Submitted to the Annals of Applied Statistics

SUPPLEMENT TO "SPIKE-AND-SLAB LASSO BICLUSTERING"

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

Columbia University[*], University of Chicago[†] and University of Pennsylvania[‡]

1. 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:

$$\log \pi(\boldsymbol{\beta}_k) = \sum_{j=1}^{G} -\lambda_1|\beta_{jk}| + \log[p^*(0;\theta_{jk})/p^*(\beta_{jk};\theta_{jk})], \tag{1.1}$$

$$\text{where } p^*(\beta;\theta) = \theta\psi(\beta|\lambda_1)/[\theta\psi(\beta|\lambda_1) + (1-\theta)\psi(\beta|\lambda_0)] \tag{1.2}$$

and $\theta_{jk} = E[\theta_k|\boldsymbol{\beta}_{k\backslash j}]$ where $\boldsymbol{\beta}_{k\backslash j}$ denotes the vector $\boldsymbol{\beta}_k$ with the jth element removed. When G is large, $\boldsymbol{\beta}_{k\backslash j}$ 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 $\widetilde{\mathbf{\Gamma}}$ with respect to the complete log posterior and then maximize the resultant objective function:

$$Q(\mathbf{\Delta}) = \mathbb{E}_{\mathbf{X},\widetilde{\mathbf{\Gamma}}|\Delta^{(t)},\mathbf{Y}}\left[\log \pi(\mathbf{\Delta}, \mathbf{X}, \widetilde{\mathbf{\Gamma}}|\mathbf{Y})\right], \tag{1.3}$$

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

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

$$Q(\mathbf{\Delta}) = Q_1(\mathbf{B}, \mathbf{\Sigma}) + Q_2(\mathbf{T}, \boldsymbol{\nu}) + Q_3(\boldsymbol{\nu}) + C, \tag{1.4}$$

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

The first term of the above objective function is:

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

where $\pi(\boldsymbol{\beta}_k)$ is defined in (1.1). Next,

$$Q_2(\mathbf{T}) = -\frac{1}{2}\sum_{i=1}^{N}\left\{\langle \mathbf{x}_i\rangle^T\mathbf{D}_i\langle \mathbf{x}_i\rangle + \mathrm{tr}[\mathbf{D}_i(\langle \mathbf{x}_i\mathbf{x}_i'\rangle - \langle \mathbf{x}_i\rangle\langle \mathbf{x}_i\rangle')\right\} - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K^*}\log\tau_{ik}$$
$$- \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K^*}\left[\langle\widetilde{\gamma}_{ik}\rangle\widetilde{\lambda}_1^2 + (1 - \langle\widetilde{\gamma}_{ik}\rangle)\widetilde{\lambda}_0^2\right]\tau_{ik}. \tag{1.5}$$

and finally,

$$Q_3(\boldsymbol{\nu}) = \sum_{k=1}^{K^*}\left[\langle\widetilde{\boldsymbol{\gamma}}_k\rangle\log\prod_{l=1}^{k}\nu_l + (N - \langle\widetilde{\boldsymbol{\gamma}}_k\rangle)\log\left(1 - \prod_{l=1}^{K}\nu_l\right)\right]$$
$$+ \sum_{k=1}^{K^*}\left[(\widetilde{\alpha} + kd - 1)\log\nu_k - d\log(1-\nu_k)\right]. \tag{1.6}$$

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

1.0.1. *E-Step.* The conditional posterior distribution of $\mathbf{x}_i$ is given by:

$$\pi(\mathbf{x}_i|\mathbf{B}^{(t)}, \mathbf{\Sigma}^{(t)}, \mathbf{T}^{(t)}, \mathbf{y}_i) \sim N(\mathbf{V}^i\mathbf{B}'^{(t)}[\mathbf{\Sigma}^{(t)}]^{-1}\mathbf{y}_i, \mathbf{V}^i), \tag{1.7}$$

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:

$$\begin{aligned}\langle\widetilde{\gamma}_{ik}\rangle &= P\left(\widetilde{\gamma}_{ik}=1|\mathbf{T},\widetilde{\boldsymbol{\theta}}\right)\\ &= \frac{\pi(\tau_{ik}|\widetilde{\gamma}_{ik}=1)\pi(\widetilde{\gamma}_{ik}=1|\widetilde{\theta}_k)}{\pi(\tau_{ik}|\widetilde{\gamma}_{ik}=1)\pi(\widetilde{\gamma}_{ik}=1|\widetilde{\theta}_k)+\pi(\tau_{ik}|\widetilde{\gamma}_{ik}=0)\pi(\widetilde{\gamma}_{ik}=0|\widetilde{\theta}_k)}\\ &= \frac{\widetilde{\theta}_k\widetilde{\lambda}_1^2 e^{-\widetilde{\lambda}_1^2\tau_{ik}/2}}{\widetilde{\theta}_k\widetilde{\lambda}_1^2 e^{-\widetilde{\lambda}_1^2\tau_{ik}/2}+(1-\widetilde{\theta}_k)\widetilde{\lambda}_0^2 e^{-\widetilde{\lambda}_0^2\tau_{ik}/2}}. \qquad (1.8)\end{aligned}$$

1.0.2. *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

$$Q_1(\mathbf{B},\boldsymbol{\Sigma})=\sum_{j=1}^{G}Q_j(\boldsymbol{\beta}_j,\sigma_j) \qquad (1.9)$$

where

(1.10)

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

To find a maximum of (1.10) 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, Ročková and George (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):

$$\theta_k^{(t)}=\frac{a+\|\boldsymbol{\beta}_k^{(t-1)}\|_0}{a+b+G}. \qquad (1.11)$$

The update for σ_j^2 is:

$$\sigma_j^{2(t)}=\frac{\|\mathbf{y}^j-\mathbf{X}\boldsymbol{\beta}_j^{(t)}\|^2+\boldsymbol{\beta}_j^{(t)T}\mathbf{V}\boldsymbol{\beta}_j^{(t)}+\eta\xi}{N+\eta+2}. \qquad (1.12)$$

The update for τ_{ik} is given by:

$$\tau_{ik}^{(t)}=\frac{-1+\sqrt{1+4\widetilde{\lambda}_{ik}(\langle x_{ik}\rangle^2+V_{kk}^i)}}{2\widetilde{\lambda}_{ik}} \qquad (1.13)$$

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:

$$\begin{aligned} \log\left(1 - \prod_{l=1}^{k} \nu_l\right) &= \log\left(\sum_{m=1}^{k} (1 - \nu_m) \prod_{l=1}^{m-1} \nu_l\right) \\ &= \log\left(\sum_{m=1}^{k} q_{km} \frac{(1 - \nu_m) \prod_{l=1}^{m-1} \nu_l}{q_{km}}\right) \\ &\geq \sum_{m=1}^{k} q_{km} \left[\log(1 - \nu_m) + \sum_{l=1}^{m-1} \log \nu_l\right] - \sum_{m=1}^{k} q_{km} \log q_{km}. \end{aligned} \tag{1.14}$$

To make the bound (1.14) as tight as possible, we maximize over the parameter $\mathbf{q}_k$ to obtain updates $\widehat{\mathbf{q}}_k$:

$$\widehat{q}_{km}^{(t)} = \frac{\left(1 - \nu_m^{(t-1)}\right) \prod_{l=1}^{m-1} \nu_l^{(t-1)}}{1 - \prod_{l=1}^{k} \nu_l^{(t-1)}}. \tag{1.15}$$

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

$$\begin{aligned} Q_3(\boldsymbol{\nu}) \geq \sum_{k=1}^{K^*} \left[\langle \widetilde{\gamma}_k \rangle \sum_{l=1}^{k} \log \nu_l + (N - \langle \widetilde{\gamma}_k \rangle) \left[\sum_{m=1}^{k} q_{km}^{(t)} \left(\log(1 - \nu_m) + \sum_{l=1}^{m-1} \log \nu_l\right)\right]\right] \\ + \sum_{k=1}^{K^*} [(\widetilde{\alpha} + kd - 1) \log \nu_k - d \log(1 - \nu_k)]. \end{aligned} \tag{1.16}$$

Maximizing the lower bound (1.16) over $\boldsymbol{\nu}$ then yields closed form updates:

$$\nu_k^{(t)} = \frac{r_k^{(t)}}{r_k^{(t)} + s_k^{(t)}} \tag{1.17}$$

where

$$r_k^{(t)} = \sum_{m=k}^{K^*} \langle \widetilde{\boldsymbol{\gamma}}_k \rangle + \sum_{m=k+1}^{K^*} (N - \langle \widetilde{\boldsymbol{\gamma}}_k \rangle) \left(\sum_{i=k+1}^{m} q_{mi}^{(t)} \right) + \widetilde{\alpha} + kd - 1 \tag{1.18}$$

$$s_k^{(t)} = \sum_{m=k}^{K^*} (N - \langle \widetilde{\boldsymbol{\gamma}}_k \rangle) q_{mk}^{(t)} - d. \tag{1.19}$$

2. 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:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{2.1}$$

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 (2.1) 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 $\max\{|M_t|, |M_f|\}$.

3. Supplement for Simulations 1 and 2.

3.1. *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, Kutalik and Bergmann, 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).

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

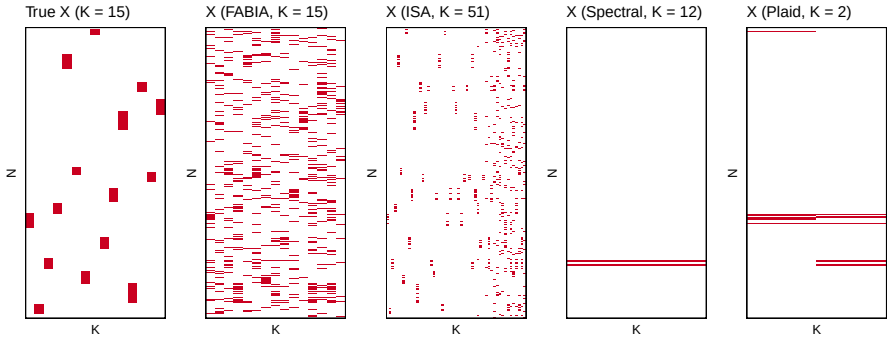
Figure 3 shows the results of SSBiEM on Simulation 1 and 2 with an initial $K^* = 30$ instead of being set to the true number of biclusters. SSBiEM can still find the true bicluster signal; however, there is no thresholding of noisy biclusters. In practice, it may be hard to distinguish between true and noisy biclusters when the actual number of biclusters is unknown.

Finally, Figure 4 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:

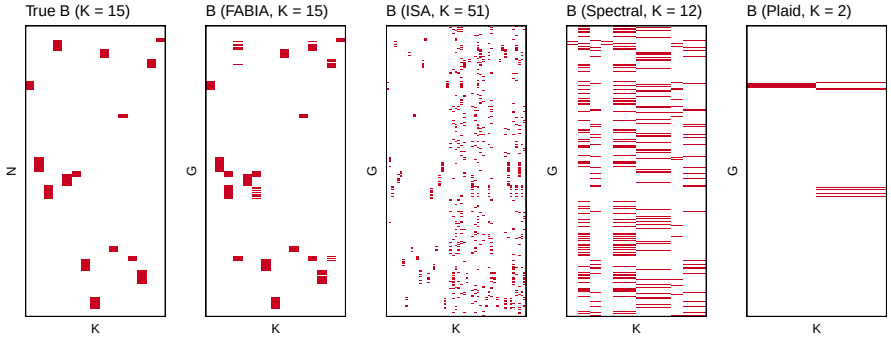
$$R^2 = 1 - \frac{\sum_{i=1}^{N} \|\mathbf{y}_i - \widehat{\mathbf{B}}\widehat{\mathbf{x}}_i\|^2}{\sum_{i=1}^{N} \|\mathbf{y}_i - \bar{\mathbf{y}}\|^2}. \tag{3.1}$$

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

Fig 1: 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.

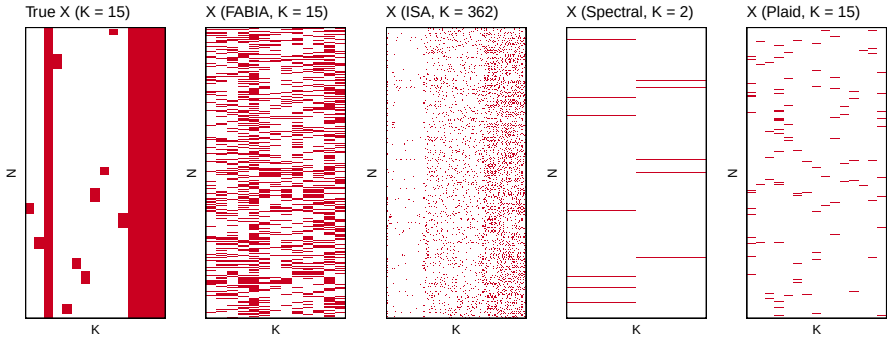


(a) Truth **(b)** FABIA **(c)** ISA **(d)** Spectral **(e)** Plaid

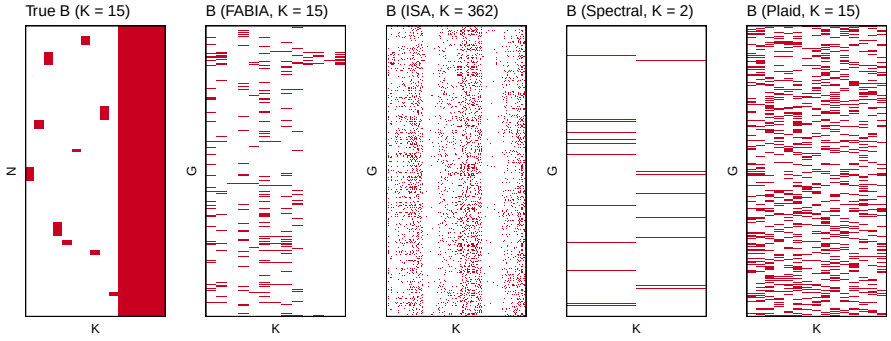


(f) Truth **(g)** FABIA **(h)** ISA **(i)** Spectral **(j)** Plaid

Fig 2: Simulation 2: 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.



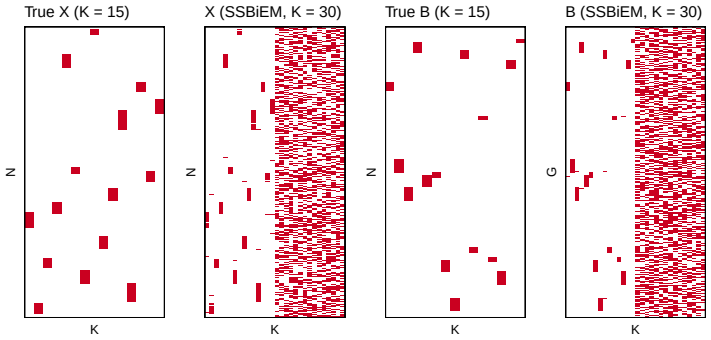
(a) Truth **(b)** FABIA **(c)** ISA **(d)** Spectral **(e)** Plaid



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

Fig 3: SSBiEM with initial $K^* = 30$

(a) Simulation 1



(b) Simulation 2

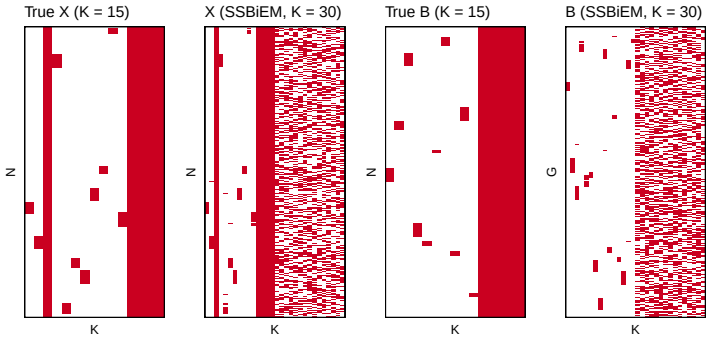
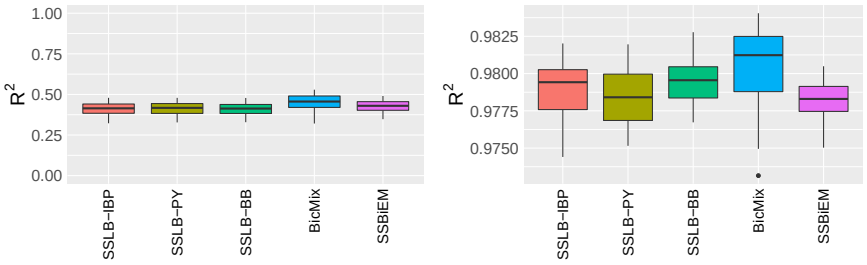


Fig 4: R^2 over 50 replications of the data.



(a) Simulation 1 **(b)** Simulation 2

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.

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

4.1. *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:

$$\mathbf{Y} = \text{Poisson}(\mathbf{X}\mathbf{B}^T). \tag{4.1}$$

Figure 6 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 1). 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 5.

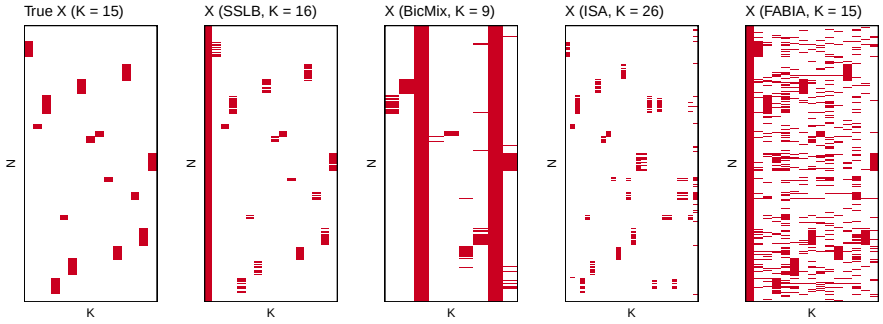
	$\widehat{K}$	
Method	Simulation 3	Simulation 4
Truth	*15*	*9*
SSLB-IBP	17.0 (0.21)	9.8 (0.19)
SSLB-PY	17.1 (0.20)	10.0 (0.18)
SSLB-BB	16.9 (0.21)	9.5 (0.15)
Bicmix	11.4 (0.23)	0.9 (0.13)
ISA	21.7 (0.52)	106.9 (3.21)
Spectral	30.6 (1.92)	1.0 (0.04)
Plaid	1.8 (0.17)	1.0 (0.00)

TABLE 1

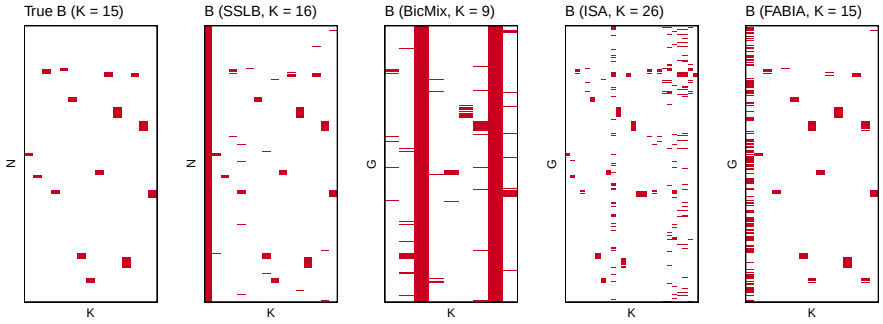
Mean estimated number of biclusters, K, over 50 replications. Standard errors are shown in parentheses.

4.2. *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 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 $\mathbf{B}$.

Fig 5: 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.

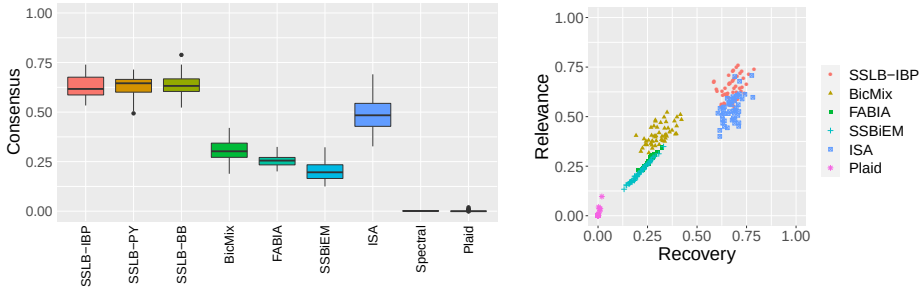


(a) Truth **(b)** SSLB-BB **(c)** BicMix **(d)** ISA **(e)** FABIA



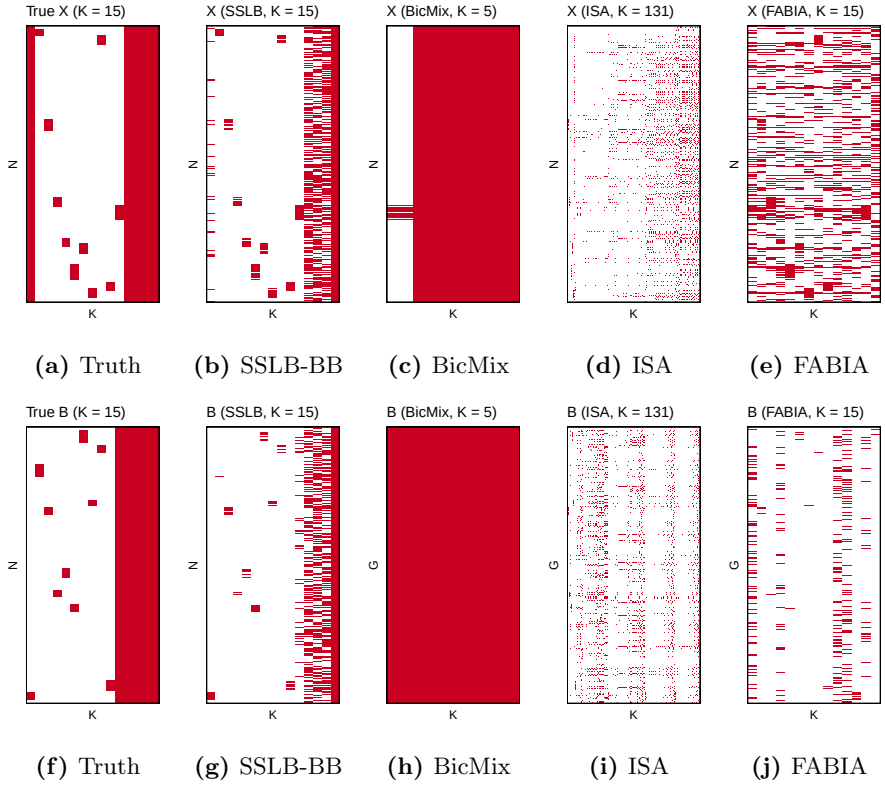
(f) Truth **(g)** SSLB-BB **(h)** BicMix **(i)** ISA **(j)** FABIA

Fig 6: Simulation 3: (a) Boxplots of the consensus scores. (b) Relevance versus recovery scores.



(a) **(b)**

Fig 7: 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.

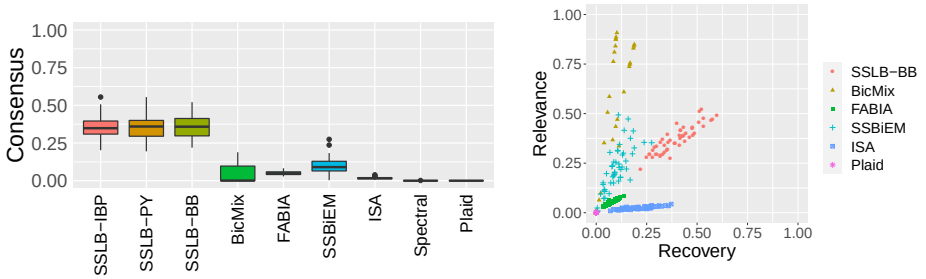


(a) Truth **(b)** SSLB-BB **(c)** BicMix **(d)** ISA **(e)** FABIA

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

In this simulation setting, the consensus of all methods are much lower than in Simulation 3 (Figure 8). For SSLB, the reduced consensus scores are due to increased false negative rates, particularly in the $\mathbf{B}$ matrix (Figure 7). 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 1). Meanwhile, SSLB slightly overestimates the number of biclusters in this setting.

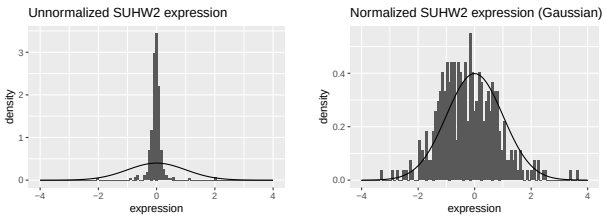
Fig 8: Simulation 4: (a) Boxplots of the consensus scores. (b) Relevance versus recovery scores.



(a) **(b)**

5. Processing Breast Cancer Data. Here, we provide more details on the processing of the breast cancer dataset in Section 4 of the main text. 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 9a). 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 9a.



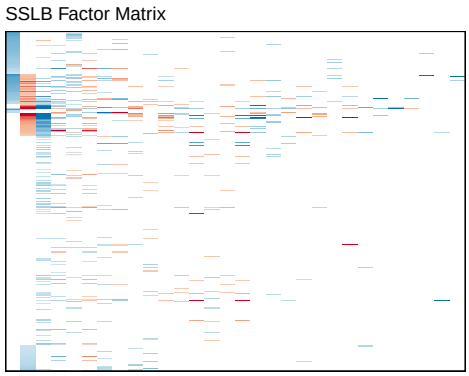
(a) **(b)**

Fig 9: 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.

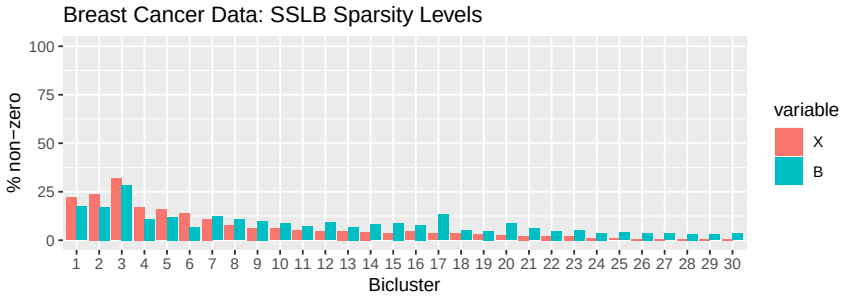
6. Additional Figures for Breast Cancer Dataset. Here, we provide additional figures for the analysis of the breast cancer microarray dataset in Section 4 of the main text. Figure 10a shows the full SSLB factor matrix, with Figure 10b showing the sparsity levels in the biclusters. The residuals from SSLB are symmetric around zero with moderately heavy tails (Figure 11a). 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 11b).

The enrichment maps (Figure 12) 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 4.3 of the main text.

Fig 10

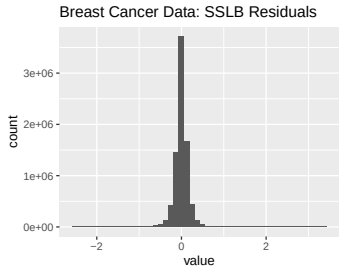


(a) SSLB factor matrix where each row corresponds to a patient and each column corresponds to a bicluster. A patient belongs to a bicluster if they have a non-zero value in that column. Rows are ordered by clinical ER status; within ER status, rows are ordered by factor values in biclusters 1 and 2.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

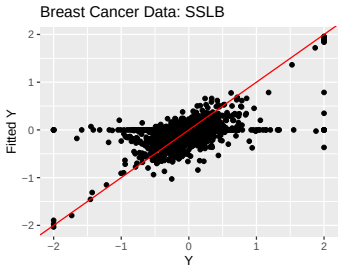


(b) Percentage of non-zero elements in each bicluster found by SSLB.

Fig 11



(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 $y = x$.

7. Processing Zeisel Dataset. Here, we describe how we processed the data in Section 5 of the main text. We followed the same pipeline as Z15 but provide the details here for completeness.

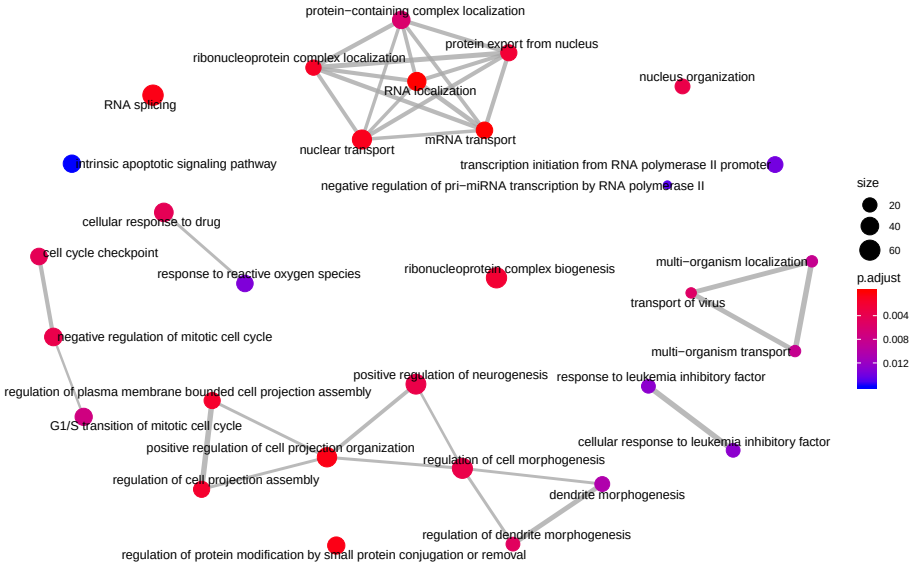
Many RNA-seq studies normalize the raw count data to the unit RPKM (Reads Per Kilobase of transcript per Million mapped reads), which accounts for longer genes having more transcripts mapped to them simply due to their length (and not meaningful biological variability). This was unnecessary for this 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.

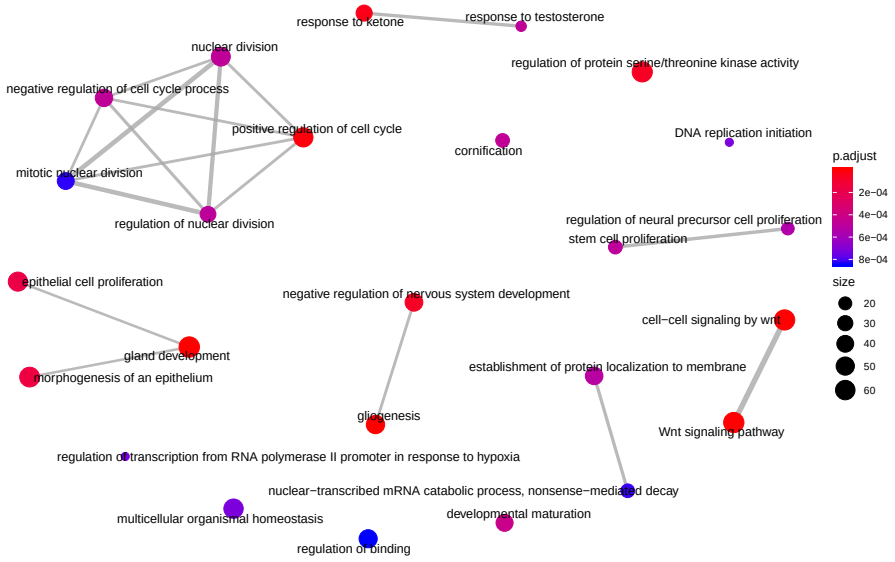
Following Z15, we:

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



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



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

Fig 12: 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.

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`[1]. 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).

8. Supplementary Figures for Zeisel Dataset. Here, we provide supplementary figures for the analysis of the mouse single-cell RNA sequencing dataset in Section 5 of the main text. Figure 13 displays full results from SSLB and FABIA. Figure 15 shows residual plots from SSLB results. SSLB residuals are very heavy tailed, but centered around zero (Figures 15a and 15b). 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 15c and 15d).

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

Enrichment maps (Figures 16 and 17) 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 of the main text.

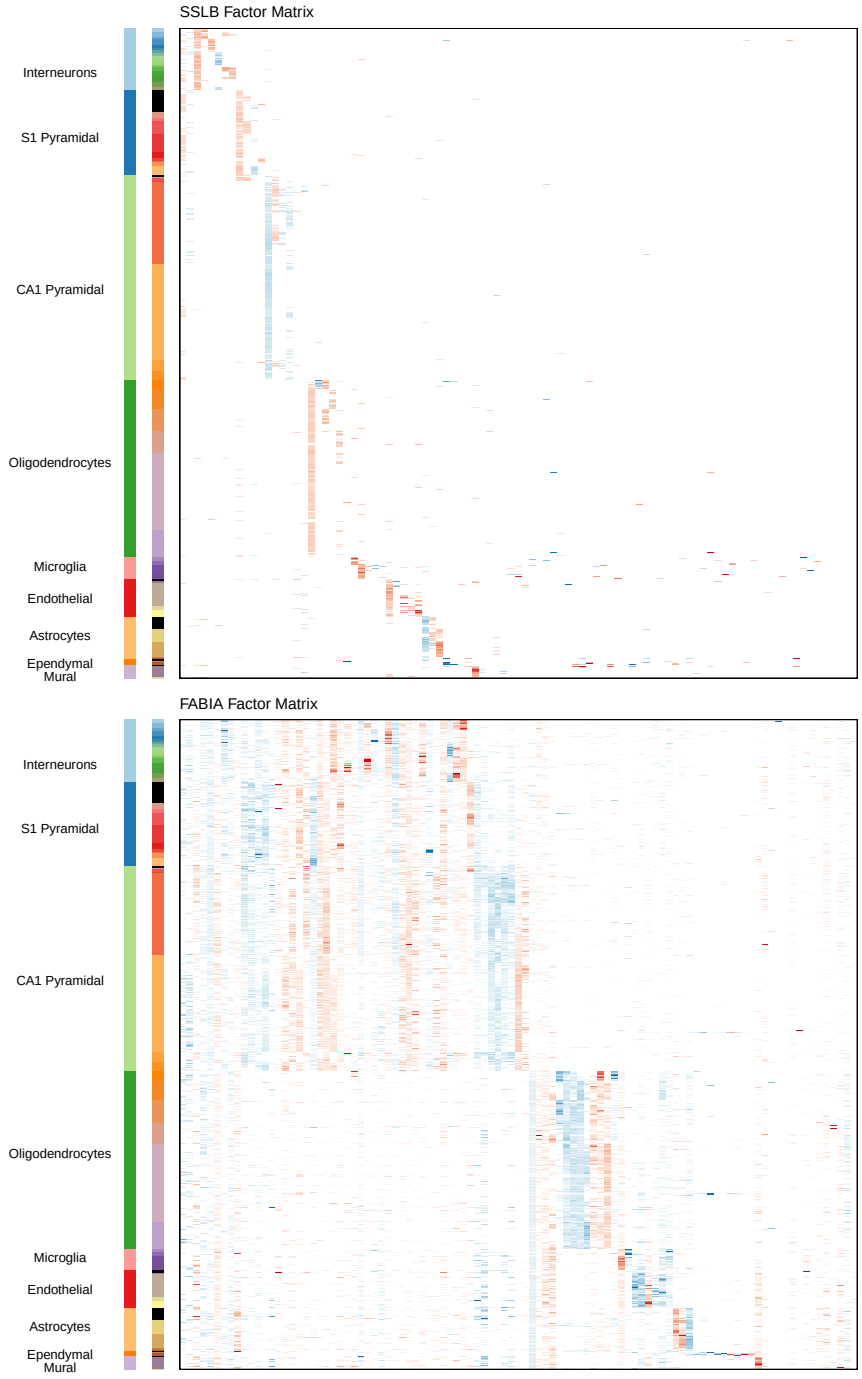


Fig 13: Zeisel dataset: Factor matrix found by SSLB (top) and FABIA (bottom). On the side of the factor matrix are the cell types and subtypes found by Z15, respectively. The rows of the factor matrices have been ordered to correspond to the Zeisel cell types. Factor values have been capped for improved visualization.

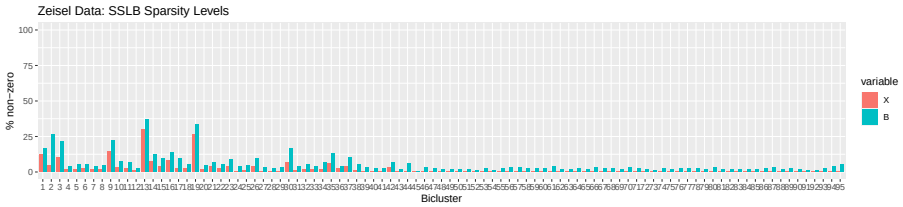
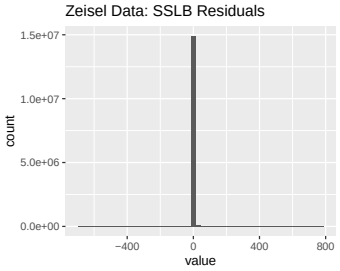
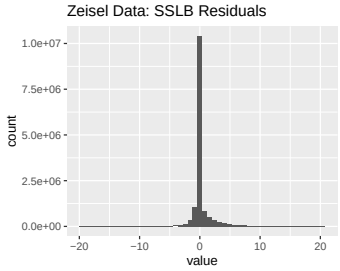


Fig 14: Percentage of non-zero elements in each bicluster found by SSLB.

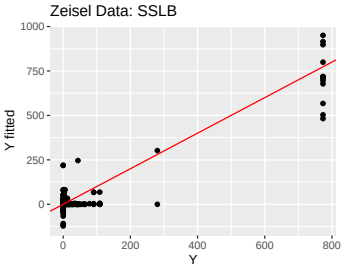
Fig 15



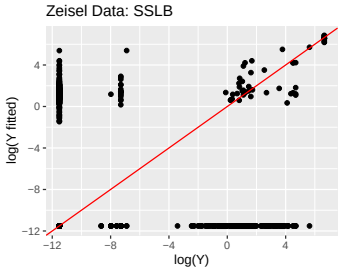
(a) Histogram of SSLB residuals.



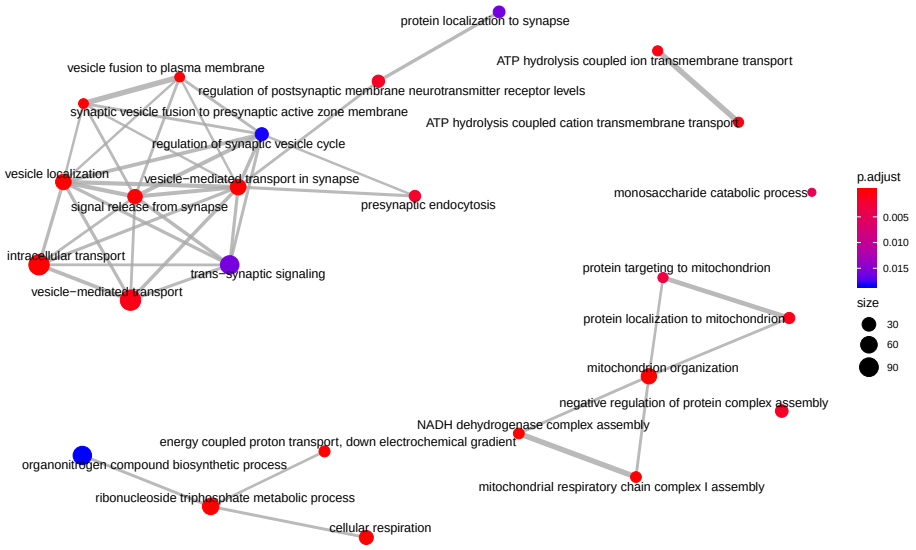
(b) Histogram of SSLB residuals with absolute values greater than 20 filtered out (removed 0.4% of the data).



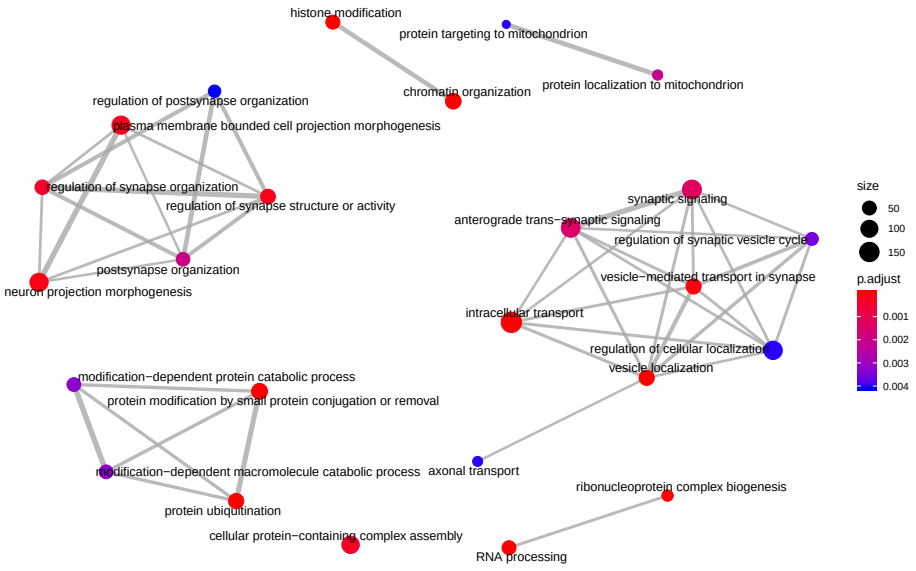
(c) Fitted SSLB $\widehat{\mathbf{Y}} = \widehat{\mathbf{X}}\widehat{\mathbf{B}}^T$ vs. observed $\mathbf{Y}$ for a randomly sampled subset of 10,000 points. Red line is $y = x$.



(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

Fig 16: 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.

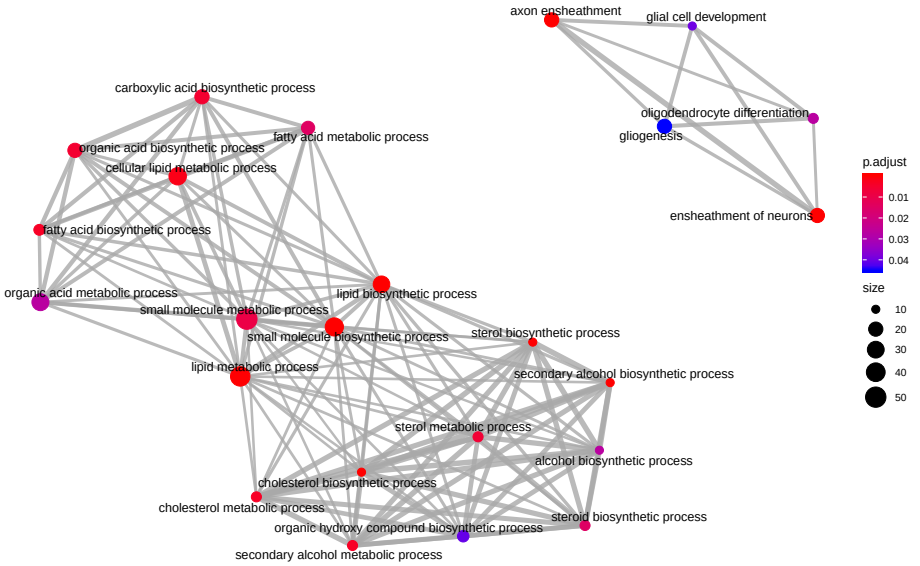


Fig 17: Zeisel dataset: enrichment map for genes in SSLB bicluster 44. Bicluster 44 contains 17 oligodendrocyte cells. Nodes represent biological processes; size of node reflects number of genes in process which were found by the method. Edges connect genes that are active in different biological processes.

References.

Bolstad, B. (2018). preprocessCore: A collection of pre-processing functions R package version 1.44.0.

Bolstad, B. M., Irizarry, R. A., Åstrand, M. and Speed, T. P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* **19** 185-193.

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

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.

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** e1004791.

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** 1520–1527.

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** 163–166.

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.

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

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

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** 1122–1129.

Ročková, V. and George, E. I. (2018). The Spike-and-Slab Lasso. *Journal of the American Statistical Association* **113** 431–444.

Yu, G. (2018). enrichplot: Visualization of Functional Enrichment Result R package version 1.2.0.

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** 1138–1142.

E-mail: gm2918@columbia.edu