1}^{N} \left\{ \langle \mathbf{z}_i \rangle \, \mathbf{D}_i \, \langle \mathbf{z}_i \rangle^T + \operatorname{tr}\left[\mathbf{D}_i \left(\left\langle \mathbf{z}_i \mathbf{z}_i^T \right\rangle - \langle \mathbf{z}_i \rangle \langle \mathbf{z}_i \rangle^T \right) \right] \right\} \\
& - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{K^*} \log \tau_{ik} - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{K^*} \left[(1 - \langle \tilde{\gamma}_{ik} \rangle) \, \widetilde{\omega}_0^2 + \langle \tilde{\gamma}_{ik} \rangle \, \widetilde{\omega}_1^2 \right] \tau_{ik} \\
& + \sum_{k=1}^{K^*} \left[\left(\sum_{i=1}^{N} \langle \tilde{\gamma}_{ik} \rangle \right) \right. \\
& \left. \log \prod_{l=1}^{k} v_l + \left(N - \sum_{i=1}^{N} \langle \tilde{\gamma}_{ik} \rangle \right) \log \left(1 - \prod_{l=1}^{k} v_l \right) \right] \\
& + (\tilde{\alpha} - 1) \sum_{k=1}^{K^*} \log v_k, \qquad \text{(A2)}
\end{aligned}
$$

where $\mathbb{E}_{\mathbf{Z}, \tilde{\Gamma} \mid \Delta^{(t)}, \mathbf{X}}[W] = \langle W \rangle$.

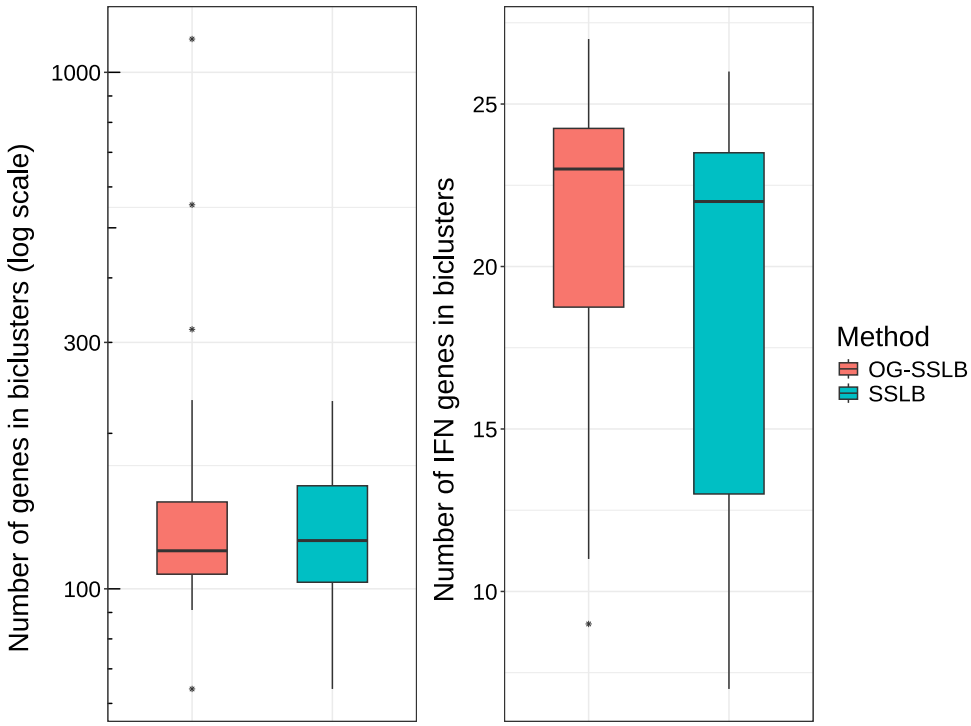

Fig. 10 Results from 20 replicates of applying SSLB and OG-SSLB to the ImmuNexUT real data, focusing on biclusters with less than 50% of the total samples and more than 6 IFN genes. Left: Distribution of the total number of genes in biclusters identified by both methods. Right: Distribution of the number of IFN gene signatures in biclusters identified by both methods.

Appendix B SSLB model with profile regression: E step

The log posterior for the SSLB model using profile regression can be expressed as follows

$$
\begin{aligned}
&\log p(\boldsymbol{\Delta}, \mathbf{Z}, \tilde{\Gamma} \mid \mathbf{X}, \mathbf{Y}) \propto -\frac{1}{2} \\
&\sum_{i=1}^{N}\left\{\left(\mathbf{x_i}-\mathbf{z}_i \boldsymbol{\Lambda}^T\right) \Sigma^{-1}\left(\mathbf{x_i}-\mathbf{z}_i \boldsymbol{\Lambda}^T\right)^T\right\} \\
&-\frac{N+\eta+2}{2} \sum_{j=1}^{G} \log \sigma_j^2-\sum_{j=1}^{G} \frac{\eta \xi}{2 \sigma_j^2} \\
&-\sum_{k=1}^{K^*} \log p\left(\boldsymbol{\lambda}^{\mathbf{k}}\right)-\frac{1}{2} \sum_{i=1}^{N} \mathbf{z_i} \mathbf{D_i} \mathbf{z_i}^T-\frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{K^*} \log \tau_{ik} \\
&-\frac{1}{2} \log \sum_{i=1}^{N} \sum_{k=1}^{K^*}\left[\left(1-\tilde{\gamma}_{ik}\right) \widetilde{\omega}_0 \exp \left(-\frac{\widetilde{\omega}_0 \tau_{ik}}{2}\right)\right. \\
&\left.+\tilde{\gamma}_{ik} \widetilde{\omega}_1 \exp \left(-\frac{\widetilde{\omega}_1 \tau_{ik}}{2}\right)\right] \\
&+\sum_{k=1}^{K^*}\left[\left(\sum_{i=1}^{N} \tilde{\gamma}_{ik}\right) \log \prod_{l=1}^{k} v_l\right. \\
&\left.+\left(N-\sum_{i=1}^{N} \tilde{\gamma}_{ik}\right) \log \left(1-\prod_{l=1}^{k} v_l\right)\right] \\
&+(\tilde{\alpha}-1) \sum_{k=1}^{K^*} \log v_k \\
&-\sum_{i=1}^{N} \sum_{l=1}^{C} y_{il} \mathbf{w}^{(l)\,T} \tilde{\boldsymbol{\gamma}}_i^{\prime}+\sum_{i=1}^{N} \log \left[\sum_{l=1}^{C} \exp \left(\mathbf{w}^{(l)\,T} \tilde{\boldsymbol{\gamma}}_i^{\prime}\right)\right] \\
&+\frac{1}{2} \zeta_{\mathrm{w}}\|\mathbf{W}\|_F^2 . \qquad \text{(B3)}
\end{aligned}
$$

Hence, the corresponding equation for the E-step is now defined as

$$
\begin{aligned}
&Q(\boldsymbol{\Delta} \mid \boldsymbol{\Delta}^{(t)}) \propto -\frac{1}{2} \sum_{i=1}^{N}\left\{\left(\mathbf{x}_i-\langle \mathbf{z}_i \rangle \boldsymbol{\Lambda}^T\right) \boldsymbol{\Sigma}^{-1}\left(\mathbf{x}_i-\langle \mathbf{z}_i \rangle \boldsymbol{\Lambda}^T\right)^T\right. \\
&\left.+\operatorname{tr}\left[\boldsymbol{\Lambda}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Lambda}\left(\left\langle \mathbf{z}_i \mathbf{z}_i^T\right\rangle-\langle \mathbf{z}_i \rangle \langle \mathbf{z}_i \rangle^T\right)\right]\right\} \\
&-\frac{N+\eta+2}{2} \sum_{j=1}^{G} \log \sigma_j^2-\sum_{j=1}^{G} \frac{\eta \xi}{2 \sigma_j^2}-\sum_{k=1}^{K^*} \log p\left(\boldsymbol{\lambda}^{\mathbf{k}}\right) \\
&-\frac{1}{2} \sum_{i=1}^{N}\left\{\langle \mathbf{z}_i \rangle \mathbf{D}_i \langle \mathbf{z}_i \rangle^T+\operatorname{tr}\left[\mathbf{D}_i\left(\left\langle \mathbf{z}_i \mathbf{z}_i^T\right\rangle-\langle \mathbf{z}_i \rangle \langle \mathbf{z}_i \rangle^T\right)\right]\right\} \\
&-\frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{K^*} \log \tau_{ik} \\
&-\frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{K^*}\left[\left(1-\langle \tilde{\gamma}_{ik} \rangle\right) \widetilde{\omega}_0^2+\langle \tilde{\gamma}_{ik} \rangle \widetilde{\omega}_1^2\right] \tau_{ik} \\
&+\sum_{k=1}^{K^*}\left[\left(\sum_{i=1}^{N}\langle \tilde{\gamma}_{ik} \rangle\right) \log \prod_{l=1}^{k} v_l\right.
\end{aligned}
$$

$$+\left(N-\sum_{i=1}^{N}\langle\tilde{\gamma}_{ik}\rangle\right)\log\left(1-\prod_{l=1}^{k}v_l\right)\Bigg]$$
$$+(\tilde{\alpha}-1)\sum_{k=1}^{K^*}\log v_k-\sum_{i=1}^{N}\sum_{l=1}^{C}y_{il}\mathbf{w}^{(l)\,T}\langle\widetilde{\boldsymbol{\gamma}}_i'\rangle$$
$$+\sum_{i=1}^{N}\left\langle\log\left[\sum_{l=1}^{C}\exp\left(\mathbf{w}^{(l)\,T}\widetilde{\boldsymbol{\gamma}}_i'\right)\right]\right\rangle+\frac{1}{2}\zeta_{\mathrm{w}}\|\mathbf{W}\|_F^2, \tag{B4}$$

and the variables at which the new $Q(\boldsymbol{\Delta}\mid\boldsymbol{\Delta}^{(t)})$ will be maximised in the M step are now $\boldsymbol{\Delta}=\{\mathbf{Z},\boldsymbol{\Sigma},\mathbf{T},\mathbf{W},\nu\}$.

Appendix C SOUL algorithm: further details

After defining in Section 3.4 the mathematical structure of the steps involved in estimating the hyperparameter ζ_{w} for the ℓ_2 regularisation term in (5), below we present the SOUL algorithm.

Algorithm 1 SOUL algorithm for the estimation of ζ_{w}

Input: $\zeta_{\mathrm{w}}^{(0)}\in\Pi_{\Theta_{\zeta_{\mathrm{w}}}}$, $W^{(0,0)}\in\mathbb{R}^{(K+1)\times C}$, δ_{ULA}, $\delta_{\mathrm{PGA}}^{(0)}\in\mathbb{R}$, $m,n\in\mathbb{N}$

1: **for** $i=1$ **to** n **do**
2: **if** $i>1$ **then**
3: Set $W^{(i,0)}=W^{(i-1,m)}$,
4: **end if**
5: **for** $j=0$ **to** $m-1$ **do**
6: $Z^{(i,j+1)}\sim N(0,\mathbf{I}_{(K+1)\times C})$
7: $W^{(i,j+1)}=W^{(i,j)}-\delta_{\mathrm{ULA}}\nabla_W\log p(W^{(i,j)}\mid\mathbf{Y},\tilde{\mathbf{\Gamma}}',\zeta_{\mathrm{w}}^{(i-1)})+\sqrt{2\delta_{\mathrm{ULA}}}Z^{(i,j+1)}$
8: **end for**
9: $\zeta_{\mathrm{w}}^{(i)}=\Pi_{\Theta_{\zeta_{\mathrm{w}}}}\left[\zeta_{\mathrm{w}}^{(i-1)}+\frac{\delta_{\mathrm{PGA}}^{(i-1)}}{2m}\sum_{j=1}^{m}\left\{\frac{(K+1)\times C}{\zeta_{\mathrm{w}}^{(i-1)}}-\|W^{(i,j)}\|_F^2\right\}\right]$
10: **end for**

Output: $\hat{\zeta}_{\mathrm{w}}^{(n)}=\sum_{i=1}^{n}u^{(i)}\zeta_{\mathrm{w}}^{(i)}/\sum_{i=1}^{n}u^{(i)}$

C.1 SOUL implementation guidelines

The implementation guidelines and details about the SOUL algorithm can be found in De Bortoli et al. (2021) and in (Vidal et al. 2020, Section 3.3). For completeness, we will provide some details below.

C.1.1 Setting $\delta_{\mathrm{PGA}}^{i}$ and m

It is suggested in Vidal et al. (2020) to set $\delta_{\mathrm{PGA}}^{(i)}=C_0 i^{-p}$ where p is within the range $[0.6,0.9]$ (in our experiments, we set $p=0.8$) and $C_0\in\mathbb{R}$ a constant that can be initially set as $(\zeta_{\mathrm{w}}^{(0)}\times(K+1)\times C)^{-1}$ and adjusted as needed. For m, we followed the recommendation in De Bortoli et al. (2021); Vidal et al. (2020) using a single sample per iteration (that is, $m=1$), as we did not observe significant differences with larger values of m.

C.1.2 Setting $u^{(i)}$

According to the guidelines in Vidal et al. (2020), it is recommended to set $u^{(i)}$ as follows:

$$u^{(i)}=\begin{cases}0 & \text{if } i<N_0,\\ 1 & \text{if } N_0\le i\le n,\end{cases}$$

where $N_0\in\mathbb{N}$ is the number of initial iterations to be discarded to reduce non-asymptotic bias, which corresponds to a burn-in stage. The range $i\in[N_0,n]$ represents the estimation phase of the averaging where the values of $\zeta_{\mathrm{w}}^{(i)}$ have reached convergence and stabilised. In the interest of computational efficiency, we have set in our experiments $N_0=75$ and $n=150$.

C.1.3 Implementation in Logarithmic Scale

The proposed methods for estimating ζ_{w} generally achieve better numerical convergence when implemented on a logarithmic scale, as recommended in (Vidal et al. 2020, Section 3.3.2). Therefore, we apply a variable transformation $\kappa=\log(\zeta_{\mathrm{w}})$, estimate $\hat{\kappa}$ using the SOUL algorithm, and then determine $\hat{\zeta}_{\mathrm{w}}=e^{\hat{\kappa}}$.

This variable transformation necessitates a slight modification in the gradient calculations, which must be scaled by $e^{\kappa^{(n)}}$ to adhere to the chain rule. For instance, step 9 in Algorithm 1 is updated to

$$\kappa^{(i+1)}=\Pi_{\Theta_\kappa}\Bigg[\kappa^{(i)}+e^{\kappa^{(i)}}\frac{\delta_{\mathrm{PGA}}^{(i+1)}}{2m}\sum_{j=1}^{m}\left\{\left((K+1)\times C\times e^{-\kappa^{(i)}}\right)-\|W^{(i,j)}\|_F^2\right\}\Bigg],$$

where $\Theta_\kappa=\{\log(\zeta_{\mathrm{w}}):\zeta_{\mathrm{w}}\in\Theta_{\zeta_{\mathrm{w}}}\}$ represents the permissible range of κ values taken logarithmically.

Appendix D Gradient and Step Size Computation for SOUL and AGD

D.1 Gradient of ridge multinomial logistic regression

In the ULA step of the SOUL method (see (9) and Algorithm 1) and the computation of $\hat{\mathbf{W}}$ (see Section 3.5), we need to compute the gradient of the logarithm of the ridge

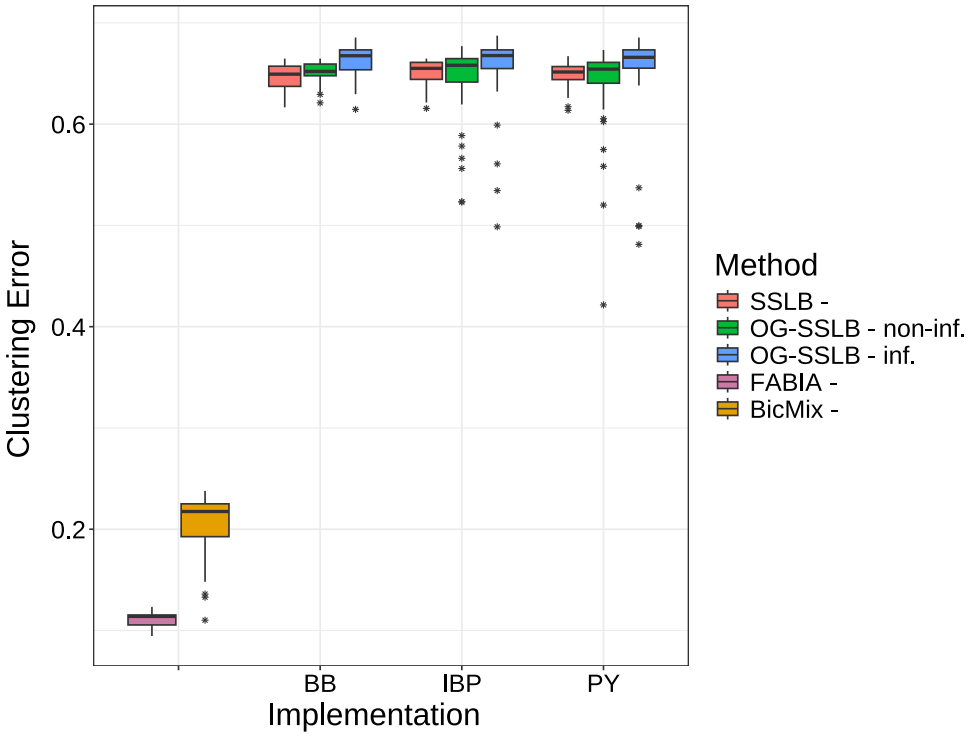

Fig. 11 Clustering error of 50 replicates for FABIA, BicMix, SSLB, and OG-SSLB. For (OG-)SSLB, we include results under the three prior choices for $\tilde{\mathbf{\Gamma}}$: BB, PY, and IBP (see Section 2.1.2). OG-SSLB results are shown for both the non-informative and informative settings described in Section 4.1 and in the caption of Figures 4.

multinomial logistic regression model, that is

$$\nabla_{\mathbf{w}} \log p(\mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \mathbf{W}, \hat{\zeta}_{\mathrm{w}}) = \nabla_{\mathbf{w}} \left[\sum_{i=1}^{N} \sum_{l=1}^{C} y_{il} \mathbf{w}_l^T \langle \widetilde{\boldsymbol{\gamma}}_i' \rangle + \sum_{i=1}^{N} \left\langle \log \left[\sum_{l=1}^{C} \exp\left(\mathbf{w}_l^T \widetilde{\boldsymbol{\gamma}}_i' \right) \right] \right\rangle + \frac{1}{2} \hat{\zeta}_{\mathrm{w}} \|\mathbf{W}\|_F^2 \right].$$

This gradient is a well-known result in the literature (see, e.g., (Hastie et al. 2009, Ex. 4.4)). Since it involves an expectation that does not have a closed-form solution, we must compute a Monte Carlo estimate of this log-sum-exp expectation term beforehand. This process was previously explained in Section 3.3. The gradient is then given by

$$\begin{aligned} \nabla_{\mathbf{w}} \log p(\mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \mathbf{W}, \hat{\zeta}_{\mathrm{w}}) &\approx -\frac{1}{J} \sum_{j=1}^{J} \widetilde{\mathbf{\Gamma}}'^{(j)\mathsf{T}} \left[p\left(\mathbf{Y} \mid \tilde{\mathbf{\Gamma}}'^{(j)}, \mathbf{W}, \hat{\zeta}_{\mathrm{w}} \right) - \mathbf{Y} \right] + \hat{\zeta}_{\mathrm{w}} \mathbf{W} \\ &:= \overline{\nabla}_{\mathbf{w}} \log p(\mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \mathbf{W}, \hat{\zeta}_{\mathrm{w}}), \end{aligned}$$

where we have generated a collection of $\tilde{\mathbf{\Gamma}}'^{(1)}, \ldots, \tilde{\mathbf{\Gamma}}'^{(J)}$ samples from $p(\widetilde{\gamma}_{ik} = 1 \mid \mathbf{Y}, \mathbf{T}, \widetilde{\boldsymbol{\theta}}, \mathbf{W})$ to compute the Monte Carlo estimate, as explained in Section 3.3. In addition, $p(\mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \mathbf{W}, \hat{\zeta}_{\mathrm{w}})$ is defined in (5). We have empirically found that $T = 30$ is enough to reach a good accuracy level while maintaining a low computational cost.

D.2 Step size δ_{ULA} & δ_{AGD}

Determining suitable step sizes for the ULA step in the SOUL method (see (9) and Algorithm 1) and AGD algorithm (see Section 3.5) is essential for guaranteeing convergence and computational efficiency. The literature offers well-established guidance on selecting the step size δ_{AGD}. Specifically, $\delta_{\mathrm{AGD}} \leq 1/L_f$ where L_f is the Lipschitz constant of $\nabla_{\mathbf{w}} \log p(\mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \mathbf{W}, \hat{\zeta}_{\mathrm{w}})$.

The value of L_f can be obtained from the Hessian of the objective function. For multinomial logistic regression (Böhning 1992), L_f is given by

$$L_f = \lambda_{\max} \left[\frac{1}{2} \left(\mathbf{I}_C - \mathbf{1}_C \mathbf{1}_C^T / C \right) \otimes \tilde{\mathbf{\Gamma}}'^T \tilde{\mathbf{\Gamma}}' \right] + \zeta_{\mathrm{w}},$$

where C is the number of classes of the multinomial logistic regression model. This ensures that the AGD update step remains within a stable region, facilitating steady progress towards the optimal solution.

For the ULA in the SOUL method, when $\nabla_{\mathbf{w}} \log p(\mathbf{Y} \mid \tilde{\mathbf{\Gamma}}', \mathbf{W}, \hat{\zeta}_{\mathrm{w}})$ is Lipschitz continuous with Lipschitz constant L_f, it has been shown that $\delta_{\mathrm{ULA}} \in (0, 2/L_f]$ ensures that the Markov chain $(W^{(k)})_{k \geq 0}$ described in (9) is ergodic with stationary distribution close to the true target distribution (Durmus and Moulines 2017). Therefore, following both AGD and ULA specifications, we have decided to set $\delta_{\mathrm{AGD}} = \delta_{\mathrm{ULA}} = 0.95/L_f$.

Appendix E Simulation Study: Clustering Error

To complement the evaluation based on consensus scores presented in Section 4.1, we also assessed the accuracy of the recovered biclusters using the *Clustering Error* (CE) metric (Horta and Campello 2014; Nicholls and Wallace 2021). This metric quantifies the similarity between predicted and true biclusters based on the overlap of matrix elements, and is commonly used in benchmarking biclustering algorithms when ground truth is available.

Let $A_1, \ldots, A_p$ be the true biclusters and $B_1, \ldots, B_q$ the predicted biclusters. The clustering error is defined as:

$$\mathrm{CE} = \frac{d_{\max}}{|U|},$$

where $d_{\max}$ is the total number of matrix elements correctly matched across an optimal pairing of biclusters, and $|U|$ is the size of the union of all matrix elements involved in either the predicted or true biclusters.

Figure 11 shows the clustering similarity scores across 50 replicates for each method, including the FABIA and BicMix algorithms. As with the consensus score results in Section 4.1, the informative OG-SSLB implementation outperforms SSLB, FABIA and BicMix across all prior settings (BB, PY, IBP).

Acknowledgements CW and LAVM are funded by the Wellcome Trust (WT220788). CW and PDWK are supported by the Medical Research Council (MC_UU_00040/01 and MC_UU_00040/05, respectively).

Author Contributions All authors contributed equally to this work and reviewed the manuscript.

Data Availability The implementation of the OGSSLB algorithm can be found as an R/C++ package at https://github.com/luisvargasmieles/OGSSLB. The code that supports the findings of this paper is available on the GitHub page: https://github.com/luisvargasmieles/OGSSLB-examples. The dataset used for the Breast Cancer Microarray experiment is publicly available in the R package breastCancerNKI. The dataset for the Immune Cell Gene Expression Atlas (ImmuNexUT) experiment is also in the public domain and is available at the National Bioscience Database Center (NBDC), with the study accession code E-GEAD-397.

Declarations

Competing Interests Chris Wallace is a part-time employee of GSK. GSK had no role in this study or the decision to publish.

References

Armagan, A., Clyde, M., Dunson, D.: Generalized beta mixtures of gaussians. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 24, pp. 523–531. Curran Associates Inc, Granada, Spain (2011)

Bair, E.: Semi-supervised clustering methods. WIREs Comput. Stat. **5**(5), 349–361 (2013). https://doi.org/10.1002/wics.1270

Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. Journal of the American Statistical Association 112(518), 859–877 (2017) https://doi.org/10.1080/01621459.2017.1285773

Beall, J., Li, H., Martin-Harris, B., Neelon, B., Elm, J., Graboyes, E., Hill, E.: Bayesian hierarchical profile regression for binary covariates. Stat. Med. **43**(18), 3432–3446 (2024). https://doi.org/10.1002/sim.10119

Böhning, D.: Multinomial logistic regression algorithm. Annals of the Institute of Statistical Mathematics 44, 197–200 (1992) https://doi.org/10.1007/BF00048682

Chipman, H.A., George, E.I., McCulloch, R.E.: Bart: Bayesian additive regression trees. The Annals of Applied Statistics **4**(1), 266–298 (2010). https://doi.org/10.1214/09-AOAS285

Chauvel, C., Novoloaca, A., Veyre, P., Reynier, F., Becker, J.: Evaluation of integrative clustering methods for the analysis of multi-omics data. Brief. Bioinform. **21**(2), 541–552 (2019). https://doi.org/10.1093/bib/bbz015

Carbonetto, P., Stephens, M.: Scalable variational inference for bayesian variable selection in regression, and its accuracy in genetic association studies. Bayesian Anal. **7**(1), 73–108 (2012). https://doi.org/10.1214/12-BA703

Dalalyan, A.S.: Theoretical guarantees for approximate sampling from smooth and log-concave densities. Journal of the Royal Statistical Society. Series B (Statistical Methodology) 79(3), 651–676 (2017)

De Bortoli, V., Durmus, A., Pereyra, M., Vidal, A.F.: Efficient stochastic optimisation by unadjusted langevin monte carlo: Application to maximum marginal likelihood and empirical bayesian estimation. Statistics and Computing 31, 1–18 (2021) https://doi.org/10.1007/s11222-020-09986-y

Durmus, A., Moulines, É.: Nonasymptotic convergence analysis for the unadjusted langevin algorithm. Ann. Appl. Probab. **27**(3), 1551–1587 (2017). https://doi.org/10.1214/16-AAP1238

Douc, R., Moulines, E., Stoffer, D.: Nonlinear Time Series: Theory. Methods and Applications with R Examples. Chapman & Hall/CRC, New York (2014)

Eren, K., Deveci, M., Küçüktunç, O., Çatalyürek, Ü.V.: A comparative analysis of biclustering algorithms for gene expression data. Brief. Bioinform. **14**(3), 279–292 (2012). https://doi.org/10.1093/bib/bbs032

Ghahramani, Z., Griffiths, T.: Infinite latent feature models and the indian buffet process. In: Weiss, Y., Schölkopf, B., Platt, J. (eds.) Advances in Neural Information Processing Systems, vol. 18. MIT Press, Vancouver, BC, Canada (2005)

Gao, C., McDowell, I.C., Zhao, S., Brown, C.D., Engelhardt, B.E.: Context specific and differential gene co-expression networks via bayesian biclustering. PLoS Comput. Biol. **12**(7), 1–39 (2016). https://doi.org/10.1371/journal.pcbi.1004791

Güler, O.: New proximal point algorithms for convex minimization. SIAM J. Optim. **2**(4), 649–664 (1992). https://doi.org/10.1137/0802032

Gong, Y., Xu, J., Wu, M., Gao, R., Sun, J., Yu, Z., Y., Z.: Single-cell biclustering for cell-specific transcriptomic perturbation detection in ad progression. Cell Reports Methods **4**(4) (2024) https://doi.org/10.1016/j.crmeth.2024.100742

Hochreiter, S., Bodenhofer, U., Heusel, M., Mayr, A., Mitterecker, A., Kasim, A., Khamiakova, T., Van Sanden, S., Lin, D., Talloen, W., Bijnens, L., Göhlmann, H.W.H., Shkedy, Z., Clevert, D.A.: Fabia: factor analysis for bicluster acquisition. Bioinformatics **26**(12), 1520–1527 (2010). https://doi.org/10.1093/bioinformatics/btq227

Horta, D., Campello, R.J.G.B.: Similarity measures for comparing biclusterings. IEEE/ACM Trans. Comput. Biol. Bioinf. **11**(5), 942–954 (2014). https://doi.org/10.1109/TCBB.2014.2325016

Hastie, T., Tibshirani, R., Friedman, J.: Linear Methods for Classification, pp. 101–137. Springer, New York, NY (2009). https://doi.org/10.1007/978-0-387-84858-7_4

Koestler, D.C., Marsit, C.J., Christensen, B.C., Karagas, M.R., Bueno, R., Sugarbaker, D.J., Kelsey, K.T., Houseman, E.A.: Semi-supervised recursively partitioned mixture models for identifying cancer subtypes. Bioinformatics **26**(20), 2578–2585 (2010). https://doi.org/10.1093/bioinformatics/btq470

Kamperman, R.G., Veldkamp, S.R., Evers, S.W., Lim, J., Schaik, I., Royen-Kerkhof, A., Wijk, F., Kooi, A.J., Jansen, M., Raaphorst, J.: Type i interferon biomarker in idiopathic inflammatory myopathies: associations of siglec-1 with disease activity and treatment response. Rheumatology **64**(5), 2979–2986 (2024). https://doi.org/10.1093/rheumatology/keae630

Lundberg, I., Helmers, S.B.: The type i interferon system in idiopathic inflammatory myopathies. Autoimmunity **43**(3), 239–243 (2010). https://doi.org/10.3109/08916930903510955

Love, M.I., Huber, W., Anders, S.: Moderated estimation of fold change and dispersion for rna-seq data with deseq2. Genome Biology **15**(12) (2014) https://doi.org/10.1186/s13059-014-0550-8

Liverani, S., Hastie, D.I., Azizi, L., Papathomas, M., Richardson, S.: Premium: An r package for profile regression mixture models using dirichlet processes. Journal of Statistical Software 64(7), 1–30 (2015) https://doi.org/10.18637/jss.v064.i07

Levitin, E.S., Polyak, B.T.: Constrained minimization methods. USSR Comput. Math. Math. Phys. **6**(5), 1–50 (1966). https://doi.org/10.1016/0041-5553(66)90114-5

Meng, L., Avram, D., Tseng, G., Huo, Z.: Outcome-guided sparse k-means for disease subtype discovery via integrating phenotypic data with high-dimensional transcriptomic data. J. R. Stat. Soc.: Ser. C: Appl. Stat. **71**(2), 352–375 (2022). https://doi.org/10.1111/rssc.12536

Molitor, J., Papathomas, M., Jerrett, M., Richardson, S.: Bayesian profile regression with an application to the national survey of children's health. Biostatistics **11**(3), 484–498 (2010). https://doi.org/10.1093/biostatistics/kxq013

Mesko, B., Poliska, S., Nagy, L.: Gene expression profiles in peripheral blood for the diagnosis of autoimmune diseases. Trends Mol. Med. **17**(4), 223–233 (2011). https://doi.org/10.1016/j.molmed.2010.12.004

Moran, G.E., V., R., George, E.I.: Spike-and-slab lasso biclustering. The Annals of Applied Statistics 15(1), 148–173 (2021)

Nesterov, Y.: A method for solving the convex programming problem with convergence rate o(1/k2) (1983)

Nikolakis, D., Garantziotis, P., Sentis, G., Fanouriakis, A., Bertsias, G., Frangou, E., Nikolopoulos, D., Banos, A., Boumpas, D.T.: Restoration of aberrant gene expression of monocytes in systemic lupus erythematosus via a combined transcriptome-reversal and network-based drug repurposing strategy. BMC genomics **24**(207) (2023) https://doi.org/10.1186/s12864-023-09275-8

Nicholls, K., Kirk, P.D.W., Wallace, C.: Bayesian clustering with uncertain data. bioRxiv (2022). https://doi.org/10.1101/2022.12.07.519476

Nicholls, K., Wallace, C.: Comparison of sparse biclustering algorithms for gene expression datasets. Briefings in Bioinformatics **22**(6) (2021) https://doi.org/10.1093/bib/bbab140

Oyelade, J., Isewon, I., Oladipupo, F., Aromolaran, O., Uwoghiren, E., Ameh, F., Achas, M., Adebiyi, E.: Clustering algorithms: Their application to gene expression data. Bioinformatics and Biology Insights 10, 237–253 (2016) https://doi.org/10.4137/BBI.S38316

Ota, M., Nagafuchi, Y., Hatano, H., Ishigaki, K., Terao, C., Takeshima, Y., Yanaoka, H., Kobayashi, S., Okubo, M., Shirai, H., Sugimori, Y., Maeda, J., Nakano, M., Yamada, S., Yoshida, R., Tsuchiya, H., Tsuchida, Y., Akizuki, S., Yoshifuji, H., Ohmura, K., Mimori, T., Yoshida, K., Kurosaka, D., Okada, M., Setoguchi, K., Kaneko, H., Ban, N., Yabuki, N., Matsuki, K., Mutoh, H., Oyama, S., Okazaki, M., Tsunoda, H., Iwasaki, Y., Sumitomo, S., Shoda, H., Kochi, Y., Okada, Y., Yamamoto, K., Okamura, T., Fujio, K.: Dynamic landscape of immune cell-specific gene regulation in immune-mediated diseases. Cell **184**(11), 3006–3021 (2021). https://doi.org/10.1016/j.cell.2021.03.056

Padilha, V.A., Campello, R.J.G.B.: A systematic comparative evaluation of biclustering techniques. BMC Bioinformatics **18**(55) (2017) https://doi.org/10.1186/s12859-017-1487-1

Peeters, R.: The maximum edge biclique problem is np-complete. Discret. Appl. Math. **131**(3), 651–654 (2003). https://doi.org/10.1016/S0166-218X(03)00333-0

Perez, R.K., Gordon, M.G., Subramaniam, M., Kim, M.C., Hartoularos, G.C., Targ, S., Sun, Y., Ogorodnikov, A., Bueno, R., Lu, A., et al.: Single-cell rna-seq reveals cell type–specific molecular and genetic associations to lupus. Science **376**(6589) (2022) https://doi.org/10.1126/science.abf1970

Ročková, V., George, E.I.: Emvs: The em approach to bayesian variable selection. Journal of the American Statistical Association 109(506), 828–846 (2014) https://doi.org/10.1080/01621459.2013.869223

Ročková, V., George, E.I.: The spike-and-slab lasso. Journal of the American Statistical Association 113(521), 431–444 (2018) https://doi.org/10.1080/01621459.2016.1260469

Rouanet, A., Johnson, R., Strauss, M., Richardson, S., Tom, B.D., White, S.R., Kirk, P.D.W.: Bayesian profile regression for clustering analysis involving a longitudinal response and explanatory variables. J. R. Stat. Soc.: Ser. C: Appl. Stat. **73**(2), 314–339 (2023). https://doi.org/10.1093/jrsssc/qlad097

Robinson, M.D., McCarthy, D.J., Smyth, G.K.: edger: a bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics **26**(1), 139–140 (2009). https://doi.org/10.1093/bioinformatics/btp616

Roberts, G.O., Tweedie, R.L.: Exponential convergence of langevin distributions and their discrete approximations. Bernoulli **2**(4), 341–363 (1996)

Saelens, W., Cannoodt, R., Saeys, Y.: A comprehensive evaluation of module detection methods for gene expression data. Nature Communications **9**(1) (2018) https://doi.org/10.1038/s41467-018-03424-4

Salzo, S., Villa, S.: Inexact and accelerated proximal point algorithms. J. Convex Anal. **19**(4), 1167–1192 (2012)

Teh, Y.W., Grür, D., Ghahramani, Z.: Stick-breaking construction for the indian buffet process. In: Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 2, pp. 556–563. PMLR, San Juan, Puerto Rico (2007)

Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. Bioinformatics 18, 136–144 (2002) https://doi.org/10.1093/bioinformatics/18.suppl_1.S136

Vidal, A.F., De Bortoli, V., Pereyra, M., Durmus, A.: Maximum likelihood estimation of regularization parameters in high-dimensional inverse problems: An empirical bayesian approach part i: Methodology and experiments. SIAM J. Imag. Sci. **13**(4), 1945–1989 (2020). https://doi.org/10.1137/20M133982

Vijver, M.J., He, Y.D., Veer, L.J., Dai, H., Hart, A.A., Voskuil, D.W., J., S.G., Peterse, J.L., Roberts, C., Marton, M.J., Parrish, M., Atsma, D., Witteveen, A., Glas, A., Delahaye, L., Velde, T., H., B., S., R., Rutgers, E.T., Friend, S.H., Bernards, R.: A gene-expression signature as a predictor of survival in breast cancer. New England Journal of Medicine 347(25), 1999–2009 (2002) https://doi.org/10.1056/NEJMoa021967

Veer, L.J., Dai, H., Vijver, M.J., He, Y.D., Hart, A.A.M., Mao, M., Peterse, H.L., Kooy, K., Marton, M.J., Witteveen, A.T., Schreiber, G.J., Kerkhoven, R.M., Roberts, C., Linsley, P.S., Bernards, R., Friend, S.H.: Gene expression profiling predicts clinical outcome of breast cancer. Nature **415**(6871), 530–536 (2002). https://doi.org/10.1038/415530a

Wang, L., Zhang, H., Chang, H., Qin, Q., Zhang, B., Li, X., Zhao, T., Zhang, T.: Gaebic: A novel biclustering analysis method for mirna-targeted gene data based on graph autoencoder. J. Comput. Sci. Technol. **36**(2), 299–309 (2021). https://doi.org/10.1007/s11390-021-0804-3

Xie, J., Ma, A., Fennell, A., Ma, Q., Zhao, J.: It is time to apply biclustering: a comprehensive review of biclustering applications in biological and biomedical data. Brief. Bioinform. **20**(4), 1450–1465 (2018). https://doi.org/10.1093/bib/bby014

Xie, J., Ma, A., Zhang, Y., Liu, B., Cao, S., Wang, C., Xu, J., Zhang, C., Ma, Q.: Qubic2: a novel and robust biclustering algorithm for analyses and interpretation of large-scale rna-seq data. Bioinformatics **36**(4), 1143–1149 (2019). https://doi.org/10.1093/bioinformatics/btz692

Zhang, Y., Parmigiani, G., Johnson, W.E.: Combat-seq: batch effect adjustment for rna-seq count data. NAR Genomics and Bioinformatics **2**(3) (2020) https://doi.org/10.1093/nargab/lqaa078