Spike-and-Slab Lasso Biclustering

Gemma E. Moran [*], Veronika Ročková [†], and Edward I. George [‡]

August 24, 2020

Abstract

Biclustering methods simultaneously group samples and their associated features. In this way, biclustering methods differ from traditional clustering methods, which utilize the entire set of features to distinguish groups of samples. Motivating applications for biclustering include genomics data, where the goal is to cluster patients or samples by their gene expression profiles; and recommender systems, which seek to group customers based on their product preferences. Biclusters of interest often manifest as rank-1 submatrices of the data matrix. This submatrix detection problem can be viewed as a factor analysis problem in which both the factors and loadings are sparse. In this paper, we propose a new biclustering method called Spike-and-Slab Lasso Biclustering (SSLB) which utilizes the Spike-and-Slab Lasso of Ročková and George (2018) to find such a sparse factorization of the data matrix. SSLB also incorporates an Indian Buffet Process prior to automatically choose the number of biclusters. Many biclustering methods make assumptions about the size of the latent biclusters; either assuming that the biclusters are all of the same size, or that the biclusters are very large or very small. In contrast, SSLB can adapt to find biclusters which have a continuum of sizes. SSLB is implemented via a fast EM algorithm with a variational step. In a variety of simulation settings, SSLB outperforms other biclustering methods. We apply SSLB to both a microarray dataset and a single-cell RNA-sequencing dataset and highlight that SSLB can recover biologically meaningful structures in the data. The SSLB software is available as an R/C++ package at `https://github.com/gemoran/SSLB`.

1 Introduction

Standard clustering methods typically group samples based on their entire set of observed features. In large datasets, however, only a few features may play a role in distinguishing

[*]Data Science Institute, Columbia University, New York, NY 10027. Email: `gm2918@columbia.edu`
[†]Booth School of Business, University of Chicago, Chicago, IL 60637.
[‡]Department of Statistics, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104.

different clusters. As an example, consider the gene expression microarray dataset of Van De Vijver et al. (2002); Van't Veer et al. (2002), which we later revisit in Section 4. This data consists of the expression levels of $24,158$ genes from the breast cancer tumors of 337 patients with stage I or II breast cancer. Like many cancers, breast cancer is a heterogenous disease, comprising a number of sub-types which have contrasting prognoses and require different treatment regimens (Howlader et al., 2014). A patient's sub-type is typically determined based on their expression of hormone receptors (estrogen and progesterone) and human epidermal growth factor 2 (HER2) (Howlader et al., 2014).

Specifically, the goal is to group the patients into different sub-types based on their gene expression levels, where only a small fraction of their genes are expected to play a role in each sub-type. This is the problem of biclustering; simultaneously grouping both the samples, and the features associated with these samples. The benefits of such an approach to clustering are two-fold. First, biclustering can identify clusters which otherwise may not be found by using the entire feature set. Second, biclustering identifies which features are relevant for each cluster and so provides more interpretable solutions.

In this paper, we propose a new method for biclustering, which we call Spike-and-Slab Lasso Biclustering (SSLB). Before introducing our method and discussing related work, we preview the results of SSLB on the previously described dataset of Van De Vijver et al. (2002); Van't Veer et al. (2002). Figure 1 shows the gene expression microarray data before any clustering. We applied SSLB and re-ordered this gene expression matrix to correspond to one of the resulting SSLB biclusters (Figure 1, middle). This SSLB bicluster corresponds very well to patients' clinical estrogen receptor (ER) status, showing that SSLB can recover meaningful biological signal in the data. Note that this clinical information was not given to SSLB, which is an unsupervised method.

Later, we also apply SSLB to the single-cell RNA sequencing dataset of Zeisel et al. (2015). Zeisel et al. (2015) used single-cell RNA-sequencing (scRNA-seq) to obtain counts of RNA molecules in 3005 cells from the mouse somatosensory cortex and hippocampal CA1 region. The goal of their study was to characterize the RNA-expression levels in different cell-types of the mouse brain. Previously, cell types in the brain had been defined by alternative features such as location, morphology, and electrophysiological characteristics, combined with molecular markers (Zeisel et al., 2015). Defining cell-types instead by expression levels requires clustering both the cells and the genes; that is, biclustering.

Along with genomics data (Cheng and Church, 2000), biclustering methods have also been applied to recommender systems, which seek to group consumers based on their ratings of different products (De Castro et al., 2007; Zhu et al., 2016); neuroscience (Fan et al., 2010); and agriculture (Mucherino et al., 2009).

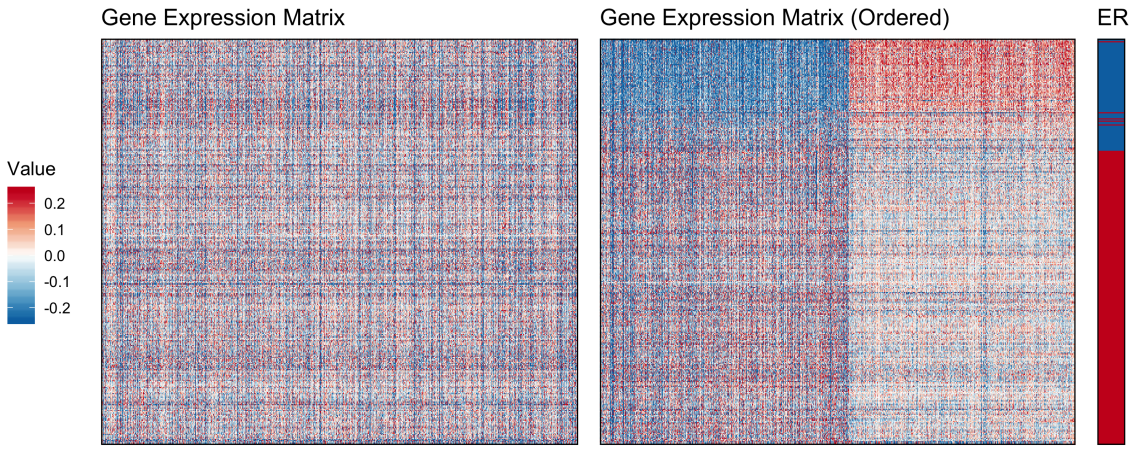


Figure 1: Left: submatrix of gene expression values (patients by genes). Middle: submatrix of gene expression values, rows and columns re-ordered to correspond to bicluster found by Spike-and-Slab Lasso Biclustering. Right: Clinical Estrogen Receptor (ER) status (Blue = ER-negative, Red = ER-positive).

1.1 Our Approach: Spike-and-Slab Lasso Biclustering

We now describe our proposed method, Spike-and-Slab Lasso Biclustering (SSLB). The observed data is the matrix of samples by features, denoted by

$$\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^T \in \mathbb{R}^{N \times G},$$

where Y_{ij} is the measurement of feature j in sample i for $i = 1, \ldots, N$, and $j = 1, \ldots, G$. The goal is to find submatrices of the data matrix (up to permutation of rows and columns) for which the elements Y_{ij} are "similar". The row and column indices of such a submatrix are then referred to as a "bicluster". Our method assumes that biclusters manifest as rank-1 submatrices of the data matrix, $\mathbf{Y}$. Intuitively, we seek samples which exhibit the same behavior, modulated by a scaling factor, on a subset of features.

This assumption corresponds to a factor analysis model where both the factors and the loadings are sparse. That is, we assume that $\mathbf{Y}$ has the following structure:

$$\mathbf{Y} = \sum_{k=1}^{K} \mathbf{x}^k \boldsymbol{\beta}^{kT} + \mathbf{E}, \tag{1.1}$$

where $\mathbf{X} = [\mathbf{x}^1, \ldots \mathbf{x}^K] \in \mathbb{R}^{N \times K}$ is the factor matrix, $\mathbf{B} = [\boldsymbol{\beta}^1, \ldots, \boldsymbol{\beta}^K] \in \mathbb{R}^{G \times K}$ is the loadings matrix and $\mathbf{E} = [\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_N]^T \in \mathbb{R}^{N \times G}$ is a matrix of Gaussian noise with $\boldsymbol{\varepsilon}_i \overset{ind}{\sim} N_G(\mathbf{0}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} = \text{diag}\{\sigma_j^2\}_{j=1}^G$ for $i = 1, \ldots, N$. We allow for the number of biclusters,

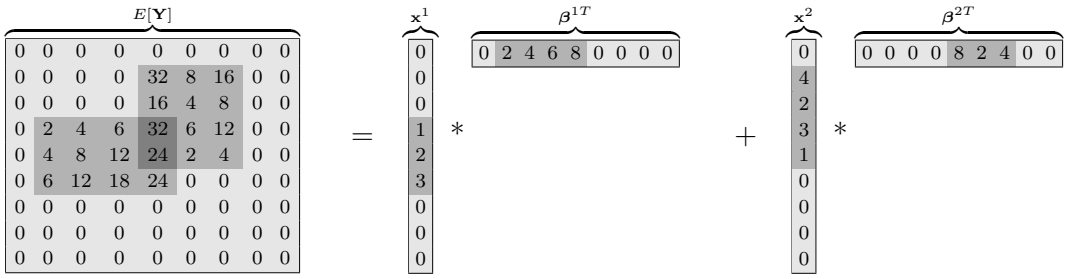


Figure 2: Mean of a data matrix with two biclusters: $E[\mathbf{Y}] = \mathbf{x}^1\boldsymbol{\beta}^{1T} + \mathbf{x}^2\boldsymbol{\beta}^{2T}$.

K to be unknown. We use the convention that the superscript $\mathbf{x}^k$ refers to the kth column of $\mathbf{X}$, and the subscript $\mathbf{x}_i$ to refers to the ith row of $\mathbf{X}$. The mean of a data matrix with two rank-1 biclusters is shown in Figure 2. An example of a data matrix with $K = 10$ biclusters is shown in Figure 3.

The benefits of utilizing a factor model for biclustering are threefold. Firstly, it is interpretable. Using gene expression data as an example: the genes (i.e. features) in a bicluster may be expressed at different levels to drive a biological process. This expression pattern in turn may be weaker or stronger in different samples, as determined by the sample-specific multiplicative effect. Secondly, there are many applications in which features and samples have been shown to be well approximated by such multiplicative effect models (Hochreiter et al., 2010). Thirdly, the definition allows for the specification of the model (1.1), allowing for systematic analysis of the noise variance and, in possible future work, coherent inclusion of prior information regarding the features or the samples.

In (1.1), x_{ik} is non-zero if sample i belongs to bicluster k and β_{jk} is non-zero if feature j belongs to bicluster k. As such, the problem of finding the biclusters in this framework can be viewed as a two-way variable selection problem: identifying biclusters corresponds to finding the support of $\mathbf{x}^k$ and $\boldsymbol{\beta}^k$. To address this problem, we adopt a Bayesian framework and place sparsity-inducing Spike-and-Slab Lasso priors (Ročková and George, 2018) on each of the columns of the factor matrix, $\mathbf{X}$, and of the loadings matrix, $\mathbf{B}$. The Spike-and-Slab Lasso was introduced by Ročková and George (2018) for variable selection in linear regression and has subsequently been used in grouped regression (Bai et al., 2019), multivariate regression (Deshpande et al., 2019) and sparse factor analysis (Ročková and George, 2016). A difference here from Ročková and George (2016) is that we induce sparsity in both the factor matrix and the loadings matrix, instead of only the loadings matrix. A benefit of the Spike-and-Slab Lasso is that it can adapt to the underlying levels of sparsity (or lack thereof) in the data. As we will show, this allows the method to find biclusters of a range of different sizes.

To determine the number of biclusters, K, we use a Bayesian nonparametric strategy. Specifically, we use an Indian Buffet Process prior (IBP, Griffiths and Ghahramani, 2011)

on the "size" of each bicluster, which ensures that each new bicluster is smaller than the previous one. We also allow for the IBP prior to be extended to a Pitman-Yor IBP (Teh et al., 2007), which drives the size of consecutive biclusters to decrease as a power law. This extension may be appropriate in applications where one expects a larger number of biclusters of a smaller size.

For implementation, we develop a fast, deterministic EM algorithm with a variational step to find the modal estimates of $\mathbf{X}$ and $\mathbf{B}$. Biclustering is in general NP-hard (Peeters, 2003). The Spike-and-Slab Lasso prior ameliorates such computational difficulties as it uses a continuous relaxation of bicluster membership.

We note that the factorization (1.1) is similar to the singular value decomposition (SVD) of $\mathbf{Y}$. However, the SVD assumption forces the columns of $\mathbf{X}$ and $\mathbf{B}$ to be orthogonal, a requirement which is relaxed here, as is done in factor analysis more generally. A benefit of not requiring orthogonality is that it allows for biclusters to overlap, enabling samples and features to belong to more than one bicluster. Further, samples and features do not have to belong to any biclusters.

A potential issue with not requiring orthogonality is that the model (1.1) is not identifiable up to rotation. Specifically, for a given solution pair $\{\mathbf{X}, \mathbf{B}\}$, one may rotate the matrices to obtain $\{\widetilde{\mathbf{X}} = \mathbf{PX}, \widetilde{\mathbf{B}} = \mathbf{PB}\}$, where $\mathbf{P}$ is any rotation matrix. Under the model (1.1), both $\{\mathbf{X}, \mathbf{B}\}$ and $\{\widetilde{\mathbf{X}}, \widetilde{\mathbf{B}}\}$ have equal likelihood, and so we cannot distinguish between the two solutions.

In factor analysis, this identifiability issue is often solved by placing hard constraints on the form of $\mathbf{B}$, such as a lower triangular requirement (Frühwirth-Schnatter and Lopes, 2010). We avoid placing hard constraints on the form of $\mathbf{X}$ or $\mathbf{B}$ and instead mitigate the identifiability issue by anchoring our priors on a sparse factorization of $\mathbf{Y}$. Such a sparse factorization is encouraged by both the IBP prior on the number of biclusters, K, and the sparsity-inducing priors on $\mathbf{X}$ and $\mathbf{B}$, as we will outline in Section 2. These sparsity priors softly constrain the posteriors of $\mathbf{X}$ and $\mathbf{B}$ away from rotational invariance; by Maxwell's theorem, the multivariate Gaussian is the only rotationally invariant distribution of independent variables (Maxwell (1860) and III, 4 in Feller (1971)). Consequently, a sparse factorization of $\mathbf{Y}$ is more likely to have a unique posterior probability and thus be identifiable (as there are few or zero rotation matrices which allow for the same posterior probability).

Recently, Rohe and Zeng (2020) formalized these arguments regarding identifiability in sparse factor models for a specific algorithm. Specifically, Rohe and Zeng (2020) prove that applying a Varimax rotation (Kaiser, 1958) to the principal components of the data matrix results in an identifiable solution, provided the true factors are sparse and independent. The Varimax rotation is constructed to find a coordinate basis (if any) in which the factors are generally axis-aligned, or sparse. In this work, we do not utilize such a Varimax rotation

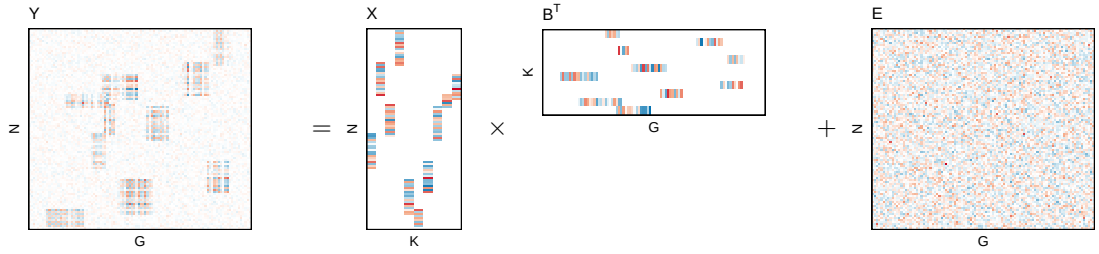


Figure 3: Simulated data with $K = 10$ biclusters each manifesting as a rank-1 submatrix in a data matrix: $\mathbf{Y} = \mathbf{X}\mathbf{B}^T + \mathbf{E}$.

to find sparse solutions, and instead encourage sparsity via Spike-and-Slab Lasso priors on both $\mathbf{X}$ and $\mathbf{B}$. This is because we seek a solution that is simultaneously sparse in $\mathbf{X}$ and $\mathbf{B}$, whereas a Varimax rotation often gives a sparse solution for $\mathbf{B}$ only. We conjecture that sparsity priors may play a similar role to the Varimax rotation in terms of finding identifiable sparse solutions, but formalizing this conjecture is beyond the scope of this work.

1.2 Related Work

A number of biclustering methods have utilized the same factor analysis model (1.1) with alternate sparsity-inducing priors for the factor and loading matrices. The first method to do so was Factor Analysis for Bicluster Acquisition (FABIA, Hochreiter et al., 2010) who placed single Laplace priors on both $\mathbf{x}^k$ and $\boldsymbol{\beta}^k$. However, the posterior resulting from Laplacian priors does not place enough mass on sparse solutions in variable selection problems (Castillo et al., 2015). This is because such a single Laplace prior has one variance parameter and so cannot both shrink negligible values to zero and maintain the larger signal. As a result, the estimates of $\mathbf{X}$ and $\mathbf{B}$ from FABIA are not sparse; the authors recommend a heuristic thresholding rule to then determine bicluster membership. In contrast, the Spike-and-Slab Lasso performs selective shrinkage on the latent variables; indeed the Spike-and-Slab Lasso concentrates at the optimal rate for sparse models (Ročková et al., 2018). Further, our method gives an indicator of bicluster membership, precluding the need for an arbitrary thresholding strategy. Finally, FABIA does not automatically select the number of biclusters, requiring this to be set in advance.

Gao et al. (2016) also begin with the model (1.1) for their method, BicMix. They allow for the components $\mathbf{x}^k$ and $\boldsymbol{\beta}^k$ to be either sparse, or dense to account for potential confounders. To achieve strong regularization on the sparse components, the authors utilize a three parameter beta distribution (Armagan et al., 2011), a generalization of the horseshoe prior (Carvalho et al., 2010). Whilst this dichotomous framework may be appropriate in some applications, in other cases it may be more appropriate to allow for a continuum of sparsity

levels. Such a continuum is achieved in our model as the Spike-and-Slab Lasso prior is indexed by a continuous parameter which controls the proportion of non-zero values in each bicluster. Further, the Spike-and-Slab Lasso automatically thresholds negligible values to zero; such thresholding does not occur automatically for the horseshoe prior and generalizations thereof. Gao et al. (2016) also allow for the number of biclusters, K, to be unknown by starting with an overestimate of K, imposing strong regularization on $\mathbf{X}$ and $\mathbf{B}$, and then removing zero columns. This strategy is similar to to our Bayesian nonparametric strategy; the difference is that the IBP prior which we utilize increases the strength of the regularization of $\mathbf{X}$ and $\mathbf{B}$ as a function of the column number k, as opposed to BicMix which applies the same regularization to each column.

Recently, Denitto et al. (2017) proposed the similarly named method "Spike and Slab Biclustering". Despite this likeness, there are a number of differences between our methods. Firstly, Denitto et al. (2017) utilize Gaussians distributions for both their spike and slab priors, whereas we use Laplacian priors. In Bayesian variable selection, the slab distribution requires tails at least as heavy as the Laplace for optimal posterior concentration (Castillo and van der Vaart, 2012). Furthermore, Denitto et al. (2017) do not use a non-parametric strategy to estimate the number of biclusters, instead requiring this number to be set in advance. Finally, Denitto et al. (2017) use an augmented Lagrangian method which they note is not guaranteed to increase the EM objective function.

Up to this point, we have reviewed only biclustering methods which utilize the factor model (1.1). In the literature, there have been a variety of methods which use different notions of similarity to define biclusters. Generally speaking, these notions of similarity can be grouped into four categories, as outlined by Madeira and Oliveira (2004).

The first category assumes that biclusters manifest as submatrices of constant values (for example, Hartigan, 1972; Shabalin et al., 2009; Prelić et al., 2006). The second category extends this constant submatrix assumption to accommodate additive row and column bicluster-specific effects in a similar manner to two-way ANOVA (see, for example, Cheng and Church, 2000; Lazzeroni and Owen, 2002; Gu and Liu, 2008). A related method is COSA (Friedman and Meulman, 2004), which finds biclusters which minimize the interquartile range between samples on a subset of the features; this can be viewed as a robust version of the aforementioned methods which minimize the variance on a subset of features. The third category assumes multiplicative row and column effects, instead of additive. That is, biclusters are assumed to manifest as rank-1 submatrices in the data matrix, up to permutations of rows and columns. This category includes the factor analysis methods discussed above, and methods which rely on singular value decomposition (Kluger et al., 2003). Further methods in this category are those which utilize Pearson's correlation as a criterion for bicluster membership (Bozdağ et al., 2009; Bhattacharya and Cui, 2017) and Rangan et al. (2018), which uses a loop-counting method to find rank-1 submatrices. Methods in the fourth category do not assume a model for the data matrix but instead

search for patterns in the data matrix. Such patterns may be viewed as generalizations of the additive or multiplicative assumptions. For example, the Iterative Signature Algorithm (ISA, Bergmann et al., 2003) finds submatrices in which all rows and all columns are above a certain threshold. Ben-Dor et al. (2003) generalize the multiplicative effects assumption to find subsets of features which have the same order on a subset of samples, which can be thought of as a slightly more flexible correlation structure.

In addition to how they define biclusters, methods can also be classified according to other criteria, including: the types of algorithms they utilize to find such biclusters; the assumptions they make regarding the noise distribution; and whether features and samples are allowed to belong to more than one bicluster, to name a few. For more detailed reviews of biclustering methods, see Madeira and Oliveira (2004); Prelić et al. (2006); Bozdağ et al. (2010); Eren et al. (2012); Padilha and Campello (2017).

2 Hierarchical Model for SSLB

In this section, we outline the Spike-and-Slab Lasso Biclustering (SSLB) model in greater detail. We adopt the factor analysis model in (1.1). To allow for uncertainty in the number of biclusters, K, we initialize the factor and loading matrices with an overestimate, K^*. The IBP prior discourages biclusters with negligible signal from entering consideration, and so the estimated factor and loading matrices will contain columns of all zeroes, provided K^* is a true overestimate. After removing these zero columns, the number of remaining columns is the estimated number of biclusters.

We also restrict $\mathbf{X}$ and $\mathbf{B}$ to be matrices with at least two non-zero entries per column (Frühwirth-Schnatter and Lopes, 2010; Ročková and George, 2016). This avoids a singleton column in either $\mathbf{X}$ or $\mathbf{B}$ which would be unidentifiable with regard to the noise matrix $\mathbf{\Sigma}$ in the marginal covariance of $\mathbf{Y}$ (after marginalizing over either $\mathbf{B}$ or $\mathbf{X}$, respectively).

2.1 Hierarchical structure for loadings B

For each column $\boldsymbol{\beta}^k$, we have a Spike-and-Slab Lasso prior. That is, each β_{jk} is drawn *a priori* from either a Laplacian "spike" parameterized by λ_0 and is consequently negligible, or a Laplacian "slab" parameterized by λ_1 and thus can be large:

$$\pi(\beta_{jk}|\gamma_{jk}, \lambda_0, \lambda_1) = (1 - \gamma_{jk})\psi(\beta_{jk}|\lambda_0) + \gamma_{jk}\psi(\beta_{jk}|\lambda_1), \quad 1 \leq j \leq G,\ 1 \leq k \leq K^*, \quad (2.1)$$

where the Laplace density is denoted by $\psi(\beta|\lambda) = \frac{\lambda}{2}e^{-\lambda|\beta|}$ and γ_{jk} is a binary indicator variable. Here, $\gamma_{jk} = 1$ if feature j is active in bicluster k, and $\gamma_{jk} = 0$ if feature j has a negligible contribution to bicluster k. We allow for uncertainty in bicluster membership

by using the common Beta-Bernoulli prior for the latent indicators:

$$\begin{aligned} \gamma_{jk}|\theta_k &\sim \text{Bernoulli}(\theta_k), \\ \theta_k &\sim \text{Beta}(a, b). \end{aligned} \tag{2.2}$$

It is important to emphasize here the "sparsity-indexing" parameter θ_k. Due to the Beta-Bernoulli prior, it has a natural interpretation as the percentage of non-zero elements in the column $\boldsymbol{\beta}^k$. By allowing θ_k to vary continuously, the method can adapt to differing levels of sparsity in each of the different columns of $\mathbf{B} = [\boldsymbol{\beta}^1, \ldots, \boldsymbol{\beta}^K]$.

Here, we can use a finite approximation to the IBP by setting the hyperparameters of the Beta prior in (2.2) to: $a \propto 1/K^*$, $b = 1$ (Ghahramani and Griffiths, 2006). This ensures that in the limit as $K^* \to \infty$, this prior is the IBP. While this is the default choice for these hyperparameters, we note that they can be easily tailored to the problem at hand. For instance, a choice of $a = 1/G$, $b = 1/G$ will result in the prior mass concentrating around $\theta = 0$ and $\theta = 1$, which may be preferred when both very dense and very sparse biclusters are expected.

2.2 Hierarchical structure for factors X

To find biclusters, we also want sparsity in the columns of $\mathbf{X}$. To this end, we place a Spike-and-Slab Lasso prior on each x_{ik}. However, we require an alternate formulation of the Spike-and-Slab Lasso prior to Section 2.1 for the x_{ik} in order to yield a tractable EM algorithm. This is accomplished by introducing auxiliary variables $\{\tau_{ik}\}_{i,k=1}^{N,K^*}$ for the variance of each $\mathbf{x}_{ik}$:

$$x_{ik}|\tau_{ik} \sim N(0, \tau_{ik}) \quad 1 \leq i \leq N,\ 1 \leq k \leq K^*. \tag{2.3}$$

Then, the τ_{ik} are each assigned a mixture of exponentials prior, where τ_{ik} is drawn *a priori* from either an exponential "spike" parameterized by $\widetilde{\lambda}_0^2$ and consequently is small, or from an exponential "slab" parameterized by $\widetilde{\lambda}_1^2$ and hence can be large:

$$\pi(\tau_{ik}|\widetilde{\gamma}_{ik}) = \widetilde{\gamma}_{ik}\frac{\widetilde{\lambda}_1^2}{2}e^{-\widetilde{\lambda}_1^2\tau_{ik}/2} + (1-\widetilde{\gamma}_{ik})\frac{\widetilde{\lambda}_0^2}{2}e^{-\widetilde{\lambda}_0^2\tau_{ik}/2} \tag{2.4}$$

where $\widetilde{\gamma}_{ik}$ is a binary indicator variable. This augmentation strategy uses the fact that the Laplace distribution can be represented as a scale mixture of a normal with an exponential mixing density; marginalizing over the τ_{ik} yields the usual Spike-and-Slab Lasso prior in (2.1).

We place independent Bernoulli priors on each of the $\widetilde{\gamma}_{ik}$ binary indicators. Similarly as before, $\widetilde{\gamma}_{ik} = 1$ if sample i is active in bicluster k, and $\widetilde{\gamma}_{ik} = 0$ if sample i has a negligible

contribution to bicluster k. The Bernoulli priors are parameterized by the "sparsity indexing" parameters $\widetilde{\theta}_k$. Instead of placing a Beta prior on the $\widetilde{\theta}_k$ as for the hierarchical model for the loadings $\mathbf{B}$, we use an Indian Buffet Process prior with an optional Pitman-Yor extension. This is achieved using the stick-breaking construction of Teh et al. (2007):

$$\begin{aligned}
\widetilde{\gamma}_{ik} &\sim \text{Bernoulli}(\widetilde{\theta}_{(k)}), \\
\widetilde{\theta}_{(k)} &= \prod_{l=1}^{k} \nu_{(l)}, \\
\nu_{(k)} &\sim \text{Beta}(\widetilde{\alpha} + kd, 1-d), \qquad \text{where } d \in [0,1), \quad \widetilde{\alpha} > -d.
\end{aligned} \tag{2.5}$$

When $d = 0$, the above formulation is the usual IBP prior. When $0 < d < 1$, the ordered sparsity weights, $\widetilde{\theta}_{(k)}$, decrease in expectation as a $O(k^{-1/d})$ power-law (Teh et al., 2007). This may be useful in applications where there are expected to be more, but smaller, biclusters.

We note that we only utilize this stick-breaking formulation of the IBP prior for the sparsity weights for the factors, $\mathbf{X}$, and not the loadings, $\mathbf{B}$. This is because this formulation requires ordering the columns of $\mathbf{X}$ from most dense to least dense. There is no reason to assume that the bicluster with the largest number of samples (i.e. non-zero x_{ik}) would also have the largest number of features (i.e. non-zero β_{jk}). That is, the most dense column of $\mathbf{X}$ should not be forced to line up with the most dense column of $\mathbf{B}$, which would be the case if we used a similar stick-breaking construction for the priors of $\mathbf{B}$.

In the simulation studies in Section 3, we will also consider the finite approximation to the IBP for comparison. Similarly as for the loadings $\mathbf{B}$, this formulation has a Beta prior on the sparsity weights, $\widetilde{\theta}_k \sim \text{Beta}(\widetilde{a}, \widetilde{b})$ with $\widetilde{a} \propto 1/K^*$ and $\widetilde{b} = 1$.

To complete the model, we place an inverse gamma prior on the elements of the covariance matrix, $\mathbf{\Sigma}$:

$$\sigma_j^2 \sim IG\left(\tfrac{\eta}{2}, \tfrac{\eta\xi}{2}\right). \tag{2.6}$$

Finally, we use the notation $\mathbf{T} = \{\tau_{ik}\}_{i,k=1}^{N,K^*} \in \mathbb{R}^{N \times K^*}$, $\widetilde{\mathbf{\Gamma}} = \{\widetilde{\gamma}_{ik}\}_{i,k=1}^{N,K^*}$ and $\mathbf{D}_i = \text{diag}\{\tau_{i1}^{-1}, \ldots, \tau_{iK^*}^{-1}\}$.

2.3 Implementation

We develop an EM algorithm with a variational step to quickly target modes of the posterior. In the E-Step, we compute the expectation of the factors $\mathbf{X}$ and factor indicators $\widetilde{\mathbf{\Gamma}}$, conditional on the data and current values of the rest of the parameters. This step is rendered tractable by the augmentation strategy outlined in Section 2.2. In the M-Step, we marginalize over the loading indicators, $\mathbf{\Gamma}$, and use a coordinate ascent strategy to find

the modes of $\mathbf{B}$ (Ročková and George, 2018). For this algorithm, we also use the variance updates detailed by Moran et al. (2018). To maximize the parameters of the IBP prior, we implement a variational step with closed form updates inspired by Doshi et al. (2009). Further details of the algorithm are given in Appendix A.

We adopt a dynamic posterior exploration strategy for finding estimates of $\mathbf{B}$ (Ročková and George, 2018). Specifically, we hold the slab parameters λ_1 fixed and then gradually increase the spike parameter λ_0 along a "ladder" of values, propagating the solutions forward as "warm starts" for the next largest spike values in the ladder. As outlined by Ročková and George (2018), holding the slab parameter fixed serves to stabilize the large coefficients; this is in contrast to the Lasso, which shrinks the larger coefficients along with the small. Meanwhile, gradually increasing λ_0 over a ladder of values progressively thresholds negligible coefficients to zero.

For the factor matrix, $\mathbf{X}$, we modify this dynamic posterior exploration strategy slightly. As we are calculating the conditional mean of $\mathbf{X}$, values of x_{ik} that were previously zero may re-enter the bicluster for very large $\widetilde{\lambda}_0$. This phenomenon is illustrated in the following simple example: suppose the true value is $x_{ik} = 0.005$. Then, the contribution of sample i is essentially negligible and so x_{ik} should reasonably "belong" to the spike. However, if spike parameter is $\widetilde{\lambda}_0 = 200$, it is actually unlikely that x_{ik} was drawn from the spike distribution; this is because this $\widetilde{\lambda}_0$ corresponds to an extremely small spike variance of 5×10^{-5}. While this would also occur for the mean of $\mathbf{B}$, we are instead estimating the mode of $\mathbf{B}$. In modal estimation, this problem does not occur as previously thresholded values do not seem to re-enter the bicluster. Consequently, to estimate the mean of $\mathbf{X}$, we recommend a stopping rule for $\widetilde{\lambda}_0$. We have found that an effective data-driven strategy is to "freeze" $\widetilde{\lambda}_0$ at a value at which $\mathbf{X}$ is the most sparse, whilst continuing to increase λ_0 (the spike parameter for $\mathbf{B}$).

Alternatively, to obtain an idea of what the order of $\widetilde{\lambda}_0$ should be, one may use the following informal empirical Bayes strategy. First, pre-determine a "negligible" value of x_{i1}, a value which one would expect to be too small to meaningfully contribute to a bicluster. Then, set the maximum value of $\widetilde{\lambda}_0$ to that which gives x_{i1} a prior probability of belonging to the "spike" equal to 0.5. Specifically, for a prior guess of θ_1 and a pre-specified value of $\widetilde{\lambda}_1 = 1$, solve for $\widetilde{\lambda}_0$ in:

$$P(\widetilde{\gamma}_{i1} = 0 | x_{i1}) = \frac{\widetilde{\theta}_1 \exp\{-\widetilde{\lambda}_1 |x_{i1}|\}}{\widetilde{\theta}_1 \exp\{-\widetilde{\lambda}_1 |x_{i1}|\} + (1 - \widetilde{\theta}_1) \exp\{-\widetilde{\lambda}_0 |x_{i1}|\}} = 0.5. \tag{2.7}$$

We also implement a re-scaling step for the columns of $\mathbf{X}$ and $\mathbf{B}$. Whilst sparsity-inducing priors mitigate to some extent the identifiability problems of the likelihood in regard to rotation, the scale of the columns of the factor and loadings matrices remains unidentifiable. That is, $\mathbf{x}^k \boldsymbol{\beta}^{kT}$ is equivalent to $(c_k^{-1}\mathbf{x}^k)(c_k \boldsymbol{\beta}^k)^T$ for any constant $c_k \in \mathbb{R}$. The focus of biclustering, however, is to find the non-zero elements of these matrices; it is the

covarying subsets that are of interest, and not their magnitude. As the scale is not of particular interest, we re-scale $\mathbf{X}$ and $\mathbf{B}$ at each step of the EM algorithm to ensure that the corresponding columns have the same norm. That is, for each $k = 1, \ldots, K$, we set

$$c_k \leftarrow \sqrt{\frac{\|\mathbf{x}^k\|_1}{\|\boldsymbol{\beta}^k\|_1}}, \quad \mathbf{x}^k \leftarrow \frac{1}{c_k}\mathbf{x}^k, \quad \boldsymbol{\beta}^k \leftarrow c_k\boldsymbol{\beta}^k. \tag{2.8}$$

The re-scaling step is also important to ensure that the default choices of regularization parameters $\lambda_0, \widetilde{\lambda}_0$ are appropriate; if $\mathbf{X}$ and $\mathbf{B}$ have vastly different scales, then one matrix may be over-thresholded whilst the other is under-thresholded.

The complexity of the SSLB algorithm is $O(NK^{*3} + GK^*)$, assuming that the initial number of biclusters, K^*, is less than both the number of samples, N, and the number of features, G. The first term comes from the E-Step for $\mathbf{X}$, where the $K^* \times K^*$ matrix $\mathbf{V}^i$ needs to be inverted for $i = 1, \ldots, N$. The second term comes from the M-Step for $\mathbf{B}$, where the coordinate ascent algorithm has complexity K^* and is applied to each of the G rows. However, the E-Step and M-Step are trivially parallelizable across the samples and features, respectively. Such a parallelization would yield an improved complexity of $O(K^{*3})$.

2.4 Automatic Thresholding

A key benefit of SSLB is that it automatically thresholds negligible elements of the loadings matrix, $\mathbf{B}$, to zero. This allows for a direct interpretation of bicluster membership: if the estimated $\widehat{\beta}_{jk} \neq 0$, then feature j is included in bicluster k. To determine bicluster membership for the samples, SSLB calculates the posterior mean of the indicator variables, $\widetilde{\boldsymbol{\Gamma}}$. The indicator $\widetilde{\gamma}_{ik}$ may be interpreted as the posterior probability that sample i belongs to bicluster k. If this posterior probability is greater than 0.5, we include sample i in bicluster k. More precisely, we implement the following thresholding rule after convergence of the SSLB algorithm:

$$\widehat{x}_{ik} = \begin{cases} \widehat{x}_{ik} & \text{if } E[\widetilde{\gamma}_{ik}|\mathbf{Y}, \mathbf{T}^*, \widetilde{\boldsymbol{\theta}}^*] > 0.5, \quad 1 \leq i \leq N,\ 1 \leq k \leq K^* \\ 0 & \text{if } E[\widetilde{\gamma}_{ik}|\mathbf{Y}, \mathbf{T}^*, \widetilde{\boldsymbol{\theta}}^*] \leq 0.5, \end{cases} \tag{2.9}$$

where $\mathbf{T}^*$ and $\widetilde{\boldsymbol{\theta}}^*$ are the solutions obtained after convergence of the EM algorithm. That is, if the posterior probability of x_{ik} belonging to the "spike" is greater than 0.5, it is thresholded to zero.

The natural thresholding scheme that arises from the SSLB model is in contrast to both FABIA and BicMix. FABIA utilizes an ad-hoc post-processing thresholding step, while the three-parameter beta prior of BicMix does not exactly threshold small values of the factors and loadings to zero.

2.5 Default Settings

The default hyper-parameters settings are as follows. For both the loadings and the factors, $\mathbf{B}$ and $\mathbf{X}$, the slab parameters are set to $\lambda_1, \widetilde{\lambda}_1 = 1$. The increasing ladder of spike parameters for $\mathbf{B}$ are set to $\lambda_0 \in \{1, 5, 10, 50, 100, 500, 10^3, 10^4, 10^5, 10^6, 10^7\}$. We set the spike parameters for $\mathbf{X}$ to $\widetilde{\lambda}_0 \in \{1, 5, \ldots, 5\}$ to match the length of the λ_0 sequence. Specifically, the $\widetilde{\lambda}_0$ sequence is frozen at $\widetilde{\lambda}_0 = 5$.

The above default values of λ_1 and λ_0 rely on a normalization of the data matrix, $\mathbf{Y}$. We implicitly normalize $\mathbf{Y}$ with a data-driven strategy to estimate the column variances, $\{\sigma_j^2\}_{j=1}^G$. Specifically, we use an informal empirical Bayes strategy, motivated by Chipman et al. (2010), to determine the hyper-parameters of the inverse-Gamma prior on σ_j^2 (2.6). This calibrates the prior towards values of σ_j^2 which are in accordance with the observed scale of the data. The intuition for our strategy is as follows: if we assume that most biclusters are sparse, then small values of the sample column variances, $\{s_j^2\}_{j=1}^G$, are essentially "pure noise" and contain no signal. Hence, the prior for σ_j^2 should be centered around a small value of the $\{s_j^2\}_{j=1}^G$. In addition, we recommend using a small value of the degrees of freedom parameter, η, to allow for prior uncertainty while avoiding too much probability near zero or in the tail. As a default, we take $\eta = 3$. More specifically, we calculate the 5% quantile of the s_j^2 and set the value of ξ such that this 5% quantile is the median of the prior distribution.

We initialize the parameters of SSLB as follows. Each entry of $\mathbf{B}$ is 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 on $\mathbf{X}$. The sparsity weights, θ_k, are initialized at 0.5. The IBP parameters, $\boldsymbol{\nu}$, are generated independently from a Beta$(1, 1)$ distribution and then ordered from largest to smallest.

For the initialization of K, we recommend $K^* = 50$ as an initial overestimate. If SSLB obtains a final estimate of $\widehat{K} = 50$ biclusters, this is an indication that the initial choice $K^* = 50$ underestimated the true number of biclusters; in this case, we recommend running SSLB again with a larger initial K^*.

3 Simulation Studies

In this section, we compare the performance of SSLB to a number of other biclustering methods in two simulation settings. The methods we compare are: (i) FABIA (Hochreiter et al., 2010); (ii) BicMix (Gao et al., 2016); (iii) spike-and-slab biclustering (SSBiEM, Denitto et al., 2017); (iv) Iterative Signature Algorithm (ISA, Bergmann et al., 2003); (v) Spectral (Kluger et al., 2003); and (vi) Plaid (Lazzeroni and Owen, 2002). Plaid belongs to the second biclustering algorithm category as outlined in Section 1.2, where biclusters

are assumed to have additive row and column effects. Methods (i), (ii), (iii) and (v) belong in the third biclustering category for which biclusters are assumed to manifest as rank-1 submatrices in the data matrix, up to permutations of rows and columns. Unlike the other methods, however, Spectral does not allow biclusters to overlap. ISA belongs to the fourth category of biclustering methods, and finds submatrices in which all rows and all columns are above a certain threshold.

Similarly to Gao et al. (2016), the simulation studies we present illustrate the performance of our method on settings with different levels of sparsity in the biclusters. Specifically, Simulation 1 considers matrices with only sparse biclusters, while Simulation 2 considers both sparse and dense biclusters. In Appendix D, we provide further simulation studies which investigate the performance of SSLB on Poisson-distributed data.

We use the following metrics to ascertain the quality of recovered biclusters: (i) relevance and recovery (Prelić et al., 2006); and (ii) consensus (Hochreiter et al., 2010) (see Appendix B for precise definitions). Relevance measures how similar on average the biclusters found by a method are to the true biclusters (where similarity is defined by the Jaccard index). Recovery instead measures how similar the true biclusters are to the found biclusters on average. However, if many duplicated biclusters are found by a method, this will not be reflected in either the relevance or recovery scores. To provide a meaningful metric in such circumstances, Hochreiter et al. (2010) developed the consensus score. The consensus score is similar to the recovery score, but penalizes overestimation of the true number of biclusters.

3.1 Simulation 1

We first consider a simulated example with $N = 300$, $G = 1000$ and $K = 15$ biclusters. The data was simulated using settings very similar to the FABIA paper (Hochreiter et al., 2010). Specifically, the data matrix $\mathbf{Y}$ was generated as $\mathbf{XB}^T + \mathbf{E}$, where each entry of the noise matrix $\mathbf{E}$ is sampled from an independent standard normal distribution. For each column $\mathbf{x}^k$, we draw the number of samples in bicluster k uniformly from $\{5, \ldots, 20\}$. The indices of these elements were randomly selected and then assigned a value from $N(\pm 2, 1)$, with the sign of the mean chosen randomly. The elements of $\mathbf{x}^k$ not in the bicluster had values drawn from $N(0, 0.2^2)$. The columns $\boldsymbol{\beta}^k$ were generated similarly, except the number of elements in each bicluster was drawn from $\{10, \ldots, 50\}$. We allow biclusters to share at most five samples and at least fifteen features. For both SSLB and BicMix, we set the initial overestimate of the number of biclusters to be $K^* = 30$. For FABIA and SSBiEM, which requires the number of biclusters to be set in advance, we set the number of biclusters to the truth, $K = 15$. For Plaid, we set the maximum number of biclusters to be 15. Both ISA and Spectral do not allow pre-specification of the number of biclusters.

In this simulation study, we compared three implementations of Spike-and-Slab Lasso Bi-

clustering: (i) SSLB with the Pitman-Yor extension where $\widetilde{\alpha} = 1$ and $d = 0.5$ (SSLB-PY); (ii) SSLB with the stick-breaking IBP prior for the factors where $\widetilde{\alpha} = 1$ (SSLB-IBP), and (iii) SSLB with the finite approximation to the IBP prior (i.e. Beta-Binomial) for the factors where $\widetilde{a} = 1/K^*$ and $\widetilde{b} = 1$ (SSLB-BB). For each implementation, we used the default settings as outlined in Section 2.5. For the loadings matrix, $\mathbf{B}$, we set the Beta-Binomial hyperparameters to be $a = 1/K^*$, $b = 1$.

For one realization from the above simulation setting, Figure 4 displays the support of the estimated factor and loadings matrices, $\mathbf{X}$ and $\mathbf{B}$, found by each of SSLB-IBP, BicMix and SSBiEM (see Appendix C.2 for plots for the remaining methods). SSLB-IBP finds the true bicluster structure with few false positives, while BicMix finds many more false positives due to small values not being exactly thresholded to zero by the three-parameter beta prior. SSBiEM also finds the true bicluster structure with few false positives, similarly to SSLB. However, SSBiEM requires the true number of biclusters to be known; when given a larger initial K^*, SSBiEM does not threshold the additional biclusters to zero (see Figure 14a in Appendix C.2). Further, SSLB is much faster than SSBiEM: on a single dataset from this simulation study, SSLB took 4 minutes to converge, while SSBiEM took 80 minutes. SSLB also achieves significantly better performance than SSBiEM on non-Gaussian distributed data (see Appendix D).

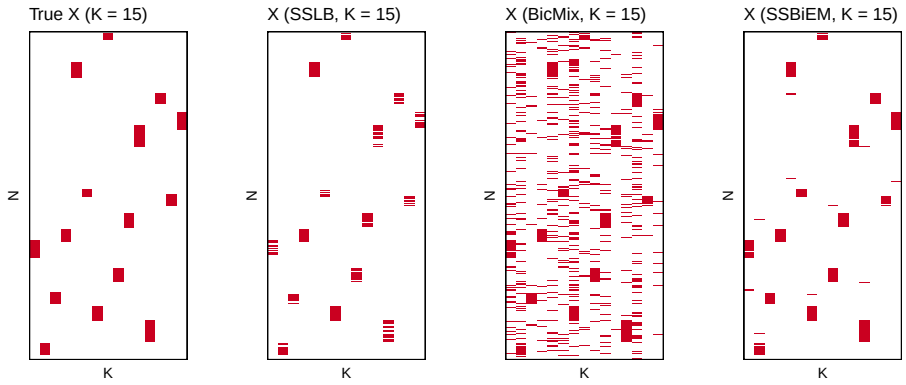
To further quantify the performance of each of the methods, we generated 50 realizations of the simulated data and calculated the consensus (Figure 5a), relevance and recovery scores (Figure 5b) for each method. SSBiEM has the highest consensus scores, followed by all versions of SSLB.

Table 1 displays the estimated number of biclusters, $\widehat{K}$, for the methods which estimate K. All implementations of SSLB very slightly overestimate the true number of biclusters, while BicMix slightly underestimates the true number of biclusters. ISA overestimates the true number biclusters by a wide margin. Spectral also overestimates the true number of biclusters. Finally, Plaid significantly underestimates the true number of biclusters; we hypothesize that this is because Plaid is designed to find only additive bicluster effects.

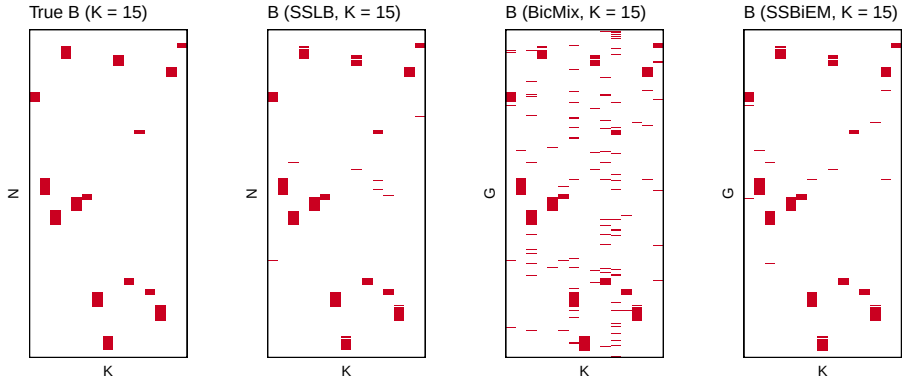
3.2 Simulation 2

We now assess how well SSLB can find both sparse and dense biclusters with a simulation study inspired by that of Gao et al. (2016). We again take $N = 300, G = 1000$ and $K = 15$. For both the factor and loading matrices, five columns are dense and ten columns are sparse. The sparse columns (corresponding to sparse biclusters) are generated as Simulation 1. The dense columns (corresponding to dense biclusters) are generated as independent $N(0, 2^2)$. We allow for one dense column in $\mathbf{X}$ to correspond to a sparse column in $\mathbf{B}$ and vice versa; this results in $K = 9$ biclusters which are sparse in both $\mathbf{X}$ and $\mathbf{B}$.

Figure 4: Simulation 1: Factor matrices, $\mathbf{X}$, and loading matrices, $\mathbf{B}$, found by different methods. Only the support of the matrix is displayed: a red value indicates a non-zero element.

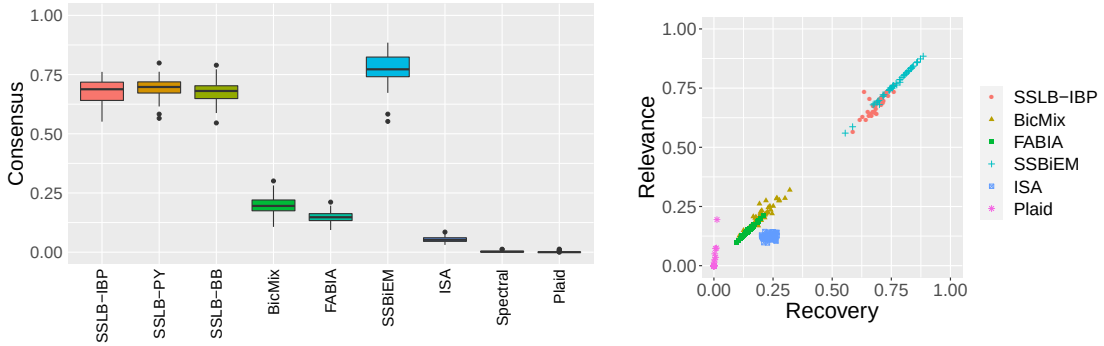


(a) Truth **(b)** SSLB-IBP **(c)** BicMix **(d)** SSBiEM



(e) Truth **(f)** SSLB-IBP **(g)** BicMix **(h)** SSBiEM

Figure 5: Simulation 1: (a) Boxplots of the consensus scores. (b) Relevance versus recovery scores.



(a) **(b)**

The goal for this simulation study is to recover the sparse biclusters while removing the effect of the dense biclusters, which are acting as confounders. As such, we calculate the recovery, relevance and consensus scores for the sparse biclusters found by each of the methods only. BicMix provides a binary indicator for whether $\mathbf{x}^k$ and $\boldsymbol{\beta}^k$ are sparse or dense; we kept BicMix biclusters for which both $\mathbf{x}^k$ and $\boldsymbol{\beta}^k$ were sparse. For the remaining methods, we determine a "sparse" bicluster to be one where both columns $\mathbf{x}^k$ and $\boldsymbol{\beta}^k$ have less than 50% of values being non-zero.

For one realization from the above simulation setting, Figure 6 displays the support of the estimated factor and loadings matrices, $\mathbf{X}$ and $\mathbf{B}$, found by SSLB-IBP, BicMix and SSBiEM (see Figure 13 in Appendix C.2 for plots of the remaining methods). SSLB finds nine of the ten true sparse biclusters with few false positives, successfully adapting to the dense and sparse structure. BicMix also finds nine out of ten true biclusters, albeit with more false positives. SSBiEM recovers the true bicluster structure very well; however, SSBiEM relies on knowing the true number of biclusters.

We again generated 50 realizations of the simulated data and calculated the consensus (Figure 5a), relevance and recovery scores (Figure 5b) for each method. SSBiEM achieves the highest consensus scores here, followed all versions of SSLB. The lower consensus scores of BicMix are again due to small values not being exactly thresholded to zero by the three-parameter beta prior. Meanwhile, the other biclustering methods are not able to find the true sparse bicluster structure.

In this setting, SSLB-IBP and SSLB-BB slightly overestimate the true number of biclusters (Table 1). SSLB-PY further overestimates the true number of biclusters; this is a result of SSLB-PY placing more prior weight on a larger number of small biclusters. BicMix approximates the true number of biclusters well, while ISA, Spectral and Plaid again

	$\widehat{K}$	
Method	Simulation 1	Simulation 2
Truth	*15*	*9*
SSLB-IBP	15.3 (0.10)	9.6 (0.11)
SSLB-PY	15.2 (0.07)	9.8 (0.10)
SSLB-BB	15.3 (0.09)	9.4 (0.10)
Bicmix	14.5 (0.18)	8.7 (0.10)
ISA	68.0 (2.01)	313.0 (5.34)
Spectral	17.9 (2.04)	1.7 (0.16)
Plaid	1.7 (0.26)	13.2 (0.45)

Table 1: Mean estimated number of biclusters, K, over 50 replications. Standard errors are shown in parentheses.

exhibit poor performance.

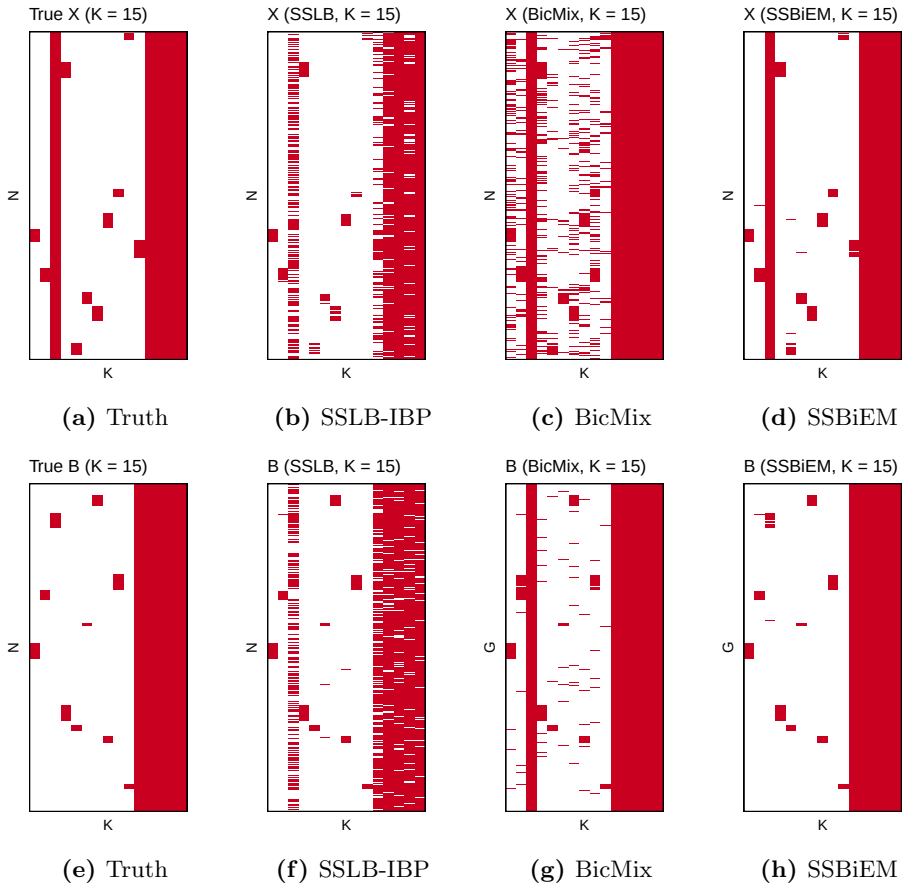
3.3 Simulation Summary

In Simulations 1 and 2, we highlighted the following benefits of SSLB.

- In contrast to BicMix, SSLB automatically thresholds small values of $\mathbf{X}$ and $\mathbf{B}$ to zero.
- In contrast to FABIA, ISA, Spectral and Plaid, SSLB can adapt to differing sparsity levels in the data.
- In contrast to SSBiEM, SSLB does not require the true number of biclusters to be known. Instead, SSLB estimates the number of biclusters from the data. In the simulation studies, SSLB approximates the true number of biclusters well.
- SSLB is computationally efficient, running 20 times as fast as SSBiEM.

Further, in Simulations 3 and 4 (Appendix D), SSLB exhibits the highest bicluster recovery performance out of all the methods when the data matrix, $\mathbf{Y}$, is distributed as a Poisson random variable (instead of a Gaussian).

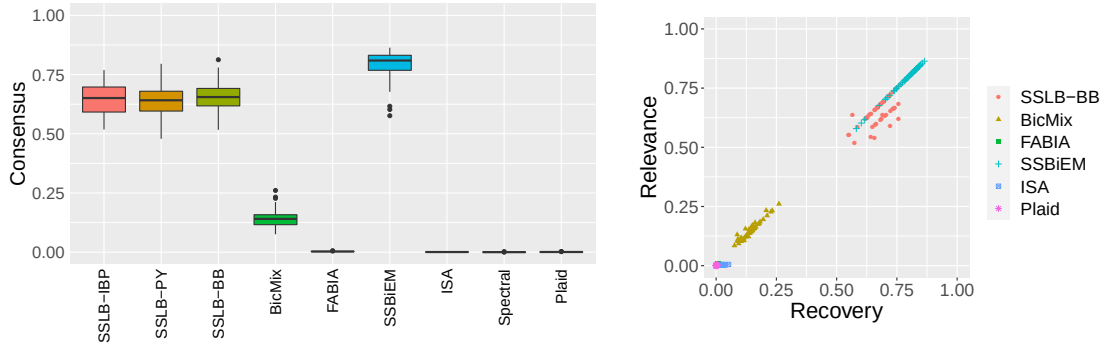
Figure 6: Simulation 2: Factor matrices, $\mathbf{X}$, and loading matrices, $\mathbf{B}$, found by each of the methods. Only the support of the matrix is displayed: a red value indicates a non-zero element.



(a) Truth **(b)** SSLB-IBP **(c)** BicMix **(d)** SSBiEM

(e) Truth **(f)** SSLB-IBP **(g)** BicMix **(h)** SSBiEM

Figure 7: Simulation 2: (a) Boxplots of the consensus scores. (b) Relevance versus recovery scores.



(a) **(b)**

4 Breast Cancer Microarray Dataset

We now return to the breast cancer microarray dataset from Section 1 and show how we obtain the ordered matrix in Figure 1. The dataset[1] consists of the expression levels of $G = 24,158$ genes from the breast cancer tumors of $N = 337$ patients with stage I or II breast cancer (Van De Vijver et al., 2002; Van't Veer et al., 2002). Gao et al. (2016) also used this dataset to illustrate the performance of their biclustering method, BicMix. We followed a similar data processing pipeline to Gao et al. (2016), without their pre-processing step (see Appendix E for details).

We ran SSLB-IBP with the initial number of biclusters set to $K^* = 50$. For the loadings, $\mathbf{B}$, we set the Beta-Binomial hyperparameters to $a = 1/(GK^*)$ and $b = 1$. This division by G places an added emphasis on sparsity. For the factors, $\mathbf{X}$, we set the IBP hyperparameter to $\widetilde{\alpha} = 1/N$ with $d = 0$. For the remaining parameters, we use the default settings outlined in Section 2.5. SSLB-IBP found $\widehat{K} = 30$ biclusters (Figure 8). The proportion of non-zero elements in $\mathbf{X}$ ranges from 31.8% to 0.6%. The proportion of non-zero elements in $\mathbf{B}$ ranges from 28.4% to 2.8% (Figure 21b, Appendix F). This suggests we are in a "sparse bicluster" regime, similar to Simulation 1, and so we do not need to filter dense biclusters.

4.1 SSLB identifies subtypes of breast cancer

Breast cancers can be broadly grouped into subtypes based on the expression levels of two genes: ESR1, which encodes an estrogen receptor (ER), and ERBB2, which encodes the human epidermal growth factor receptor 2 (HER2) (Howlader et al., 2014). A patient

[1]Data sourced from R package `breastCancerNKI` (Schroeder et al., 2011)

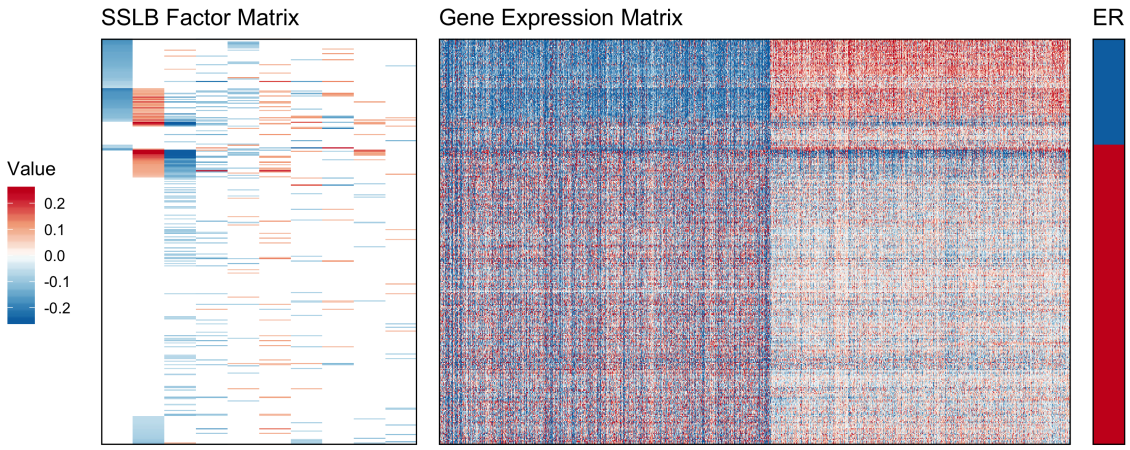


Figure 8: Left: SSLB factor matrix where each row corresponds to a patient and each column corresponds to a bicluster. A patient belongs to a bicluster if they have a non-zero value in that column. Rows are ordered by clinical ER status; within ER status, rows are ordered by factor values in biclusters 1 and 2. Only the first 10 biclusters (ordered by size) are shown for improved visualization; full factor matrix is displayed in Section F. Middle: submatrix of gene expression values where rows correspond to all samples (ordered by ER status) and columns correspond to genes in Bicluster 1 (re-ordered according to their loadings in Bicluster 1). Expression values with magnitude greater than 0.25 have had magnitude set to 0.25 for improved visualization. Right: Clinical Estrogen Receptor (ER) status (Blue = ER-, Red = ER+).

is deemed ER-positive (-negative) if they have relatively high (low) expression levels of ESR1. HER2 status is similarly defined by the expression of ERBB2. The expression levels of these genes determine four subtypes of breast cancer: (i) ER+/HER2+, (ii) ER+/HER2-, (iii) ER-/HER2+ and (iv) ER-/HER2-. These subtypes have been shown to be valuable prognostic indicators and are used to determine the treatment protocol for patients (Howlader et al., 2014). The clinical ER status of patients (determined by immunohistochemical staining, not gene expression levels) was provided with the dataset and so can provide a measure of validation for the biclusters that SSLB found. The HER2 status of patients was not recorded, however.

SSLB found four biclusters with significantly different means in the factors between the clinically ER-negative and ER-positive patients[2]. The patients with negative factors in SSLB bicluster 1 are almost all patients whose clinical status was recorded as ER-negative (Figure 8). We then investigated the genes in this bicluster and found ESR1, the gene encoding an estrogen receptor, was down-regulated for these patients. There are five patients with clinical ER-positive status who were in the ER-negative bicluster found by

[2]Biclusters 1, 2, 5 and 22 had p-values, 6.1×10^{-50}, 2.2×10^{-9}, 1.0×10^{-5} and 7.2×10^{-6}, respectively, from a Wilcoxon rank-sum test with Bonferroni significance level $0.01/\widehat{K}$

	ER+/HER2+	ER+/HER2-	ER-/HER2+	ER-/HER2-
Onitilo et al. (2009)	10.2%	68.9%	7.5%	13.4%
SSLB	7.7%	70.3%	8.9%	13.1%

Table 2: Proportion of breast cancer patients in each of the subtypes determined by ER and HER2 status from (i) the study of Onitilo et al. (2009); and (ii) SSLB.

SSLB. However, the down-regulation of the ESR1 gene in this patients suggests that the original clinical characterization was a misclassification. In the original paper analyzing this data, Van De Vijver et al. (2002) also found five patients had a discrepancy between their clinical ER-status and gene expression determined ER status, concluding that the latter classification was correct.

The gene ERBB2 is present in SSLB biclusters 1 and 2. In both biclusters, ERBB2 is up-regulated for patients with positive factors and down-regulated for patients with negative factors. For patients with negative bicluster 1 and zero bicluster 2 factors, ESR1 and ERBB2 are both down-regulated, indicating ER-/HER2- status. Meanwhile, patients with negative bicluster 1 and positive bicluster 2 factors are likely ER-/HER2+. Turning to the ER-positive patients (with zero bicluster 1 values), those with positive bicluster 2 values are potentially ER+/HER2+. Finally, ER-positive patients with negative bicluster 2 factors are likely ER+/HER2-. We note that a number of patients are in neither bicluster 1 or 2; we hypothesize that these patients are also ER+/HER2- as this is the most common breast cancer subtype (Onitilo et al., 2009). The proportions of patients in each subtype found by SSLB matches fairly well with reported subtype proportions in the literature (Table 2).

After determining these groups, we then investigated whether genes known to play a role in these subtypes were present in the biclusters. In particular, genes considered to be indicators (or markers) of ER+ status are KRT8, GATA-3, XBP-1, FOXA1 and ADH1B (Zhang et al., 2014). Four of these five marker genes were down-regulated in bicluster 1, and consequently were relatively over-expressed for the ER+ patients (p-value 0.002, Fisher's exact test). The gene GRB7 is located adjacent to the ERBB2 (HER2) gene and as such is often co-expressed with ERBB2; we indeed found that GRB7 was up-regulated in bicluster 2 (as well as down-regulated for the HER2- patients in bicluster 1).

4.2 Gene Ontology Enrichment Analysis

We next conducted gene ontology enrichment analysis on the genes found by SSLB using the R package `clusterProfiler` (Yu et al., 2012). This software conducts an overrepresentation test to determine whether genes which coordinate the same biological process are significantly co-occuring. If a subset of genes is found to be overrepresented in a set, the set is said to be "enriched" for the biological process in which those genes are active. With

a false discovery rate (FDR) threshold of 0.05, we found that the genes which were up-regulated in SSLB bicluster 1 (corresponding to the ER-negative patients) were enriched for 124 biological processes. Many of these were related to cell proliferation, including the G1/S transition of mitotic cell cycle. As cancer is fundamentally the un-regulated growth of cells, such proliferation signatures are commonly found in tumor samples (Whitfield et al., 2006). Another biological process for which the ER-negative bicluster is enriched is: response to leukemia inhibitory factor. Leukemia inhibitory factor has actually been shown to stimulate cell proliferation in breast cancer (Kellokumpu-Lehtinen et al., 1996). An enrichment map summarizing the most statistically significant processes is displayed in Figure 23a (Appendix F).

The genes up-regulated in the HER2+ patients in SSLB bicluster 2 were enriched for 495 biological processes (again with FDR threshold of 0.05). The enrichment map summarizing these processes is displayed in Figure 23b (Section F of the Appendix). In particular, these genes were enriched for the Wnt signaling pathway, the over-expression of which has been implicated in the development of cancer (Zhan et al., 2017). Further, stem cell proliferation was enriched in this bicluster; stem cells have been implicated as possible originators of tumors, and may in some cases potentially drive tumorigenesis (Reya et al., 2001).

Overall, 86.6% of the biclusters found by SSLB were enriched for biological processes. Further investigation of the remaining biclusters and their potential clinical utility may be interesting future work.

4.3 Comparison with other methods

We initially ran BicMix on this data using the default settings; however, BicMix found zero biclusters. It may be that BicMix is over regularizing on this data as it does not feature a mechanism to adapt to different noise levels. In contrast, SSLB uses an empirical Bayes-like strategy to adapt to the noise level in different datasets.

We then investigate how BicMix performed after projecting the quantiles of the data to the quantiles of a standard normal distribution, as recommended by Gao et al. (2016) . On this normalized data, BicMix found $\widehat{K} = 13$ biclusters, the first six of which had significantly different means for the ER-positive and ER-negative patients[3]. The gene ESR1, which encodes an estrogen receptor, was present only in BicMix bicluster 1. Unlike SSLB, many of the ER-positive patients in BicMix bicluster 1 have a negative x_{i1} factor value; this is likely an artifact of the quantile normalization. Marker genes for ER+ status (KRT8, GATA-3, XBP-1, FOXA1 and ADH1B) were not significantly up-regulated in BicMix bicluster 1. This may be due to the density of the bicluster; 36% of the 24,158 genes are present in bicluster 1. Meanwhile, the gene ERBB2, which encodes HER2, was

[3] p-values $9.9 \times 10^{-32}, 1.4 \times 10^{-20}, 1.3 \times 10^{-19}, 1.7 \times 10^{-16}, 3.4 \times 10^{-8}$, and 1.1×10^{-4}, respectively, from a Wilcoxon rank-sum test with Bonferroni significance level $0.01/\widehat{K}$

not present in any BicMix bicluster.

We additionally conducted gene ontology enrichment analysis on the genes found by BicMix, using the same settings as in Section . Only 53.8% of the biclusters found by BicMix were enriched for biological processes, compared to 86.6% for SSLB. We hypothesize that BicMix is combining much of the biological signal in bicluster 1. As such, we argue that SSLB finds more interpretable biclusters.

We ran FABIA on this dataset using the default settings for two different bicluster initializations: (i) $K = 10$ and (ii) $K = 30$ (the number of biclusters found by SSLB), as FABIA does not automatically select the number of biclusters (Figure 9a). In the $K = 10$ setting, FABIA found five biclusters that had a significantly different mean between ER+ and ER- patients[4]. Unlike SSLB, however, FABIA does not find a bicluster with almost exclusively ER-negative patients.

In the $K = 30$ setting, FABIA found four biclusters that had a significantly different mean between ER+ and ER- patients[5]. We can see that with a larger number of initial biclusters, the ER signal is diluted across multiple biclusters. As a result, the conclusions of FABIA seem to be highly dependent on the initial number of biclusters. Further, for this larger value of K, FABIA also does not find a bicluster consisting of almost exclusively ER-negative patients.

Next, we ran SSBiEM on this data with $K = 30$ biclusters. Surprisingly, SSBiEM found only two non-zero biclusters; this is in contrast to simulation studies where it did not threshold any biclusters exactly to zero. However, both biclusters found by SSBiEM were completely dense in both the factor and loading matrices, limiting their interpretability. One of the biclusters, however, did correspond to patient ER status (Figure 9c). We hypothesize that the poor performance of SSBiEM here is due to the recommended initialization of SSBiEM to the singular value decomposition of $\mathbf{Y}$. This resulted in very good performance in the simulation studies, but did not allow SSBiEM to find sparse, interpretable biclusters in this example.

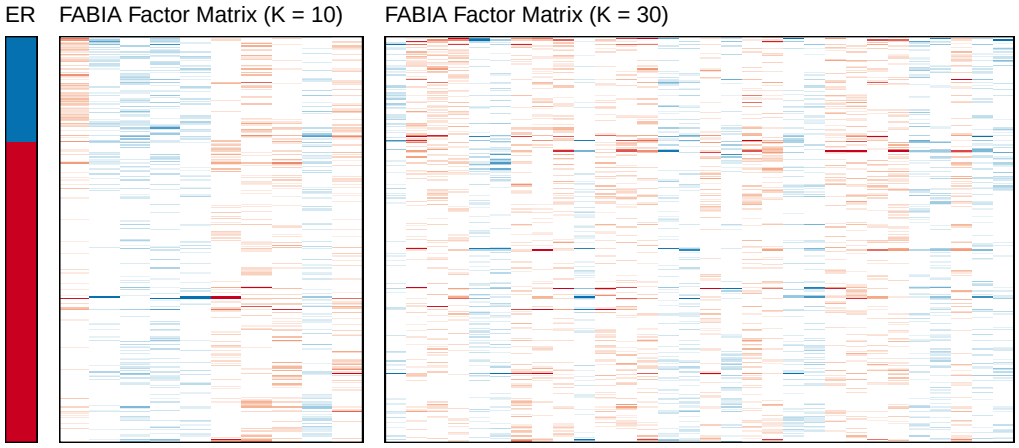
ISA found $\widehat{K} = 540$ biclusters on this data (Figure 9d). Given the propensity of ISA to overestimate the number of biclusters in the simulation studies, we anticipate that this is a significant overestimate of the actual number of biclusters. Of the 540 biclusters, 80 biclusters had a significantly different mean between the ER-positive and ER-negative patients.

Finally, neither Spectral nor Plaid returned any biclusters on this data.

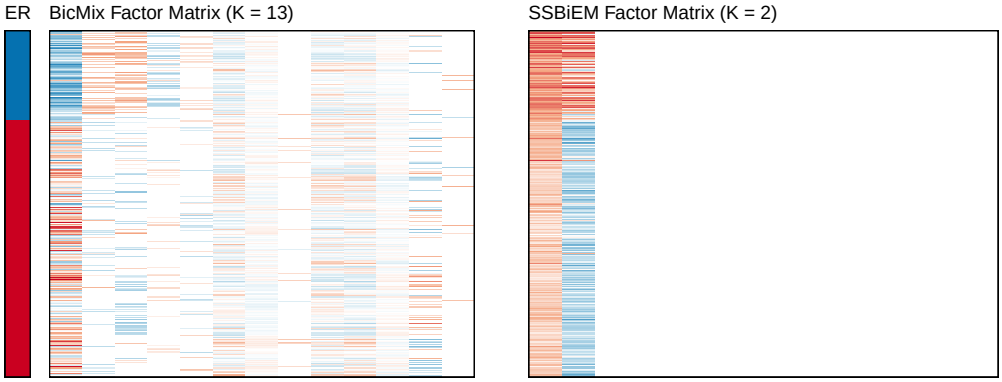
[4] p-values 3.9×10^{-24}, 2.5×10^{-12}, 9.2×10^{-10}, 4.8×10^{-8}, 1.7×10^{-7} from Wilcoxon rank-sum test with Bonferroni significance level 0.01/10

[5] p-values 2.5×10^{-9}, 1.7×10^{-7}, 6.5×10^{-6}, 3.6×10^{-5} from Wilcoxon rank-sum test with Bonferroni significance level 0.01/30

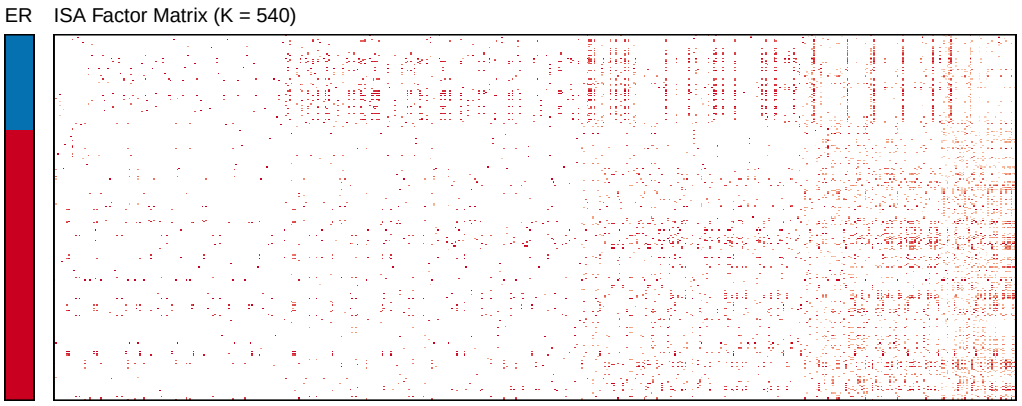
Figure 9: Breast Cancer Dataset: Results from FABIA, BicMix, SSBiEM and ISA. All matrices have rows ordered by clinical ER status.



(a) Left: Clinical Estrogen Receptor (ER) status (Blue = ER-, Red = ER+). Middle: FABIA factor matrix (initial $K^* = 10$). Right: FABIA factor matrix (initial $K^* = 30$)



(b) **(c)**



(d)

5 Mouse Cortex and Hippocampus scRNA-seq Dataset

For a second application, we assess the performance of SSLB on the data of Zeisel et al. (2015) (hereafter referred to as Z15). Z15 used single-cell RNA-sequencing (scRNA-seq) to obtain counts of RNA molecules in 3005 cells from the mouse somatosensory cortex and hippocampal CA1 region. The goal of the study was to characterize the RNA-expression levels in different cell-types of the mouse brain. Previously, cell types in the brain have been defined by alternative features such as location, morphology, and electrophysiological characteristics, combined with molecular markers Zeisel et al. (2015). Defining cell-types instead by expression levels requires clustering both the cells and the genes particularly associated with that cell cluster and as such is a biclustering problem.

Z15 developed a biclustering algorithm called BackSPIN which identified nine major types of cells in the mouse brain based on their transcription profiles: (i) interneurons; (ii) S1 pyramidal neurons; (iii) CA1 pyramidal neurons; (iv) oligodendrocytes; (v) microglia cells; (vi) endothelial cells; (vii) astrocytes; (viii) ependymal cells; and (ix) mural cells. By repeatedly applying BackSPIN on these biclusters, Z15 found a further 47 subclasses of cells. Here, we apply SSLB to the same dataset. A benefit of SSLB is that it can find classes and subclasses simultaneously without having to iteratively re-apply the method.

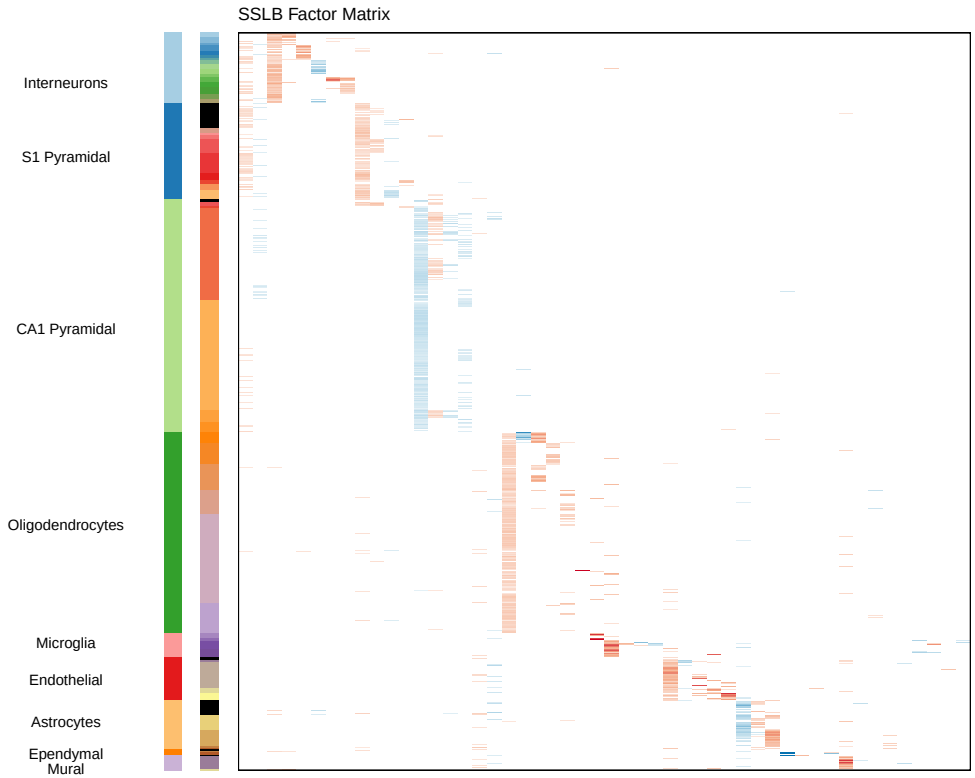
The scRNA-seq dataset made available by Z15 consists of RNA molecule counts for 19,972 genes in 3005 individual cells[6]. Following these authors, we (i) removed genes with less than 25 molecules in total over all cells; (ii) removed genes that were not correlated with more than 5 other genes; and (iii) retained the top 5000 most biologically variable genes. Further details of these processing steps are given in Appendix G. Although more sophisticated methods for removing technical variability in scRNA-seq data have been developed in recent years (for example, Huang et al., 2018), we follow the steps of Z15 to enable a direct comparison of our biclustering results.

After processing the data, the subset we used for biclustering is a matrix containing the RNA counts of $G = 5000$ genes in $N = 3005$ individual cells. We note that as a matrix of counts, this data is perhaps best modeled by a Poisson distribution, instead of assuming normally distributed residuals as in SSLB. However, Poisson-distributed data with a large rate parameter is approximately normal. As we are considering the most variable genes (with high RNA molecule counts), such a normal approximation is not too unreasonable. Despite this, there are still a high proportion of zero entries in the matrix and so this application may be seen as a test of the robustness of SSLB to model misspecification. We ran SSLB-IBP with the initial number of biclusters set to $K^* = 100$. as in Section 4, we set the Beta-Binomial hyperparameters to $a = 1/(GK^*)$ and $b = 1$, and the IBP hyperparameter to $\widetilde{\alpha} = 1/N$ with $d = 0$. For the remaining parameters, we use the default settings outlined in Section 2.5. SSLB returned $\widehat{K} = 95$ biclusters. The proportion of

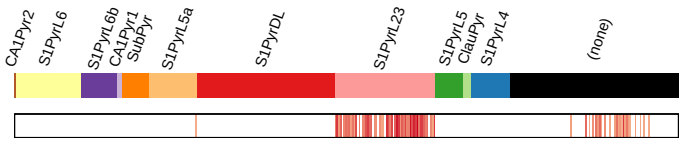
[6] `http://linnarssonlab.org/cortex`

non-zero elements in $\mathbf{X}$ ranges from 30% to 0.1%, and in $\mathbf{B}$ from 37% to 0.3% (Figure 25, Appendix H). This suggests we are in a "sparse bicluster" regime, similar to Simulation 1, and so we do not filter for dense biclusters.

Figure 10: Zeisel dataset: SSLB results



(a) Left: Cell types found by Z15. Middle: Cell subtypes found by Z15. The rows colored black were not assigned a subtype by Z15. Right: SSLB factor matrix with rows ordered to correspond to the Z15 cell types. Each row corresponds to a cell and each column corresponds to a bicluster. A cell belongs to a bicluster if they have a non-zero value in the bicluster (column). Factor values have been capped for improved visualization.



(b) "Zoom in" on S1 Pyramidal cells with subtypes annotated by Z15. Top: subtypes of S1 Pyramidal cells. Bottom: Column 10 of the SSLB factor matrix, corresponding to the cells in bicluster 10. SSLB groups a subset of the uncategorized "(none)" cells as of the S1PyrL23 subtype. (Colors have been modified from Figure 10a for improved visualization.)

5.1 SSLB recovers major cells types

SSLB recovered the nine major cell classes identified by Z15, finding a specific bicluster for each class except for the microglia class, which SSLB split into two biclusters (Figure 10a). For each class, Z15 also identified one or two potential marker genes; that is, a gene that is almost exclusively expressed in that cell class. Encouragingly, the SSLB biclusters corresponding to the major cell classes all contained the associated marker gene for that cell class. More specifically:

- The interneuron gene marker *Pnoc* was found in three SSLB biclusters, one corresponding to the major interneuron cell class and the others to subclasses of interneurons.
- The S1 pyramidal neuron marker genes *Gm*11549 and *Tbr*1 were present in two biclusters, one corresponding to the major S1 pyramidal neuron cell class and the other to a subclass of S1 pyramidal neurons. *Tbr*1 was also found in a bicluster containing cells from four different cell types, a potential false positive.
- The CA1 pyramidal neuron marker *Spink*8 was found in three biclusters. Two of these biclusters corresponded to the major CA1 pyramidal neuron cell class and a subclass of CA1 pyramidal neurons, respectively. The third bicluster contained CA1 pyramidal, S1 pyramidal and interneuron cells, suggesting that *Spink*8 may not necessarily be an exclusive marker for CA1 pyramidal neurons.
- The oligodendrocyte marker *Hapln*2 was active in three SSLB biclusters, all corresponding to either the major oligodendrocyte cell class or a subclass of oligodendrocytes. Interestingly, one of these biclusters contained 17 cells, all oligodendrocytes, but did not correspond to one of the Z15 identified subclasses; as such, this bicluster may correspond to a yet-to-be classified subtype of oligodendrocytes. Figure 28 shows the biological processes that are enriched in this bicluster, which can be broadly grouped into two categories: (i) processes related to oligodendrocyte-specific functions, including myelination, and (ii) cell metabolic processes.
- The endothelial cell marker *Ly*6*c*1 was found in four SSLB biclusters, two corresponding to the major endothelial group or a subclass. The other two biclusters were mostly all endothelial cells, but contained some astrocytes and microglia cells also.
- The mural cell marker *Acta*2 was active in three SSLB biclusters. One bicluster corresponded to the main mural bicluster and another to a bicluster with almost all mural cells. The third bicluster contained mostly endothelial cells, with a few oligodendrocyte, microglia, astrocyte and mural cells, indicating that either *Acta*2 is not exclusively expressed in mural cells, or a potential false positive of SSLB.

In addition to the nine main cell types, SSLB found two biclusters (biclusters 1 and 2) which contained many interneurons, S1 pyramidal neurons and CA1 pyramidal neurons. This is unsurprising as these cell types are all subsets of neurons, and so we would expect them to have more similar expression profiles than the other (non-neuronal) brain cells. We conducted gene ontology enrichment analysis on the genes SSLB found in these biclusters. With an FDR threshold of 0.05, bicluster 1 was enriched for 154 biological processes, the majority of which were related to cell metabolic processes and synaptic activity, as may be expected for neurons (Figure 27a). Bicluster 2 was similarly enriched for processes relating to synaptic activity, including axonal transport and synaptic signaling (Figure 27b).

The results of SSLB yield a number of observations that may warrant further scientific investigation. Firstly, while SSLB recovered the major cell types, it grouped together a number of the 47 sub-categories found by Z15. This was particularly the case for the interneuron cells, where SSLB found 5 subtypes (Z15 found 16), and the S1 pyramidal cells, where SSLB found 3 subtypes (Z15 found 12). It may be the case that SSLB has trouble finding more granular clusters, or potentially there really are fewer cell subtypes than identified by Z15.

Although SSLB collapsed many of the interneuron and S1 pyramidal subtypes, it found many more subtypes of microglia and ependymal cells than Z15. This suggests that there could be a great deal of heterogeneity in expression levels in these classes of cells, a phenomenon which may prove to be of scientific interest.

There are a number of cells which Z15 did not assign to a subtype (colored in black in Figure 10a). Interestingly, SSLB grouped a number of the previously unclassified S1 pyramidal cells into the "S1PyrL23" subtype (Figure 10b).

Finally, we conducted gene ontology enrichment analysis[7] for all of the biclusters found by SSLB. In this analysis, 83% of the biclusters identified by SSLB were enriched for at least one biological process.

5.2 Comparison with other methods

We also applied the other biclustering methods to the Zeisel dataset. Where an initial number of bicluster was required, we used $K^* = 100$. BicMix found $\widehat{K} = 94$ biclusters (Figure 11a). BicMix found many of the smaller subtypes defined by Z15 but assigns the major cell type signals to dense biclusters. This is a result of the dichotomous nature of BicMix; it finds either completely dense or very sparse biclusters. Consequently, BicMix may find "spuriously dense" biclusters. In contrast, SSLB can adapt to the underlying

[7] Using `clusterProfiler` with FDR threshold of 0.05. We took the 5000 genes obtained after processing as the "background" genes for the overrepresentation test instead of the original number of 19,972 to avoid selection bias.

sparsity, allowing it to also estimate such "medium"-sized biclusters. We argue that this allows SSLB to find more interpretable solutions.

ISA found $\widehat{K} = 107$ biclusters, a result reasonably consistent with both SSLB and BicMix (Figure 11b). We anticipate that this concordance is due to the better performance of ISA on Poisson data in simulation studies (see Appendix D) compared to Gaussian data (see Section 3). However, ISA finds multiple biclusters which correspond to the same cell type, unlike SSLB which generally finds a single cell type per major cell cluster. Further, ISA does not recover the cell subgroups as clearly as SSLB.

FABIA found $\widehat{K} = 99$ biclusters (Figure 24 in Appendix H). FABIA found many of the larger cell type biclusters but did not do well at recovering the more granular cell subtypes. This is due to FABIA having the same thresholding parameter for each bicluster; it is unable to adapt to the differing levels of sparsity.

The SSBiEM algorithm returned an error due to the size of the dataset, even with a smaller initial number of biclusters $K^* = 50$. Finally, neither Spectral nor Plaid found any biclusters on the Zeisel data.

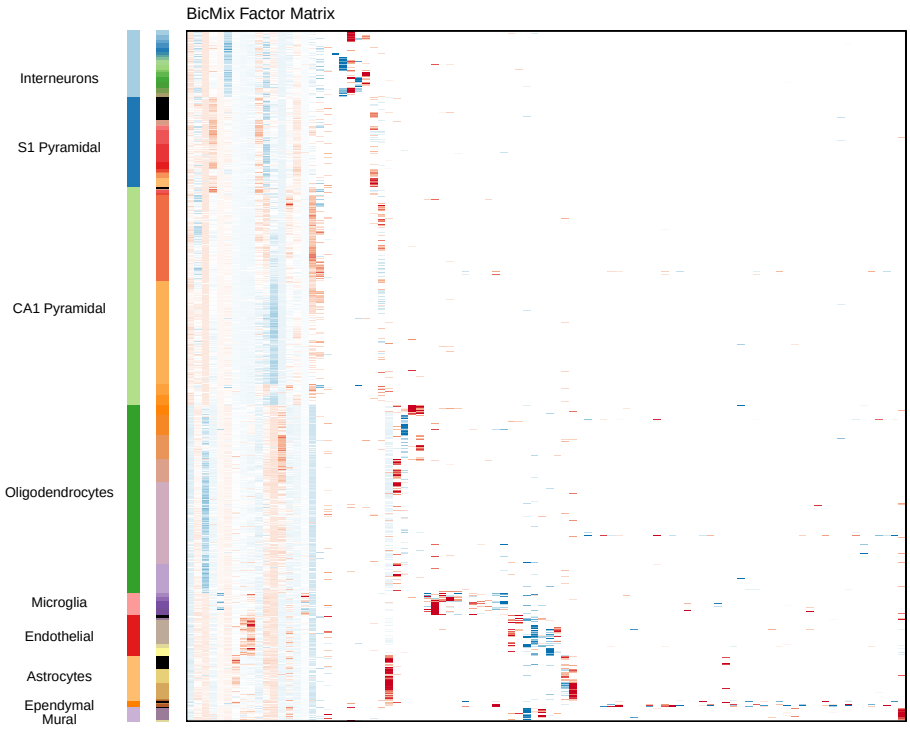
6 Conclusion

In this paper, we introduced a new method for biclustering called Spike-and-Slab Lasso Biclustering (SSLB). SSLB finds subsets of samples which co-vary on subset of features. These paired subsets manifest as rank-1 submatrices in the data, referred to as "biclusters" in this setting. To find these biclusters, SSLB performs two-way subset selection to conduct doubly-sparse factor analysis in which both the loadings and the factors are sparse. To induce this sparsity in the loadings and factors, SSLB uses the Spike-and-Slab Lasso prior of Ročková and George (2018). This prior is combined with an Indian Buffet Process prior to automatically choose the number of biclusters. SSLB utilizes a fast EM algorithm with a variational step to find the modes of the posterior. This EM algorithm is rendered tractable by a novel augmentation of the Spike-and-Slab Lasso prior.

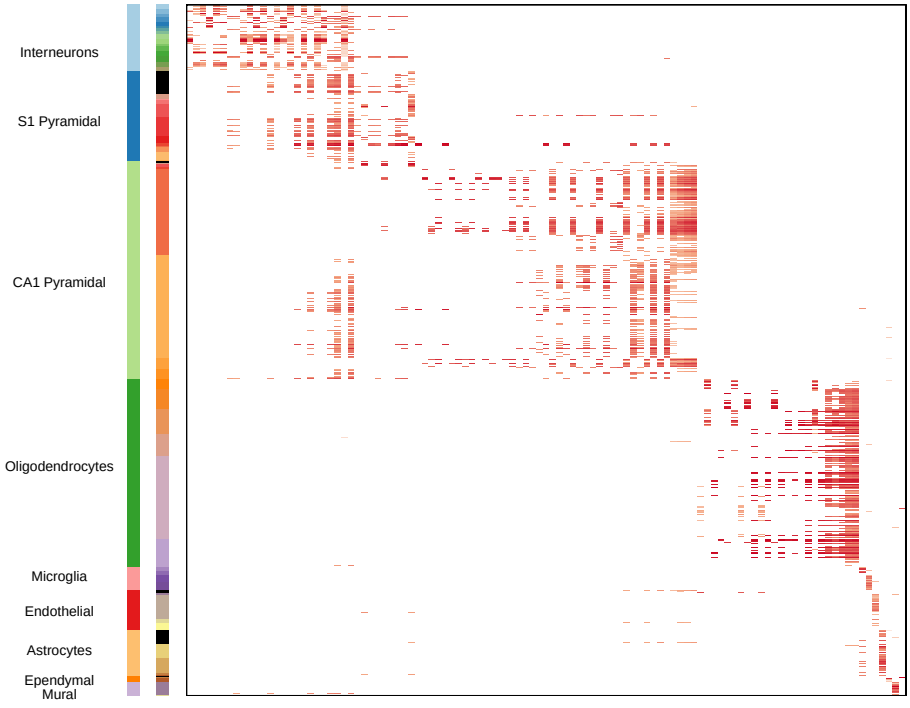
SSLB features a number of benefits over similar biclustering methods. Firstly, the adaptivity inherent in the Spike-and-Slab Lasso prior allows for SSLB to find a continuum of biclusters of different sizes. This is in contrast to other biclustering methods which have more restrictive assumptions on the sizes of the biclusters. Secondly, the Spike-and-Slab Lasso prior automatically thresholds negligible bicluster values to zero; this is unlike other biclustering methods which require post-processing thresholding steps. Finally, SSLB also demonstrates robustness to non-Gaussian distributed data, achieving higher bicluster consensus scores than alternative methods on simulated Poisson data.

SSLB out-performs a number of alternative biclustering methods on a variety of simu-

Figure 11: Factor matrices of BicMix and ISA. On the side of the factor matrix are the cell types and subtypes found by Z15, respectively. The rows of the factor matrices have been ordered to correspond to the Zeisel cell types.



(a) Factor matrix found by BicMix.



(b) Factor matrix found by ISA.

lated data. On the breast cancer microarray dataset of Van De Vijver et al. (2002); Van't Veer et al. (2002), SSLB finds biclusters corresponding to different subtypes of breast cancer. These biclusters also contained genes which were enriched for a variety of biological processes related to breast cancer. Finally, we applied SSLB to the mouse cortex and hippocampus single-cell RNA-sequencing dataset of Zeisel et al. (2015). SSLB recovered all the major cell classes found by Zeisel et al. (2015) as well as many of the cell subclasses. This performance was achieved despite the non-Gaussianity of the residual noise in the data, highlighting the potential robustness of SSLB to model misspecification. However, it would be interesting to explicitly extend SSLB to non-Gaussian residual noise models in future work. The SSLB software is available as an R/C++ package at `https://github.com/gemoran/SSLB`. Code to reproduce the results in this paper can also be found at `https://github.com/gemoran/SSLB-examples`.

References

Armagan, A., Clyde, M., and Dunson, D. B. (2011). "Generalized Beta Mixtures of Gaussians." In *Advances in Neural Information Processing Systems*, 523–531.

Bai, R., Moran, G. E., Antonelli, J., Chen, Y., and Boland, M. R. (2019). "Spike-and-Slab Group Lassos for Grouped Regression and Sparse Generalized Additive Models." *arXiv preprint arXiv:1903.01979*.

Ben-Dor, A., Chor, B., Karp, R., and Yakhini, Z. (2003). "Discovering local structure in gene expression data: the order-preserving submatrix problem." *Journal of Computational Biology*, 10(3-4): 373–384.

Bergmann, S., Ihmels, J., and Barkai, N. (2003). "Iterative signature algorithm for the analysis of large-scale gene expression data." *Phys Rev E Stat Nonlin Soft Matter Phys*, 67(3): 031902.

Bhattacharya, A. and Cui, Y. (2017). "A GPU-accelerated algorithm for biclustering analysis and detection of condition-dependent coexpression network modules." *Scientific Reports*, 7(1): 4162.

Bolstad, B. (2018). *preprocessCore: A collection of pre-processing functions*. R package version 1.44.0.
URL `https://github.com/bmbolstad/preprocessCore`

Bolstad, B., Irizarry, R., Åstrand, M., and Speed, T. (2003). "A comparison of normalization methods for high density oligonucleotide array data based on variance and bias." *Bioinformatics*, 19(2): 185–193.
URL `https://dx.doi.org/10.1093/bioinformatics/19.2.185`

Bozdağ, D., Kumar, A. S., and Catalyurek, U. V. (2010). "Comparative analysis of biclustering algorithms." In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, 265–274. ACM.

Bozdağ, D., Parvin, J. D., and Catalyurek, U. V. (2009). "A biclustering method to discover co-regulated genes using diverse gene expression datasets." In *Bioinformatics and Computational Biology*, 151–163. Springer.

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010). "The horseshoe estimator for sparse signals." *Biometrika*, 97(2): 465–480.

Castillo, I., Schmidt-Hieber, J., and Van der Vaart, A. (2015). "Bayesian linear regression with sparse priors." *The Annals of Statistics*, 43(5): 1986–2018.

Castillo, I. and van der Vaart, A. (2012). "Needles and straw in a haystack: Posterior concentration for possibly sparse sequences." *The Annals of Statistics*, 40(4): 2069–2101.

Cheng, Y. and Church, G. M. (2000). "Biclustering of expression data." In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, volume 8, 93–103. San Diego, U.S.A.

Csardi, G., Kutalik, Z., and Bergmann, S. (2010). "Modular analysis of gene expression data with R." *Bioinformatics*, 26: 1376–7.

De Castro, P. A., de França, F. O., Ferreira, H. M., and Von Zuben, F. J. (2007). "Evaluating the performance of a biclustering algorithm applied to collaborative filtering-a comparative analysis." In *Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on*, 65–70. IEEE.

Denitto, M., Bicego, M., Farinelli, A., and Figueiredo, M. A. (2017). "Spike and slab biclustering." *Pattern Recognition*, 72: 186–195.

Deshpande, S. K., Ročková, V., and George, E. I. (2019). "Simultaneous variable and covariance selection with the multivariate spike-and-slab lasso." *Journal of Computational and Graphical Statistics*, 1–11.

Doshi, F., Miller, K., Van Gael, J., and Teh, Y. W. (2009). "Variational inference for the Indian buffet process." In *Artificial Intelligence and Statistics*, 137–144.

Eren, K., Deveci, M., Küçüktunç, O., and Çatalyürek, Ü. V. (2012). "A comparative analysis of biclustering algorithms for gene expression data." *Briefings in bioinformatics*, 14(3): 279–292.

Fan, N., Boyko, N., and Pardalos, P. M. (2010). "Recent advances of data biclustering with application in computational neuroscience." In *Computational Neuroscience*, 85–112. Springer.

Feller, W. (1971). *An Introduction to Probability Theory and its Applications, Volume 2*. John Wiley & Sons.

Friedman, J. H. and Meulman, J. J. (2004). "Clustering objects on subsets of attributes (with discussion)." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(4): 815–849.

Frühwirth-Schnatter, S. and Lopes, H. F. (2010). "Parsimonious Bayesian factor analysis when the number of factors is unknown." Technical report, University of Chicago Booth School of Business.

Gao, C., McDowell, I. C., Zhao, S., Brown, C. D., and Engelhardt, B. E. (2016). "Context Specific and Differential Gene Co-expression Networks via Bayesian Biclustering." *PLoS Comput Biol*, 12(7): e1004791.

Ghahramani, Z. and Griffiths, T. L. (2006). "Infinite latent feature models and the Indian buffet process." In *Advances in Neural Information Processing Systems*, 475–482.

Griffiths, T. L. and Ghahramani, Z. (2011). "The Indian Buffet Process: An introduction and review." *Journal of Machine Learning Research*, 12(Apr): 1185–1224.

Gu, J. and Liu, J. S. (2008). "Bayesian biclustering of gene expression data." *BMC Genomics*, 9(1): S4.

Hartigan, J. A. (1972). "Direct Clustering of a Data Matrix." *Journal of the American Statistical Association*, 67(337): 123–129.
URL `http://www.jstor.org/stable/2284710`

Hastie, T., Tibshirani, R., Narasimhan, B., and Chu, G. (2018). *impute: impute: Imputation for microarray data*. R package version 1.56.0.

Hochreiter, S., Bodenhofer, U., Heusel, M., Mayr, A., Mitterecker, A., Kasim, A., Khamiakova, T., Van Sanden, S., Lin, D., Talloen, W., et al. (2010). "FABIA: factor analysis for bicluster acquisition." *Bioinformatics*, 26(12): 1520–1527.

Howlader, N., Altekruse, S. F., Li, C. I., Chen, V. W., Clarke, C. A., Ries, L. A., and Cronin, K. A. (2014). "US incidence of breast cancer subtypes defined by joint hormone receptor and HER2 status." *JNCI: Journal of the National Cancer Institute*, 106(5).

Huang, M., Wang, J., Torre, E., Dueck, H., Shaffer, S., Bonasio, R., Murray, J. I., Raj, A., Li, M., and Zhang, N. R. (2018). "SAVER: gene expression recovery for single-cell RNA sequencing." *Nature Methods*, 15(7): 539.

Islam, S., Zeisel, A., Joost, S., La Manno, G., Zajac, P., Kasper, M., Lönnerberg, P., and Linnarsson, S. (2014). "Quantitative single-cell RNA-seq with unique molecular identifiers." *Nature methods*, 11(2): 163–166.

Kaiser, H. F. (1958). "The varimax criterion for analytic rotation in factor analysis." *Psychometrika*, 23(3): 187–200.

Kaiser, S., Santamaria, R., Khamiakova, T., Sill, M., Theron, R., Quintales, L., Leisch, F., and De Troyer., E. (2020). *biclust: BiCluster Algorithms*. R package version 2.0.2. URL `https://CRAN.R-project.org/package=biclust`

Kellokumpu-Lehtinen, P., Talpaz, M., Harris, D., Van, Q., Kurzrock, R., and Estrov, Z. (1996). "Leukemia-inhibitory factor stimulates breast, kidney and prostate cancer cell proliferation by paracrine and autocrine pathways." *International journal of cancer*, 66(4): 515–519.

Kluger, Y., Basri, R., Chang, J. T., and Gerstein, M. (2003). "Spectral biclustering of microarray data: coclustering genes and conditions." *Genome research*, 13(4): 703–716.

Lazzeroni, L. and Owen, A. (2002). "Plaid models for gene expression data." *Statistica Sinica*, 61–86.

Madeira, S. C. and Oliveira, A. L. (2004). "Biclustering algorithms for biological data analysis: a survey." *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 1(1): 24–45.

Maxwell, J. C. (1860). "V. Illustrations of the dynamical theory of gases.—Part I. On the motions and collisions of perfectly elastic spheres." *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 19(124): 19–32.

Moran, G. E., Ročková, V., and George, E. I. (2018). "Variance prior forms for high-dimensional Bayesian variable selection." *Bayesian Analysis*.

Mucherino, A., Papajorgji, P., and Pardalos, P. M. (2009). *Data mining in agriculture*, volume 34. Springer Science & Business Media.

Munkres, J. (1957). "Algorithms for the assignment and transportation problems." *Journal of the Society for Industrial and Applied Mathematics*, 5(1): 32–38.

Onitilo, A. A., Engel, J. M., Greenlee, R. T., and Mukesh, B. N. (2009). "Breast cancer subtypes based on ER/PR and Her2 expression: comparison of clinicopathologic features and survival." *Clinical medicine & research*, 7(1-2): 4–13.

Padilha, V. A. and Campello, R. J. (2017). "A systematic comparative evaluation of biclustering techniques." *BMC Bioinformatics*, 18(1): 55.

Peeters, R. (2003). "The maximum edge biclique problem is NP-complete." *Discrete Applied Mathematics*, 131(3): 651–654.

Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., and Zitzler, E. (2006). "A systematic comparison and evaluation of biclustering methods for gene expression data." *Bioinformatics*, 22(9): 1122–1129.

Rangan, A. V., McGrouther, C. C., Kelsoe, J., Schork, N., Stahl, E., Zhu, Q., Krishnan, A., Yao, V., Troyanskaya, O., Bilaloglu, S., et al. (2018). "A loop-counting method for covariate-corrected low-rank biclustering of gene-expression and genome-wide association study data." *PLoS computational biology*, 14(5): e1006105.

Reya, T., Morrison, S. J., Clarke, M. F., and Weissman, I. L. (2001). "Stem cells, cancer, and cancer stem cells." *nature*, 414(6859): 105.

Ročková, V. and George, E. I. (2016). "Fast Bayesian factor analysis via automatic rotations to sparsity." *Journal of the American Statistical Association*, 111(516): 1608–1622.

— (2018). "The Spike-and-Slab Lasso." *Journal of the American Statistical Association*, 113(521): 431–444.

Ročková, V. et al. (2018). "Bayesian estimation of sparse signals with a continuous spike-and-slab prior." *The Annals of Statistics*, 46(1): 401–437.

Rohe, K. and Zeng, M. (2020). "Vintage Factor Analysis with Varimax Performs Statistical Inference." *arXiv preprint arXiv:2004.05387*.

Schroeder, M., Haibe-Kains, B., Culhane, A., Sotiriou, C., Bontempi, G., and Quackenbush, J. (2011). *breastCancerNKI: Genexpression dataset published by van't Veer et al. [2002] and van de Vijver et al. [2002] (NKI).*. R package version 1.12.0.
URL `http://compbio.dfci.harvard.edu/`

Shabalin, A. A., Weigman, V. J., Perou, C. M., Nobel, A. B., et al. (2009). "Finding large average submatrices in high dimensional data." *The Annals of Applied Statistics*, 3(3): 985–1012.

Teh, Y. W., Grür, D., and Ghahramani, Z. (2007). "Stick-breaking construction for the Indian buffet process." In *Artificial Intelligence and Statistics*, 556–563.

Van De Vijver, M. J., He, Y. D., Van't Veer, L. J., Dai, H., Hart, A. A., Voskuil, D. W., Schreiber, G. J., Peterse, J. L., Roberts, C., Marton, M. J., et al. (2002). "A gene-expression signature as a predictor of survival in breast cancer." *New England Journal of Medicine*, 347(25): 1999–2009.

Van't Veer, L. J., Dai, H., Van De Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., Van Der Kooy, K., Marton, M. J., Witteveen, A. T., et al. (2002). "Gene expression profiling predicts clinical outcome of breast cancer." *Nature*, 415(6871): 530.

Whitfield, M. L., George, L. K., Grant, G. D., and Perou, C. M. (2006). "Common markers of proliferation." *Nature Reviews Cancer*, 6(2): 99.

Yu, G. (2018). *enrichplot: Visualization of Functional Enrichment Result*. R package version 1.2.0.
URL `https://github.com/GuangchuangYu/enrichplot`

Yu, G., Wang, L.-G., Han, Y., and He, Q.-Y. (2012). "clusterProfiler: an R package for comparing biological themes among gene clusters." *OMICS: A Journal of Integrative Biology*, 16(5): 284–287.

Zeisel, A., Muñoz-Manchado, A. B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., et al. (2015). "Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq." *Science*, 347(6226): 1138–1142.

Zhan, T., Rindtorff, N., and Boutros, M. (2017). "Wnt signaling in cancer." *Oncogene*, 36(11): 1461.

Zhang, M. H., Man, H. T., Zhao, X. D., Dong, N., and Ma, S. L. (2014). "Estrogen receptor-positive breast cancer molecular signatures and therapeutic potentials." *Biomedical reports*, 2(1): 41–52.

Zhu, Y., Shen, X., and Ye, C. (2016). "Personalized prediction and sparsity pursuit in latent factor models." *Journal of the American Statistical Association*, 111(513): 241–252.

A SSLB Algorithm

In this section, we provide details for the EM algorithm we use to find the modes of the posterior. Before outlining the EM algorithm, we first marginalize over the binary indicator variables $\mathbf{\Gamma}$ (associated with the loadings $\mathbf{B}$) to yield the non-separable Spike-and-Slab Lasso prior (Ročková and George, 2018). For each column $\boldsymbol{\beta}_k$, the log of this prior (up to an additive constant) is:

$$\log \pi(\boldsymbol{\beta}_k) = \sum_{j=1}^{G} -\lambda_1|\beta_{jk}| + \log[p^*(0;\theta_{jk})/p^*(\beta_{jk};\theta_{jk})], \tag{A.1}$$

$$\text{where } p^*(\beta;\theta) = \theta\psi(\beta|\lambda_1)/[\theta\psi(\beta|\lambda_1) + (1-\theta)\psi(\beta|\lambda_0)] \tag{A.2}$$

and $\theta_{jk} = E[\theta_k|\boldsymbol{\beta}_{k\setminus j}]$ where $\boldsymbol{\beta}_{k\setminus j}$ denotes the vector $\boldsymbol{\beta}_k$ with the jth element removed. When G is large, $\boldsymbol{\beta}_{k\setminus j}$ is very similar to $\boldsymbol{\beta}_k$, so this expectation may be approximated by $E[\theta_k|\boldsymbol{\beta}_k]$.

We are now in a position to describe the EM algorithm. We find the expectation of $\mathbf{X}$ and factor indicators $\widetilde{\boldsymbol{\Gamma}}$ with respect to the complete log posterior and then maximize the resultant objective function:

$$Q(\boldsymbol{\Delta}) = \mathbb{E}_{\mathbf{X},\widetilde{\boldsymbol{\Gamma}}|\Delta^{(t)},\mathbf{Y}}\left[\log \pi(\boldsymbol{\Delta}, \mathbf{X}, \widetilde{\boldsymbol{\Gamma}}|\mathbf{Y})\right], \tag{A.3}$$

where we have used the notation $\boldsymbol{\Delta} = \{\mathbf{B}, \boldsymbol{\Sigma}, \mathbf{T}, \boldsymbol{\nu}\}$ to denote the parameters over which we will maximize. For convenience, we will use the notation $\mathbb{E}_{\mathbf{X},\widetilde{\boldsymbol{\Gamma}}|\boldsymbol{\Delta}^{(t)},\mathbf{Y}}(Z) = \langle Z\rangle$.

Now, due to the separability of the parameters in the posterior, we may write

$$Q(\boldsymbol{\Delta}) = Q_1(\mathbf{B}, \boldsymbol{\Sigma}) + Q_2(\mathbf{T}, \boldsymbol{\nu}) + Q_3(\boldsymbol{\nu}) + C, \tag{A.4}$$

where $Q_1(\mathbf{B}, \boldsymbol{\Sigma}) = \langle \pi(\mathbf{B}, \boldsymbol{\Sigma}, \boldsymbol{\Gamma}, \mathbf{X}|\mathbf{Y})\rangle$, $Q_2(\boldsymbol{\tau}, \boldsymbol{\nu}) = \langle \pi(\mathbf{X}, \mathbf{T}, \widetilde{\boldsymbol{\Gamma}}, \boldsymbol{\nu}|\mathbf{Y})\rangle$, $Q_3(\boldsymbol{\nu}) = \langle \pi(\boldsymbol{\nu}, \widetilde{\boldsymbol{\Gamma}}|\mathbf{Y})\rangle$ and $C \in \mathbb{R}$ is a constant.

The first term of the above objective function is:

$$\begin{aligned} Q_1(\mathbf{B}, \boldsymbol{\Sigma}) = C - \frac{1}{2}\sum_{i=1}^{N}\left\{(\mathbf{y}_i - \mathbf{B}\langle\mathbf{x}_i\rangle)^T\boldsymbol{\Sigma}^{-1}(\mathbf{y}_i - \mathbf{B}\langle\mathbf{x}_i\rangle) + \mathrm{tr}[\mathbf{B}'\boldsymbol{\Sigma}^{-1}\mathbf{B}(\langle\mathbf{x}_i\mathbf{x}_i'\rangle - \langle\mathbf{x}_i\rangle\langle\mathbf{x}_i\rangle')]\right\} \\ - \sum_{k=1}^{K^*}\log\pi(\boldsymbol{\beta}_k) - \frac{N+\eta+2}{2}\sum_{j=1}^{G}\log\sigma_j^2 - \sum_{j=1}^{G}\frac{\eta\xi}{2\sigma_j^2}, \end{aligned}$$

where $\pi(\boldsymbol{\beta}_k)$ is defined in (A.1). Next,

$$\begin{aligned} Q_2(\mathbf{T}) = -\frac{1}{2}\sum_{i=1}^{N}\left\{\langle\mathbf{x}_i\rangle^T\mathbf{D}_i\langle\mathbf{x}_i\rangle + \mathrm{tr}[\mathbf{D}_i(\langle\mathbf{x}_i\mathbf{x}_i'\rangle - \langle\mathbf{x}_i\rangle\langle\mathbf{x}_i\rangle')\right\} - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K^*}\log\tau_{ik} \\ - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K^*}\left[\langle\widetilde{\gamma}_{ik}\rangle\widetilde{\lambda}_1^2 + (1-\langle\widetilde{\gamma}_{ik}\rangle)\widetilde{\lambda}_0^2\right]\tau_{ik}. \end{aligned} \tag{A.5}$$

and finally,

$$\begin{aligned} Q_3(\boldsymbol{\nu}) = \sum_{k=1}^{K^*}\left[\langle\widetilde{\boldsymbol{\gamma}}_k\rangle\log\prod_{l=1}^{k}\nu_l + (N - \langle\widetilde{\boldsymbol{\gamma}}_k\rangle)\log\left(1 - \prod_{l=1}^{K}\nu_l\right)\right] \\ + \sum_{k=1}^{K^*}\left[(\widetilde{\alpha} + kd - 1)\log\nu_k - d\log(1-\nu_k)\right]. \end{aligned} \tag{A.6}$$

where $\langle\widetilde{\boldsymbol{\gamma}}_k\rangle = \sum_{i=1}^{N}\langle\widetilde{\gamma}_{ik}\rangle$.

A.0.1 E-Step

The conditional posterior distribution of $\mathbf{x}_i$ is given by:

$$\pi(\mathbf{x}_i|\mathbf{B}^{(t)}, \mathbf{\Sigma}^{(t)}, \mathbf{T}^{(t)}, \mathbf{y}_i) \sim N(\mathbf{V}^i \mathbf{B}'^{(t)} [\mathbf{\Sigma}^{(t)}]^{-1} \mathbf{y}_i, \mathbf{V}^i), \tag{A.7}$$

where $\mathbf{V}^i = [\mathbf{B}'^{(t)} [\mathbf{\Sigma}^{(t)}]^{-1} \mathbf{B}^{(t)} + \mathbf{D}_i^{(t)}]^{-1}$. Further, let $\mathbf{V} = \sum_{i=1}^{N} \mathbf{V}^i$.

We now determine the update for the indicators of the factors, $\widetilde{\mathbf{\Gamma}}$. Note that conditional on τ_{ik}, $\widetilde{\gamma}_{ik}$ is independent of x_{ik}. We have:

$$\begin{aligned}
\langle \widetilde{\gamma}_{ik} \rangle &= P\left(\widetilde{\gamma}_{ik} = 1 | \mathbf{T}, \widetilde{\boldsymbol{\theta}}\right) \\
&= \frac{\pi(\tau_{ik}|\widetilde{\gamma}_{ik} = 1)\pi(\widetilde{\gamma}_{ik} = 1|\widetilde{\theta}_k)}{\pi(\tau_{ik}|\widetilde{\gamma}_{ik} = 1)\pi(\widetilde{\gamma}_{ik} = 1|\widetilde{\theta}_k) + \pi(\tau_{ik}|\widetilde{\gamma}_{ik} = 0)\pi(\widetilde{\gamma}_{ik} = 0|\widetilde{\theta}_k)} \\
&= \frac{\widetilde{\theta}_k \widetilde{\lambda}_1^2 e^{-\widetilde{\lambda}_1^2 \tau_{ik}/2}}{\widetilde{\theta}_k \widetilde{\lambda}_1^2 e^{-\widetilde{\lambda}_1^2 \tau_{ik}/2} + (1 - \widetilde{\theta}_k) \widetilde{\lambda}_0^2 e^{-\widetilde{\lambda}_0^2 \tau_{ik}/2}}.
\end{aligned} \tag{A.8}$$

A.0.2 M-Step

Let $\mathbf{y}^1, \ldots, \mathbf{y}^G$ be the columns of $\mathbf{Y}$. Denote $\langle \mathbf{X} \rangle = [\langle \mathbf{x}_1 \rangle, \ldots, \langle \mathbf{x}_N \rangle]$ and let $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_G$ be the rows of $\mathbf{B}$. Then

$$Q_1(\mathbf{B}, \mathbf{\Sigma}) = \sum_{j=1}^{G} Q_j(\boldsymbol{\beta}_j, \sigma_j) \tag{A.9}$$

where

$$Q_j(\boldsymbol{\beta}_j, \sigma_j) = -\frac{1}{2\sigma_j^2} \|\mathbf{y}^j - \mathbf{X}\boldsymbol{\beta}_j\|^2 - \frac{1}{2\sigma_j^2} \boldsymbol{\beta}_j^T \mathbf{V} \boldsymbol{\beta}_j - \sum_{k=1}^{K^*} \log \pi(\boldsymbol{\beta}_k) - \frac{N + \eta + 2}{2} \log \sigma_j^2 - \frac{\eta \xi}{2\sigma_j^2} \tag{A.10}$$

To find a maximum of (A.10) with regard to $\boldsymbol{\beta}_j$, we use the refined thresholding scheme of Ročková and George (2018) with the extension to the unknown variance case given in Moran et al. (2018). Evaluation of $\log \pi(\boldsymbol{\beta}_k)$ requires the expectation of θ_k given the previous values of the loadings, $\boldsymbol{\beta}_k^{(t-1)}$; this yields the following update for θ_k (Ročková and George, 2018):

$$\theta_k^{(t)} = \frac{a + \|\boldsymbol{\beta}_k^{(t-1)}\|_0}{a + b + G}. \tag{A.11}$$

The update for σ_j^2 is:

$$\sigma_j^{2(t)} = \frac{\|\mathbf{y}^j - \mathbf{X}\boldsymbol{\beta}_j^{(t)}\|^2 + \boldsymbol{\beta}_j^{(t)T}\mathbf{V}\boldsymbol{\beta}_j^{(t)} + \eta\xi}{N + \eta + 2}. \tag{A.12}$$

The update for τ_{ik} is given by:

$$\tau_{ik}^{(t)} = \frac{-1 + \sqrt{1 + 4\widetilde{\lambda}_{ik}(\langle x_{ik}\rangle^2 + V_{kk}^i)}}{2\widetilde{\lambda}_{ik}} \tag{A.13}$$

where $\widetilde{\lambda}_{ik} = \langle\widetilde{\gamma}_{ik}\rangle\widetilde{\lambda}_1^2 + (1 - \langle\widetilde{\gamma}_{ik}\rangle)\widetilde{\lambda}_0^2$.

We now consider the update for the IBP stick-breaking parameters $\boldsymbol{\nu}$. This involves finding the $\boldsymbol{\nu}$ that maximize the objective in equation $Q_3(\boldsymbol{\nu})$. The difficulty in maximizing this objective is the non-linear term $\log\left(1 - \prod_{l=1}^k \nu_l\right)$. We find a lower bound for this term using a variational approximation inspired by Doshi et al. (2009).

This approximation begins with writing the non-linear term as a telescoping sum. Then, we introduce a parameter $\mathbf{q}_k = (q_{k1}, \ldots, q_{kk})$ where $\sum_{m=1}^k q_{km} = 1$, which allows the use of Jensen's inequality:

$$\begin{aligned}
\log\left(1 - \prod_{l=1}^k \nu_l\right) &= \log\left(\sum_{m=1}^k (1 - \nu_m)\prod_{l=1}^{m-1}\nu_l\right) \\
&= \log\left(\sum_{m=1}^k q_{km}\frac{(1 - \nu_m)\prod_{l=1}^{m-1}\nu_l}{q_{km}}\right) \\
&\geq \sum_{m=1}^k q_{km}\left[\log(1 - \nu_m) + \sum_{l=1}^{m-1}\log\nu_l\right] - \sum_{m=1}^k q_{km}\log q_{km}.
\end{aligned} \tag{A.14}$$

To make the bound (A.14) as tight as possible, we maximize over the parameter $\mathbf{q}_k$ to obtain updates $\widehat{\mathbf{q}}_k$:

$$\widehat{q}_{km}^{(t)} = \frac{\left(1 - \nu_m^{(t-1)}\right)\prod_{l=1}^{m-1}\nu_l^{(t-1)}}{1 - \prod_{l=1}^k \nu_l^{(t-1)}}. \tag{A.15}$$

The lower bound for the objective function for $\boldsymbol{\nu}$ at iteration t is now:

$$\begin{aligned}
Q_3(\boldsymbol{\nu}) \geq &\sum_{k=1}^{K^*}\left[\langle\widetilde{\gamma}_k\rangle\sum_{l=1}^k\log\nu_l + (N - \langle\widetilde{\gamma}_k\rangle)\left[\sum_{m=1}^k q_{km}^{(t)}\left(\log(1 - \nu_m) + \sum_{l=1}^{m-1}\log\nu_l\right)\right]\right] \\
&+ \sum_{k=1}^{K^*}[(\widetilde{\alpha} + kd - 1)\log\nu_k - d\log(1 - \nu_k)].
\end{aligned} \tag{A.16}$$

Maximizing the lower bound (A.16) over $\boldsymbol{\nu}$ then yields closed form updates:

$$\nu_k^{(t)} = \frac{r_k^{(t)}}{r_k^{(t)} + s_k^{(t)}} \tag{A.17}$$

where

$$r_k^{(t)} = \sum_{m=k}^{K^*} \langle \widetilde{\gamma}_k \rangle + \sum_{m=k+1}^{K^*} (N - \langle \widetilde{\gamma}_k \rangle) \left(\sum_{i=k+1}^{m} q_{mi}^{(t)} \right) + \widetilde{\alpha} + kd - 1 \tag{A.18}$$

$$s_k^{(t)} = \sum_{m=k}^{K^*} (N - \langle \widetilde{\gamma}_k \rangle) q_{mk}^{(t)} - d. \tag{A.19}$$

B Bicluster Quality Metrics

Here we provide the formulas for the (i) relevance; (ii) recovery; and (iii) consensus scores used to evaluate biclusters in the simulation studies. Each of these scores use the Jaccard index, a measure of similarity between two sets A and B, defined as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{B.1}$$

The Jaccard index naturally penalizes methods which find spurious bicluster elements. The relevance and recovery scores were proposed by Prelić et al. (2006) and are defined below. Denote bicluster C_k as the set non-zero entries of the vectorized matrix $\mathbf{x}^k \boldsymbol{\beta}^{kT}$. Let M_t be the set of true biclusters and let M_f be the set of biclusters found by a particular method. Then the relevance and recovery scores are given by:

$$\text{Relevance} = \frac{1}{|M_f|} \sum_{C_1 \in M_f} \max_{C_2 \in M_t} J(C_1, C_2),$$

$$\text{Recovery} = \frac{1}{|M_t|} \sum_{C_2 \in M_t} \max_{C_1 \in M_f} J(C_1, C_2).$$

The consensus score of Hochreiter et al. (2010) is computed as follows.

1. Compute the Jaccard similarity matrix, where the (i, j)th entry is the Jaccard similarity score (B.1) between the ith bicluster in M_t and the jth bicluster in M_f;
2. Find the optimal assignment (based on the highest Jaccard scores) of the true set of biclusters to the found set of biclusters using the Hungarian algorithm (Munkres, 1957);
3. Sum the similarity scores of the assigned biclusters and divide by $\max\{|M_t|, |M_f|\}$.

C Supplement for Simulations 1 and 2

C.1 Implementation details

The code source and implementation details of the methods we compared to are:

- **BicMix:** the code was obtained from `beehive.cs.princetone.edu/software` and implemented using the default parameters. Following Gao et al. (2016), we thresholded values less than 10^{-10}.
- **FABIA:** we implemented FABIA using the `fabia` R package (Hochreiter et al., 2010), using the default parameters and recommended post-processing thresholding step.
- **ISA:** we implemented ISA using the `isa2` R package (Csardi et al., 2010), using the default parameters.
- **Spectral:** we implemented Spectral using the `biclust` R package (Kaiser et al., 2020). For data matrix `Y`, we used the function call `biclust(exp(Y), method = BCSpectral())`. The data matrix was exponentiated as the default normalization for Spectral uses a log transform.
- **Plaid:** we implemented Plaid using the `biclust` R package. The function call was: `biclust(Y, method = BCPlaid(), max.layer = K)`, where `K` was the true number of biclusters (for simulation studies where `K` was known).

C.2 Additional figures

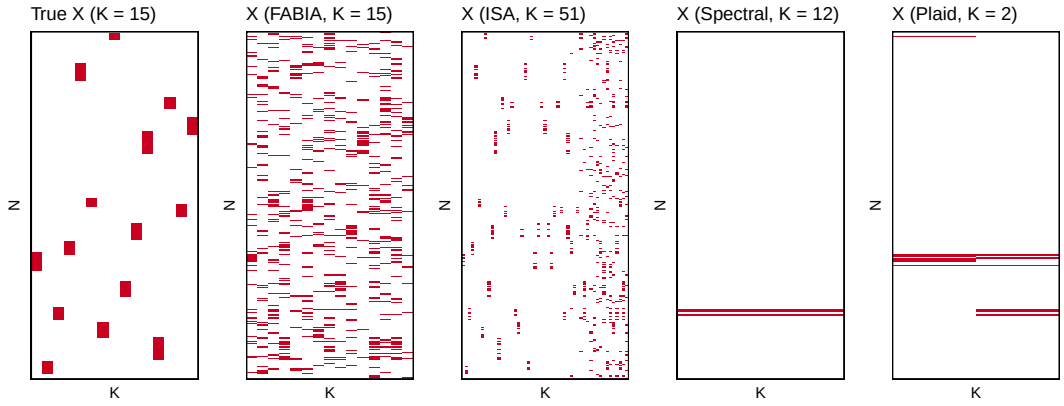
Here, we provide additional figures for Simulations 1 and 2. Figure 12 shows the biclusters found by each of FABIA, ISA, Spectral and Plaid for the dataset in Simulation 1 (Section 3.1). Figure 13 shows the biclusters found by each of FABIA, ISA, Spectral and Plaid for the dataset in Simulation 2 (Section 3.2).

Figure 14 shows the results of SSBiEM on Simulation 1 and 2 with an initial $K^* = 30$ instead of being set to the true number of biclusters. SSBiEM can still find the true bicluster signal; however, there is no thresholding of noisy biclusters. In practice, it may be hard to distinguish between true and noisy biclusters when the actual number of biclusters is unknown.

Finally, Figure 15 shows the proportion of variance in $\mathbf{Y}$ explained by each of the methods which provide a factorization of the data. This is given by:

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \|\mathbf{y}_i - \widehat{\mathbf{B}}\widehat{\mathbf{x}}_i\|^2}{\sum_{i=1}^{N} \|\mathbf{y}_i - \bar{\mathbf{y}}\|^2}. \tag{C.1}$$

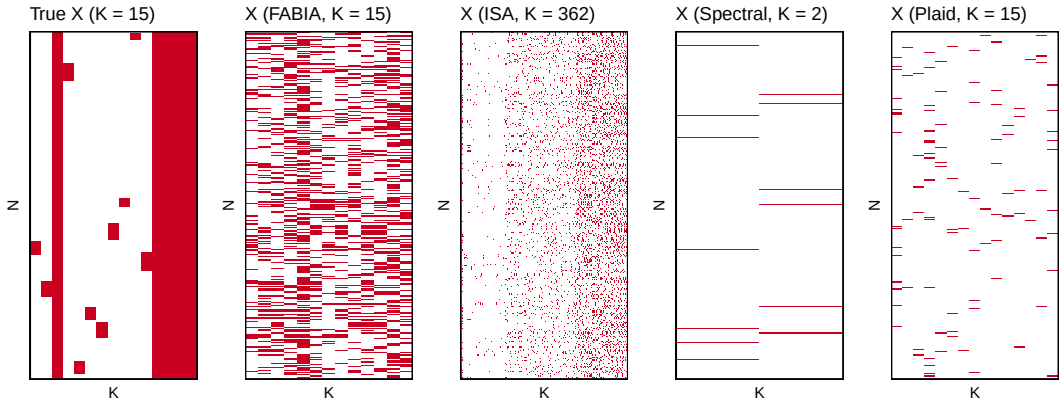
Figure 12: Simulation 1: Factor matrices, **X**, and loading matrices, **B**, found by different methods. Only the support of the matrix is displayed: a red value indicates a non-zero element.



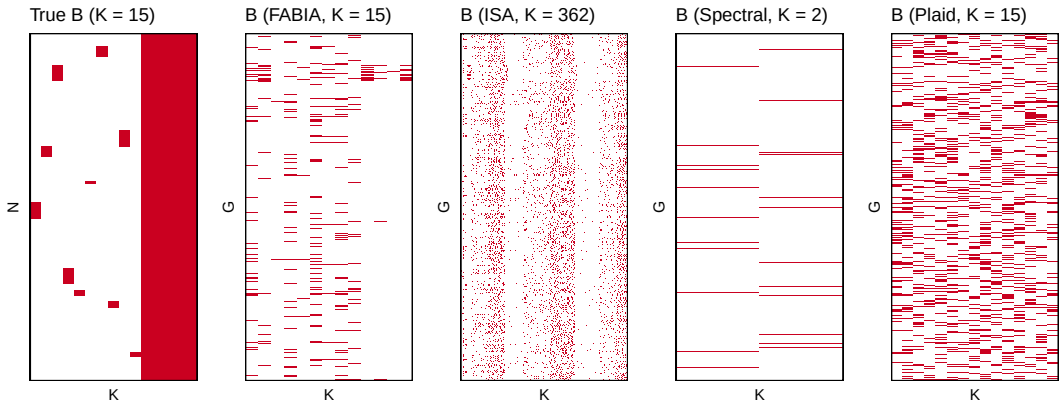
(a) Truth **(b)** FABIA **(c)** ISA **(d)** Spectral **(e)** Plaid

(f) Truth **(g)** FABIA **(h)** ISA **(i)** Spectral **(j)** Plaid

Figure 13: Simulation 2: Factor matrices, **X**, and loading matrices, **B**, found by different methods. Only the support of the matrix is displayed: a red value indicates a non-zero element.



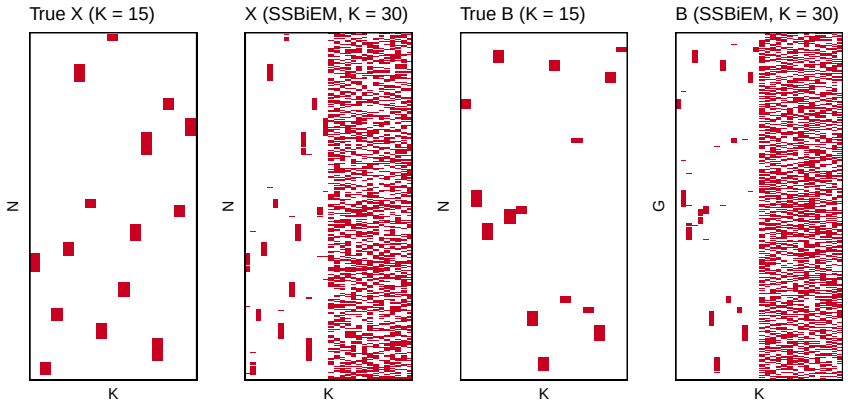
(a) Truth **(b)** FABIA **(c)** ISA **(d)** Spectral **(e)** Plaid



(f) Truth **(g)** FABIA **(h)** ISA **(i)** Spectral **(j)** Plaid

Figure 14: SSBiEM with initial $K^* = 30$

(a) Simulation 1



(b) Simulation 2

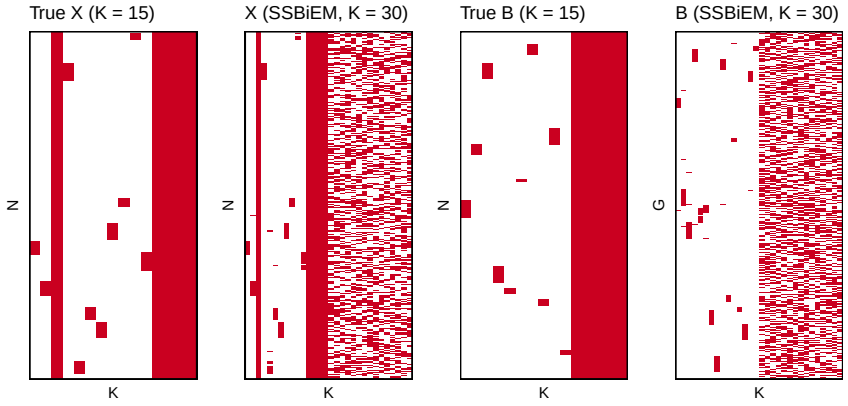
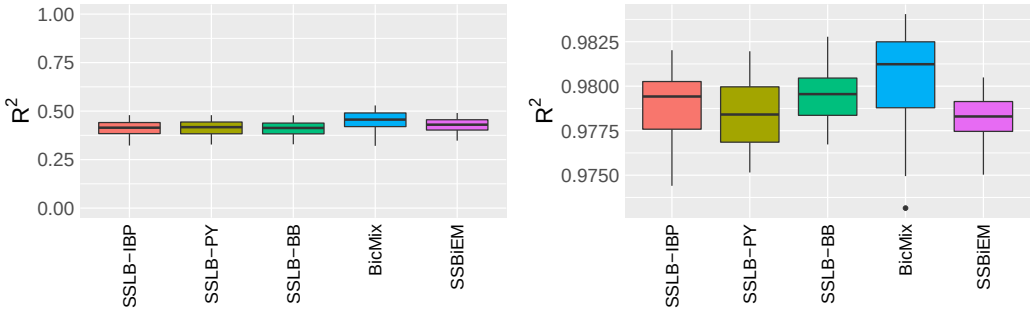


Figure 15: R^2 over 50 replications of the data.



(a) Simulation 1

(b) Simulation 2

In Simulation 1, SSLB and SSBiEM have a similar R^2, with BicMix attaining higher R^2 values. The higher R^2 values of BicMix are perhaps due to BicMix not thresholding smaller values of $\mathbf{X}$ and $\mathbf{B}$ to zero. Similarly to regression, retaining such small values leads to a better in-sample fit of $\mathbf{Y}$ and consequently higher R^2 values. An interesting direction for future work is to consider an adjusted R^2 for matrix factorization which accounts for the estimated degrees of freedom.

In Simulation 2, BicMix again has the highest R^2 values, followed by SSLB-IBP and SSLB-BB. SSLB-PY obtains similar R^2 values to SSBiEM, albeit with a slightly higher variance, which may be attributed to SSLB having to estimate the number of biclusters.

D Additional Simulation Studies

In this section, we conduct two additional simulation studies with a Poisson noise model, instead of a Gaussian noise model.

D.1 Simulation 3

We take $N = 300, G = 1000$ and $K = 15$. The simulated data was generated as follows. For biclusters $k = 1, \ldots, K$:

- For each column $\mathbf{x}^k$, we draw the number of samples in bicluster k uniformly from $\{5, \ldots, 20\}$. The indices of these elements were randomly selected and then assigned a value from a folded normal distribution with mean $\mu = 2$ and variance $\sigma^2 = 1$. The elements of $\mathbf{x}_k$ not in the bicluster were drawn from a folded normal with mean zero and variance $\sigma^2 = 0.2^2$.

- For each column $\boldsymbol{\beta}_k$, we draw the number of samples in bicluster k uniformly from $\{10, \ldots, 50\}$. The indices of these elements were randomly selected and then assigned a value from a folded normal distribution with mean $\mu = 1$ and variance $\sigma^2 = 1$. The elements of $\mathbf{x}_k$ not in the bicluster were drawn from a folded normal with mean zero and standard deviation $\sigma = 0.2$.

The matrix $\mathbf{Y}$ was then generated as:

$$\mathbf{Y} = \text{Poisson}(\mathbf{X}\mathbf{B}^T). \tag{D.1}$$

Figure 17 shows the consensus, relevance and recovery scores for each of the methods. All implementations of SSLB have higher consensus scores than the other methods. Interestingly, ISA has the next highest consensus scores in this setting. This improved performance is possibly attributed to ISA not requiring modeling assumptions; ISA finds submatrices in which all rows and columns are above a certain threshold. However, ISA still tends to overestimate the true number of biclusters, albeit by a smaller margin than in Simulation 1 (Table 3). SSLB also overestimates the true number of biclusters, while BicMix underestimates the true number of biclusters. For one of the 50 replicated datasets, the results from each of the methods are plotted in Figure 16.

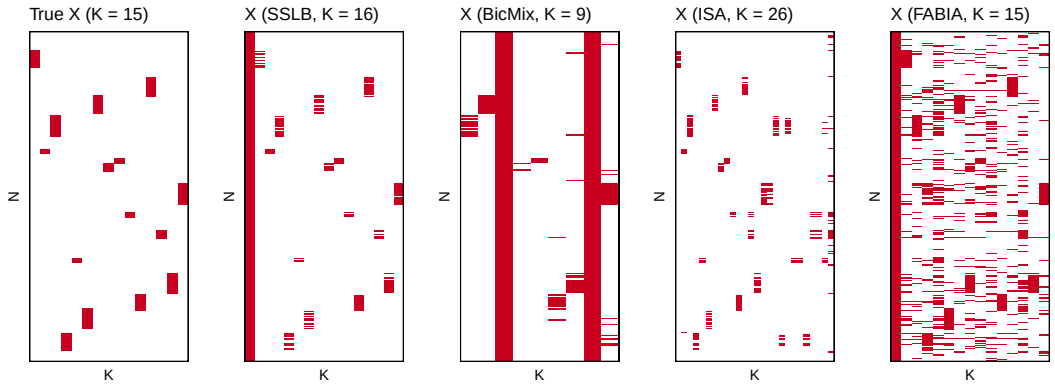
	$\widehat{K}$	
Method	Simulation 3	Simulation 4
Truth	*15*	*9*
SSLB-IBP	17.0 (0.21)	9.8 (0.19)
SSLB-PY	17.1 (0.20)	10.0 (0.18)
SSLB-BB	16.9 (0.21)	9.5 (0.15)
Bicmix	11.4 (0.23)	0.9 (0.13)
ISA	21.7 (0.52)	106.9 (3.21)
Spectral	30.6 (1.92)	1.0 (0.04)
Plaid	1.8 (0.17)	1.0 (0.00)

Table 3: Mean estimated number of biclusters, K, over 50 replications. Standard errors are shown in parentheses.

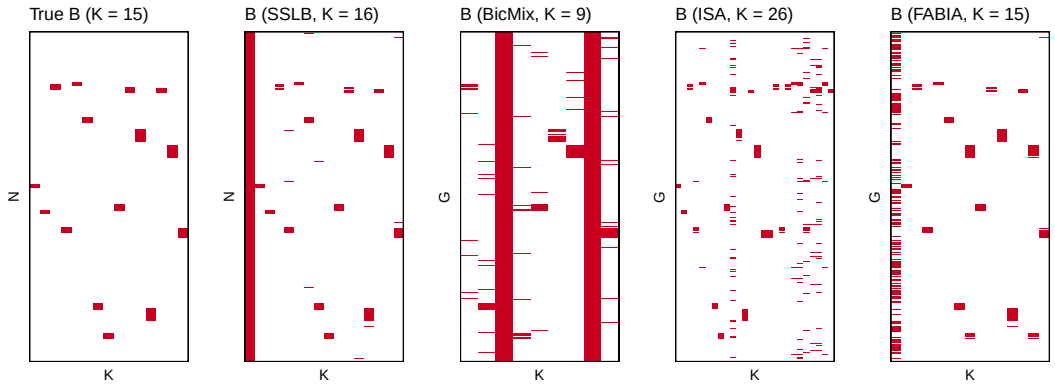
D.2 Simulation 4

For simulation 4, we again take $N = 300, G = 1000$ and $K = 15$. For both the factor and loading matrices, five columns are dense and ten columns are sparse. The sparse columns (corresponding to sparse biclusters) are generated as Simulation 1. The dense

Figure 16: Simulation 3: Factor matrices, $\mathbf{X}$, and loading matrices, $\mathbf{B}$, found by different methods. Only the support of the matrix is displayed: a red value indicates a non-zero element.

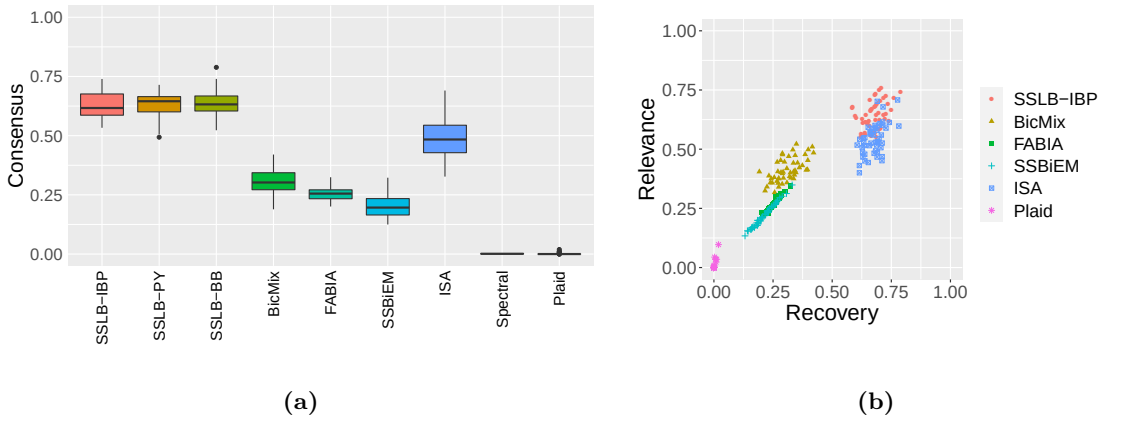


(a) Truth **(b)** SSLB-BB **(c)** BicMix **(d)** ISA **(e)** FABIA



(f) Truth **(g)** SSLB-BB **(h)** BicMix **(i)** ISA **(j)** FABIA

Figure 17: Simulation 3: (a) Boxplots of the consensus scores. (b) Relevance versus recovery scores.

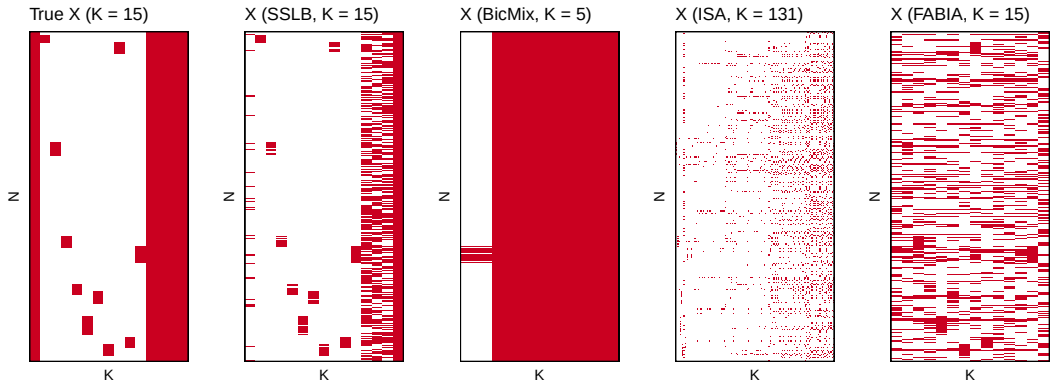


(a) **(b)**

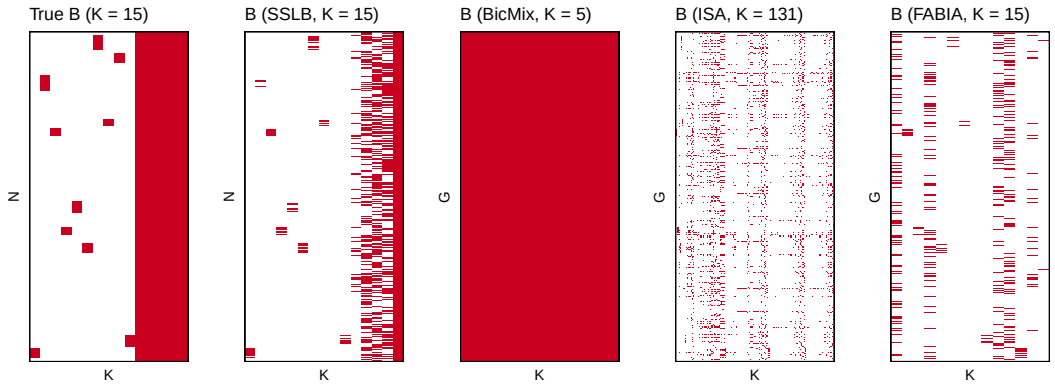
columns (corresponding to dense biclusters) are generated as independent folded normal distributions with $\mu = 0$ and $\sigma = 2$. We allow for one dense column in $\mathbf{X}$ to correspond to a sparse column in $\mathbf{B}$ and vice versa; this results in $K = 9$ biclusters which are sparse in both $\mathbf{X}$ and $\mathbf{B}$.

In this simulation setting, the consensus of all methods are much lower than in Simulation 3 (Figure 19). For SSLB, the reduced consensus scores are due to increased false negative rates, particularly in the $\mathbf{B}$ matrix (Figure 18). Encouragingly, however, SSLB does not seem to be finding spurious biclusters. This is unlike BicMix and FABIA, which find many more false positives. ISA also has lower consensus scores in this setting; we hypothesize ISA is better suited to detecting sparse biclusters, instead of a mix of both sparse and dense. ISA also overestimates the true number of biclusters again (Table 3). Meanwhile, SSLB slightly overestimates the number of biclusters in this setting.

Figure 18: Simulation 4: Factor matrices, $\mathbf{X}$, and loading matrices, $\mathbf{B}$, found by different methods. Only the support of the matrix is displayed: a red value indicates a non-zero element.

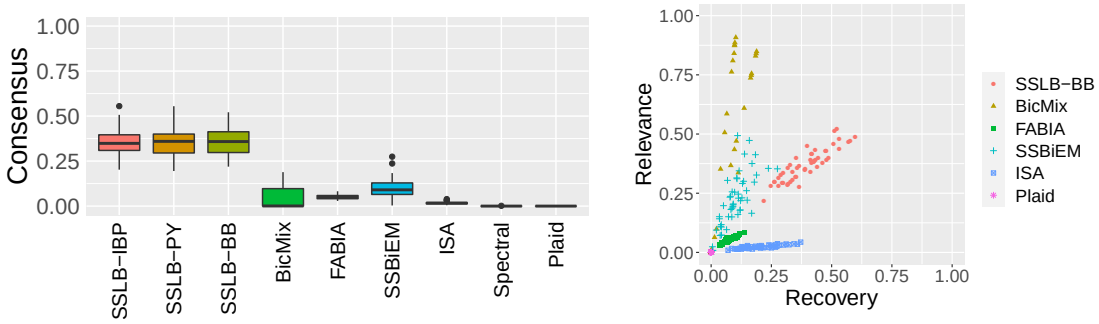


(a) Truth **(b)** SSLB-BB **(c)** BicMix **(d)** ISA **(e)** FABIA



(f) Truth **(g)** SSLB-BB **(h)** BicMix **(i)** ISA **(j)** FABIA

Figure 19: Simulation 4: (a) Boxplots of the consensus scores. (b) Relevance versus recovery scores.



(a) **(b)**

E Processing Breast Cancer Data

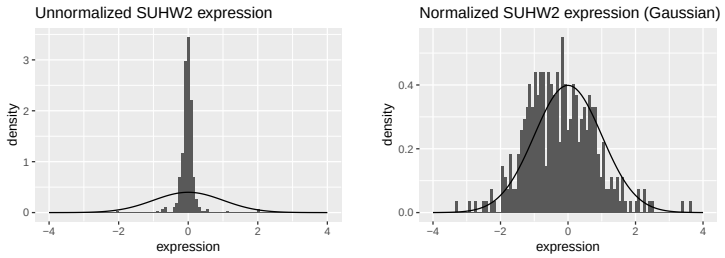
Here, we provide more details on the processing of the breast cancer dataset in Section 4. We first removed genes with more than 10% of values missing and imputed the remaining missing values with k nearest neighbors ($k = 10$), implemented using the R package `impute` (Hastie et al., 2018). We chose not to project the quantiles of the gene expression levels to the standard normal distribution, as done by Gao et al. (2016).

This is because the unnormalized gene expression values were mostly clustered around zero with heavy tails (Figure 20a). Although SSLB assumes that the errors are normally distributed, the gene loadings $\{\beta_{jk}\}_{j,k=1}^{G,K}$ are assumed to be drawn a priori from either a Laplacian spike concentrated around zero or a Laplacian slab. We assume that such a mixture model is flexible enough to model the gene expression levels exemplified in Figure 20a.

F Additional Figures for Breast Cancer Dataset

Here, we provide additional figures for the analysis of the breast cancer microarray dataset in Section 4. Figure 21a shows the full SSLB factor matrix, with Figure 21b showing the sparsity levels in the biclusters. The residuals from SSLB are symmetric around zero with moderately heavy tails (Figure 22a). The fitted $\widehat{\mathbf{Y}} = \widehat{\mathbf{X}}\widehat{\mathbf{B}}^T$ from SSLB generally approximates the observed $\mathbf{Y}$ well; however, SSLB shrinks a number of values of $\mathbf{Y}$ to zero (Figure 22b).

The enrichment maps (Figure 23) were created using the R package `enrichplot` (Yu, 2018)



(a) **(b)**

Figure 20: Histogram of (a) unnormalized expression values for gene $SUHW2$, (b) quantile normalized expression values for gene $SUHW2$ with standard normal distribution as reference. For both histograms, a standard normal density is overlaid.

and display the top 30 biological processes (with lowest FDR q-values satisfying threshold of 0.05) found in the gene ontology enrichment analysis as described in Section 4.3.

G Processing Zeisel Dataset

Here, we describe how we processed the data in Section 5. We followed the same pipeline as Z15 but provide the details here for completeness.

Many RNA-seq studies normalize the raw count data to the unit RPKM (Reads Per Kilobase of transcript per Million mapped reads), which accounts for longer genes having more transcripts mapped to them simply due to their length (and not meaningful biological variability). This was unnecessary for this dataset as only the 5' end of each RNA was sequenced and thus the read number was not proportional to gene length (Islam et al., 2014). Additionally, many single-cell RNA-seq studies account for differing cell sizes as larger cells have more RNA. However, this normalization was not done for this dataset as such information is informative in clustering different cell types.

The scRNA-seq data is provided by Z15 at `http://linnarssonlab.org/cortex` and consists of molecule counts for 19,972 genes in 3005 individual cells.

Following Z15, we:

1. Removed all genes that have less than 25 molecules in total over all cells
2. Calculated correlation matrix over the genes and define a threshold as 90th percentile of this matrix ($\rho = 0.2091$). Removed all genes which have less than 5 other genes which correlate more than this threshold.

The next step of data processing was to identify the noisiest genes. Assuming that most of the variability of the genes across the cells can be attributed to the underlying biological processes, these genes are the ones which are most informative for clustering of cells. The strategy of Z15 was to search for genes whose noise - measured by coefficient of variation (CV, standard deviation divided by mean) - was high compared to a Poisson distribution with inflated CV. The rationale for this was outlined in Islam et al. (2014) which used the same single-cell RNA-seq protocol as Z15 but for mouse embryonic stem cells. First, Islam et al. (2014) noted that the technical noise distribution of ERCC (External RNA Controls Consortium) spike-in molecules (which have no biological variability) followed that of a Poisson, but its CV was inflated by constant factor. The CVs of endogenous genes were inflated above those of the ERCCs, suggesting that this variation is driven by biological factors rather than the variation induced by loss of transcripts in cDNA synthesis.

Z15 implemented the same procedure to identify genes with the greatest biological variability. We followed this procedure: for the genes remaining after the aforementioned data cleaning steps, the mean and CV was calculated. The noise model

$$\log_2(CV) = \log_2(mean^{\alpha} + k)$$

was fit using the software `ceftools`[8]. The best fit was found to be $\alpha = -0.55$ and $k = 0.64$. Next all genes were ranked by their distance from the fit line and the top 5000 genes with the largest distance were selected as informative for further clustering.

Finally, we normalized the gene counts using quantile normalization (using the R package `preprocessCore` (Bolstad, 2018)). Note we used the commonly used "average distribution" as the reference distribution to which to project the quantiles of the raw gene expression levels. The average distribution is obtained by taking the average of each quantile across the samples (Bolstad et al., 2003).

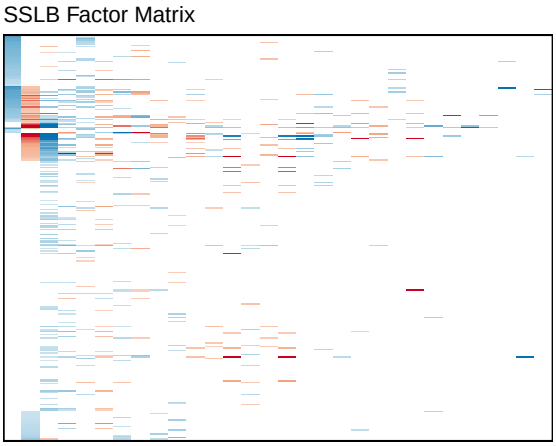
H Supplementary Figures for Zeisel Dataset

Here, we provide supplementary figures for the analysis of the mouse single-cell RNA sequencing dataset in Section 5. Figure 24 displays full results from SSLB and FABIA. Figure 26 shows residual plots from SSLB results. SSLB residuals are very heavy tailed, but centered around zero (Figures 26a and 26b). Fitted SSLB values estimate the observed data for the most part; however, there are a number of zeroes mis-estimated as non-zero values, and vice versa (Figures 26c and 26d).

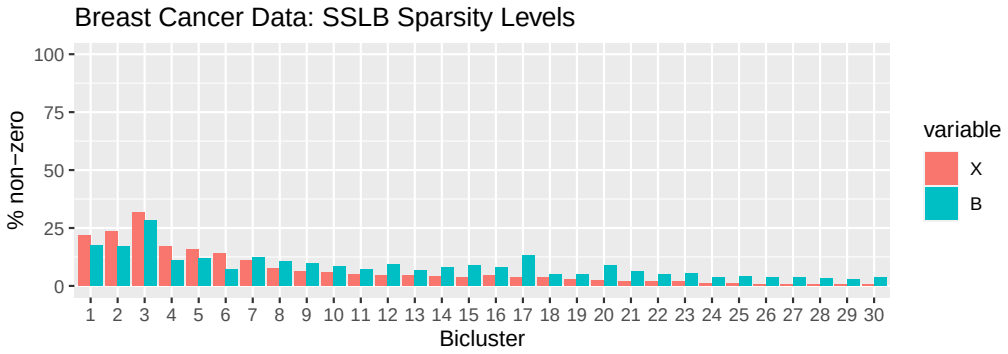
Enrichment maps (Figures 27 and 28) were created using the R package `enrichplot` (Yu, 2018) and display the top 30 biological processes (with lowest FDR q-values satisfying threshold of 0.05) found in the gene ontology enrichment analysis as described in Section 5.1.

[8] `https://github.com/linnarsson-lab/ceftools`

Figure 21

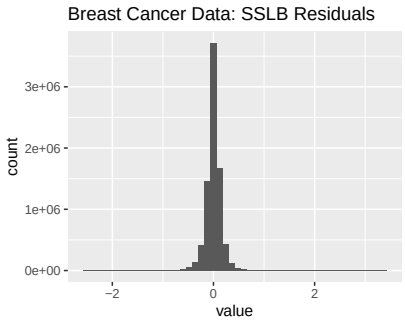


(a) SSLB factor matrix where each row corresponds to a patient and each column corresponds to a bicluster. A patient belongs to a bicluster if they have a non-zero value in that column. Rows are ordered by clinical ER status; within ER status, rows are ordered by factor values in biclusters 1 and 2. All 30 biclusters found by SSLB are shown.

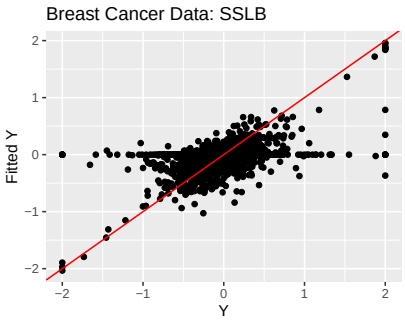


(b) Percentage of non-zero elements in each bicluster found by SSLB.

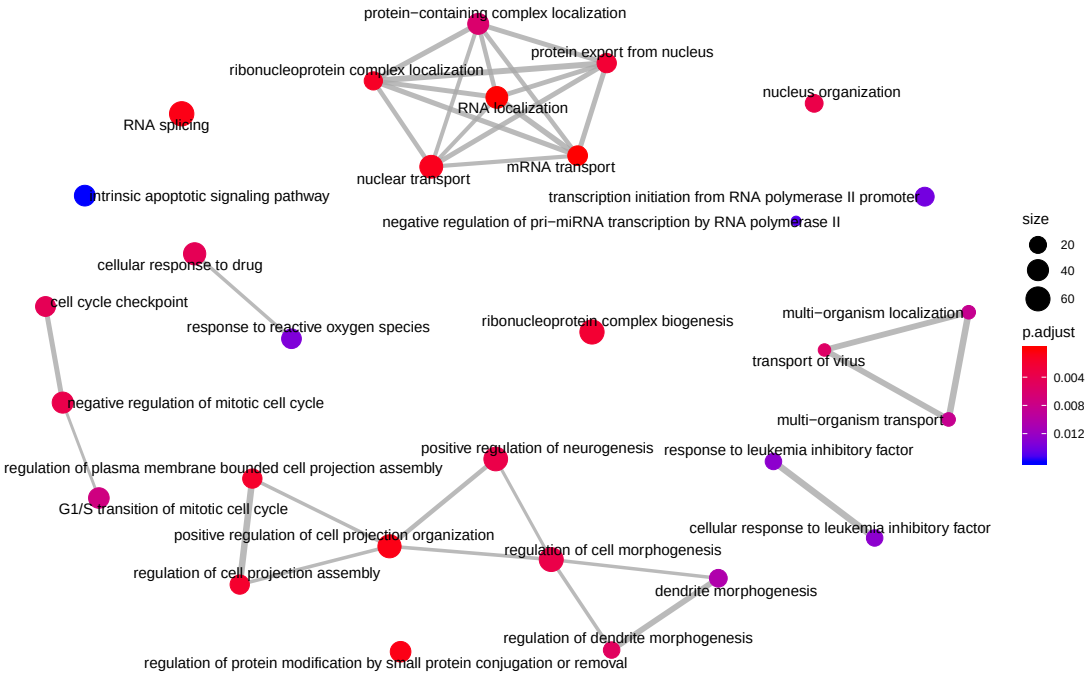
Figure 22



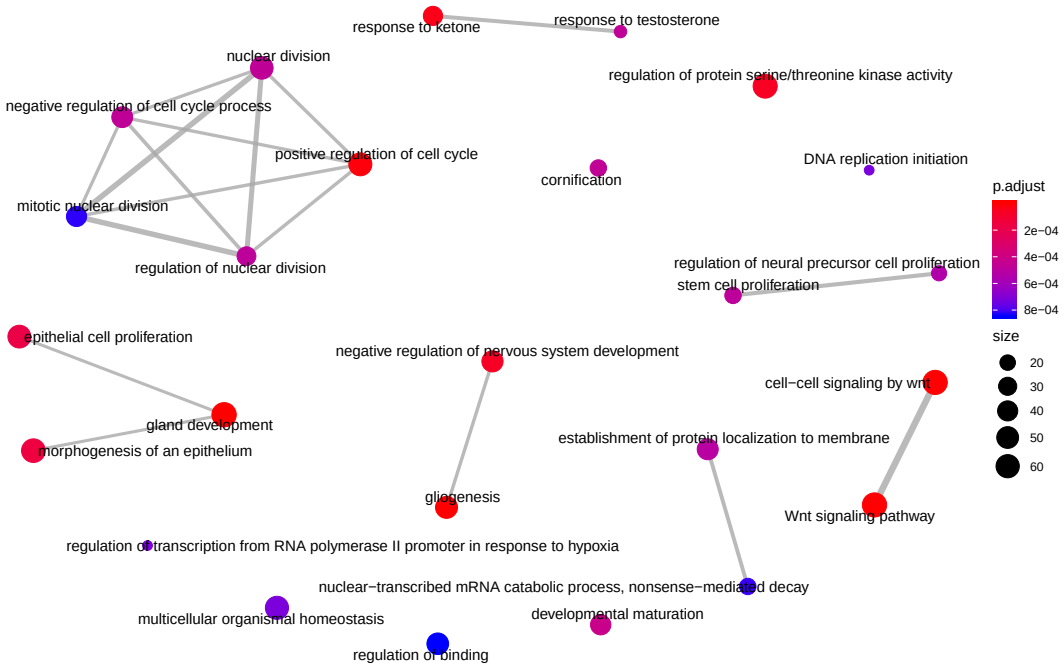
(a) Histogram of SSLB residuals.



(b) Fitted SSLB $\widehat{\mathbf{Y}} = \widehat{\mathbf{X}}\widehat{\mathbf{B}}^T$ vs. observed $\mathbf{Y}$ for a randomly sampled subset of 10,000 points. Red line is $y = x$.



(a) Enrichment map for genes up-regulated in ER-negative patients.



(b) Enrichment map for genes up-regulated in HER2+ patients.

Figure 23: Breast cancer data: enrichment maps for SSLB genes (a) up-regulated in ER-negative patients, and (b) up-regulated in HER2+ patients. Nodes represent biological processes; size of node reflects number of genes in process which were found by the method. Edges connect genes that are active in different biological processes.

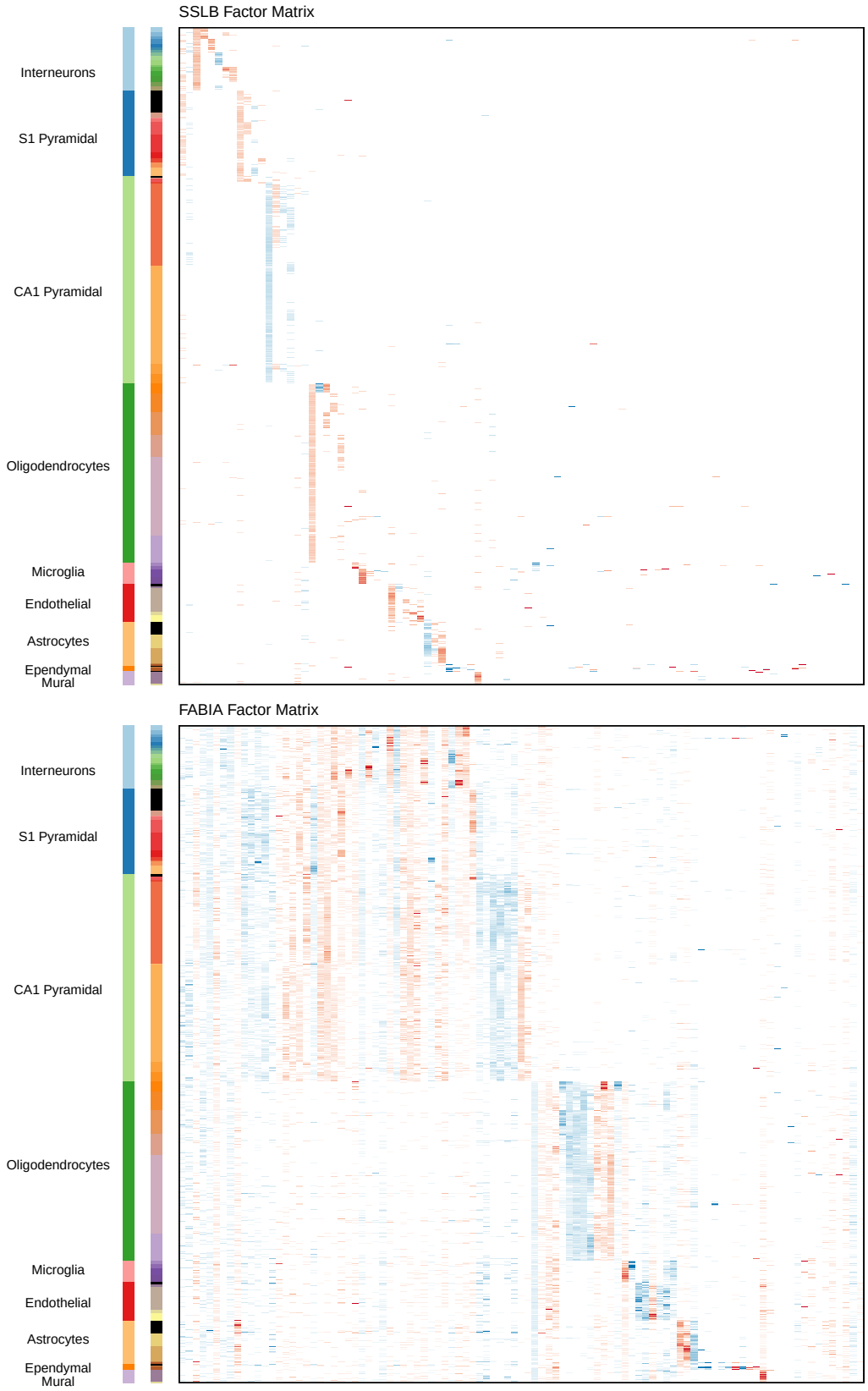


Figure 24: Zeisel dataset: Factor matrix found by SSLB (top) and FABIA (bottom). On the side of the factor matrix are the cell types and subtypes found by Z15, respectively. The rows of the factor matrices have been ordered to correspond to the Zeisel cell types. Factor values have been capped for improved visualization.

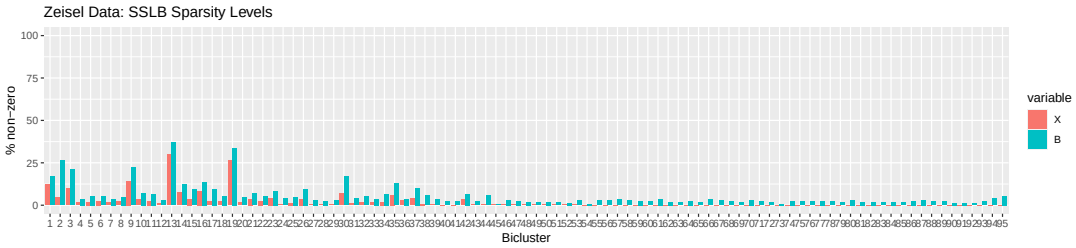
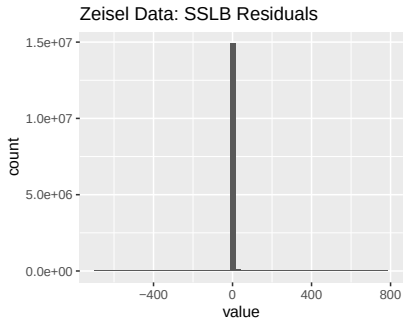
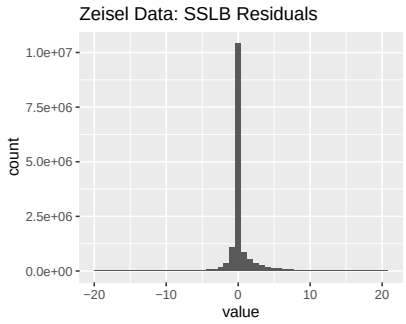


Figure 25: Percentage of non-zero elements in each bicluster found by SSLB.

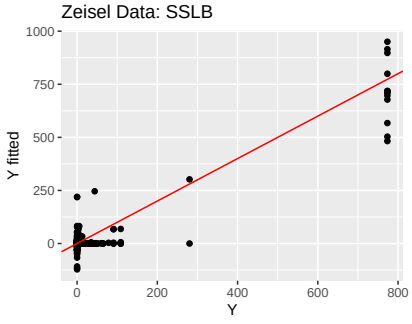
Figure 26



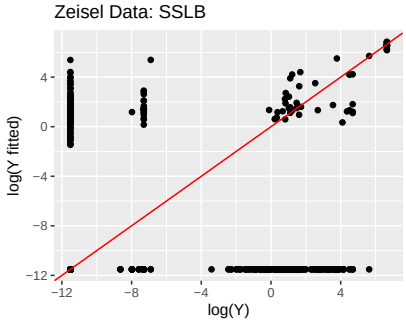
(a) Histogram of SSLB residuals.



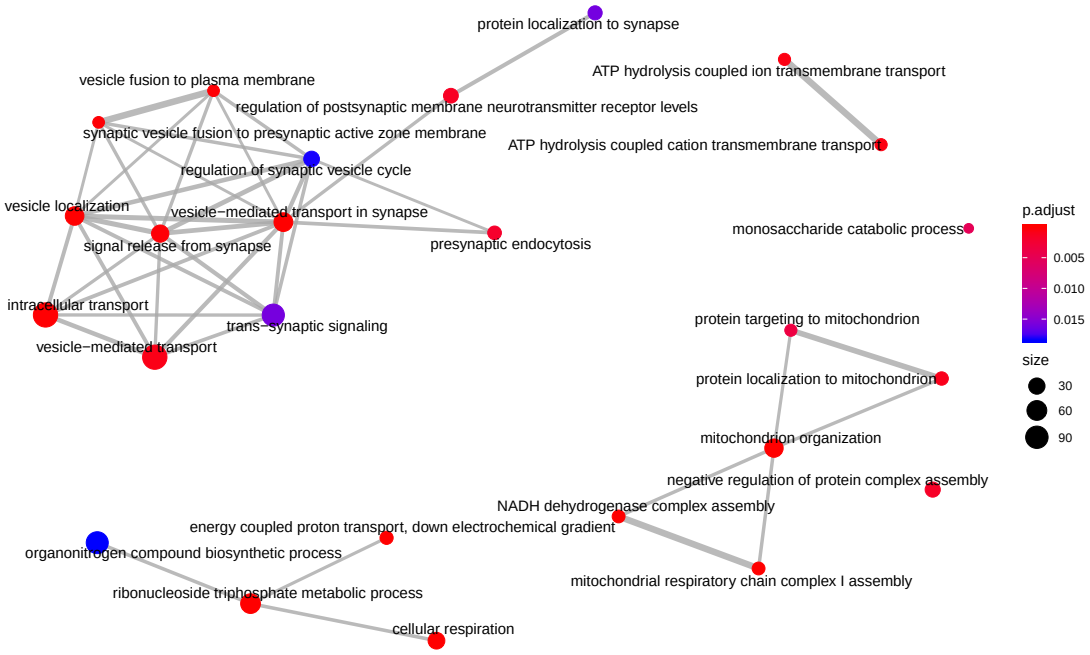
(b) Histogram of SSLB residuals with absolute values greater than 20 filtered out (removed 0.4% of the data).



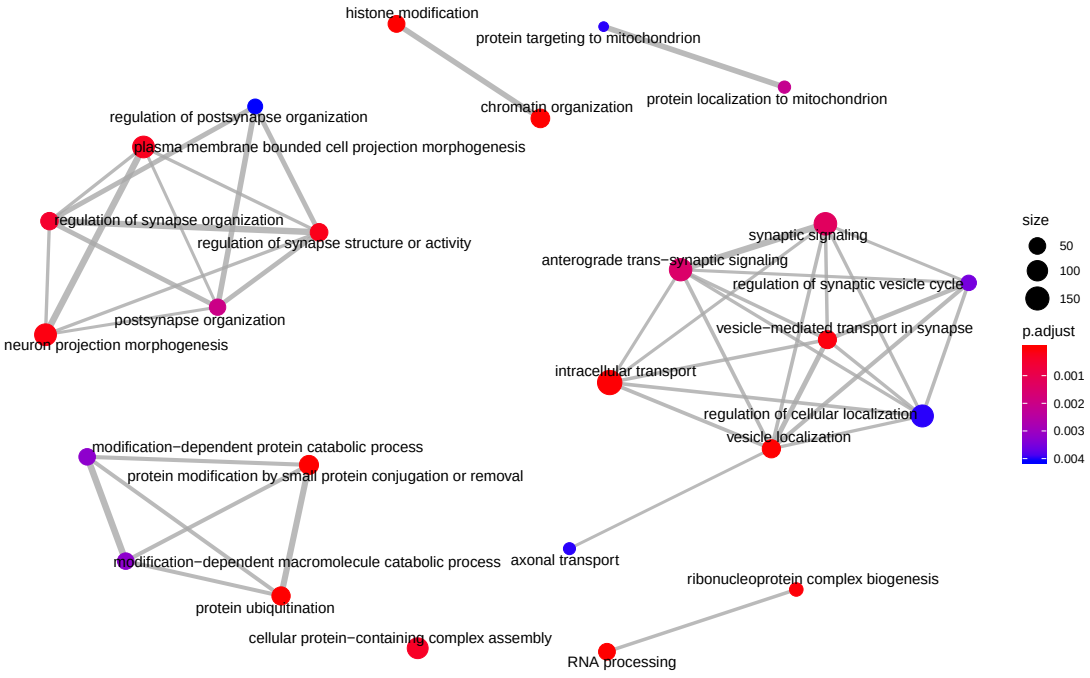
(c) Fitted SSLB $\widehat{\mathbf{Y}} = \widehat{\mathbf{X}}\widehat{\mathbf{B}}^T$ vs. observed $\mathbf{Y}$ for a randomly sampled subset of 10,000 points. Red line is $y = x$.



(d) Log of fitted SSLB matrix vs. log of observed data for a randomly sampled subset of 10,000 points. SSLB values less than zero were set to zero. Offset of 10^{-6} was added before taking the log.



(a) SSLB Bicluster 1



(b) SSLB Bicluster 2

Figure 27: Zeisel dataset: enrichment maps for SSLB genes in (a) bicluster 1 and (b) bicluster 2. Each bicluster contains a mixture of interneurons, S1 pyramidal neurons and CA1 pyramidal neurons. Nodes represent biological processes; size of node reflects number of genes in process which were found by the method. Edges connect genes that are active in different biological processes.

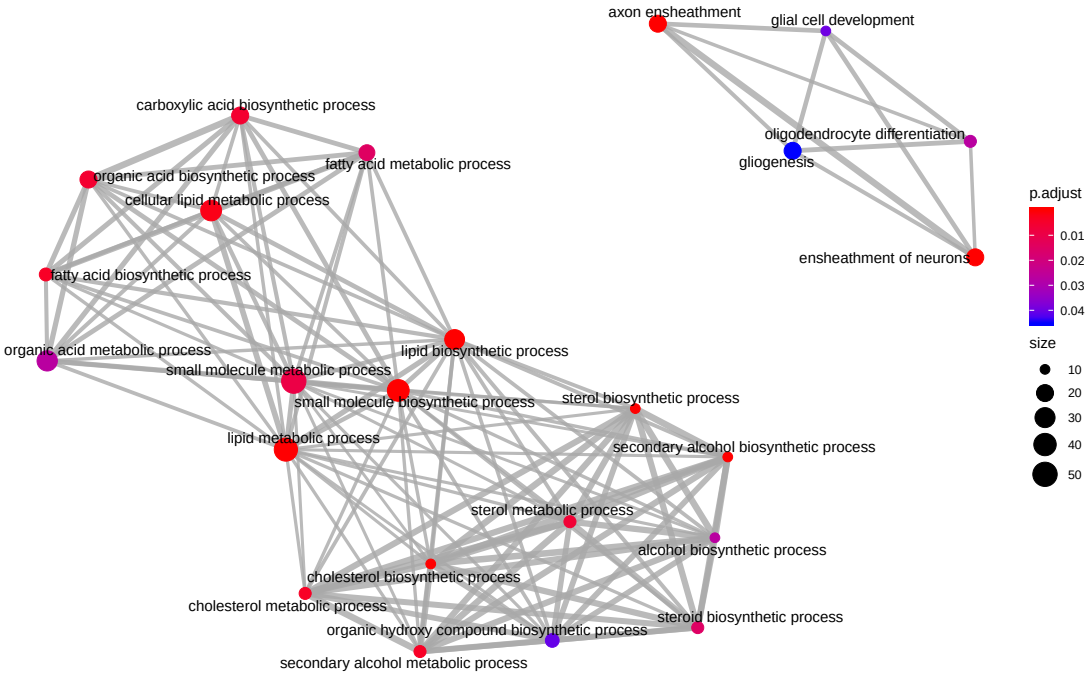


Figure 28: Zeisel dataset: enrichment map for genes in SSLB bicluster 44. Bicluster 44 contains 17 oligodendrocyte cells. Nodes represent biological processes; size of node reflects number of genes in process which were found by the method. Edges connect genes that are active in different biological processes.