Sparse Biclustering of Transposable Data

Kean Ming Tan and Daniela M. Witten

October 2, 2013

Abstract

We consider the task of simultaneously clustering the rows and columns of a large transposable data matrix. We assume that the matrix elements are normally distributed with a bicluster-specific mean term and a common variance, and perform biclustering by maximizing the corresponding log likelihood. We apply an ℓ_1 penalty to the means of the biclusters in order to obtain sparse and interpretable biclusters. Our proposal amounts to a sparse, symmetrized version of k-means clustering. We show that k-means clustering of the rows and of the columns of a data matrix can be seen as special cases of our proposal, and that a relaxation of our proposal yields the singular value decomposition. In addition, we propose a framework for biclustering based on the matrix-variate normal distribution. The performances of our proposals are demonstrated in a simulation study and on a gene expression data set. This article has supplementary material online.

Key Words: Clustering; Gene expression; ℓ_1 penalty; Matrix-variate normal distribution; Unsupervised learning.

1 Introduction

In recent years, much interest has centered around the unsupervised analysis of gene expression data and other types of high-dimensional biological data. Many proposals involve clustering the n observations on the basis of the p features, or clustering the p features on the basis of the n observations. We will refer to such proposals as *one-way clustering* in this paper, since either the rows or columns of a data matrix are clustered, but not both. An overview of some popular one-way clustering procedures can be found in Hastie et al. (2009).

In certain cases, we may be faced with *transposable* data, characterized by the fact that both the rows and columns are of scientific interest and may contain clusters or other structure (Lazzeroni

(a)

2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0

(b)

4.0	5.0	7.0	3.0
5.0	6.0	8.0	4.0
3.0	4.0	6.0	2.0
1.0	2.0	4.0	0.0

(c)

0.5	1.0	2.0	1.5
2.0	4.0	8.0	6.0
1.5	3.0	6.0	4.5
1.0	2.0	4.0	3.0

Table 1: Biclusters with (a): constant values; (b): additive coherent values; and (c): multiplicative coherent values. Table adapted from Madeira & Oliveira (2004).

& Owen 2002). One such example is gene expression data, in which the rows represent tissue samples and the columns represent genes for which expression measurements were obtained. In this case, there may be subgroups among the rows (corresponding to distinct sets of patients, perhaps with different subtypes of a disease) or subgroups among the columns (corresponding to groups of genes with shared expression patterns, potentially revealing important biological pathways) (Eisen et al. 1998). In this setting, one-way clustering seems inappropriate since it does not reflect the fact that both the rows and the columns are of scientific interest. To address this shortcoming, a number of proposals have been made for *biclustering*, which involves simultaneously clustering the rows and columns of a data matrix (among others, Cheng & Church 2000, Lazzeroni & Owen 2002, Getz et al. 2000, Tang et al. 2001, Madeira & Oliveira 2004, Cho et al. 2004, Cho & Dhillon 2008, Lee et al. 2010, Hochreiter et al. 2010). We define a *bicluster* to be a subset of the data matrix, corresponding to a set of observations and a set of features, such that all elements within the subset are *similar* to each other; some authors refer to this as a *co-cluster*. The concept of similarity must be defined based on the data set and the scientific question.

In the literature, various authors have used the term *bicluster* in different ways. Three distinct types of biclusters are displayed in Table 1. The simplest type of bicluster is a *constant bicluster* (Table 1(a)), in which all elements take on approximately a constant value. Within an *additive coherent bicluster* (Table 1(b)), an additive model holds for each element; this is related to a two-way ANOVA model. Finally, a *multiplicative coherent bicluster* (Table 1(c)) stems from a multiplicative model. Biclustering proposals have taken a number of forms, and have been aimed at detecting all three types of biclusters.

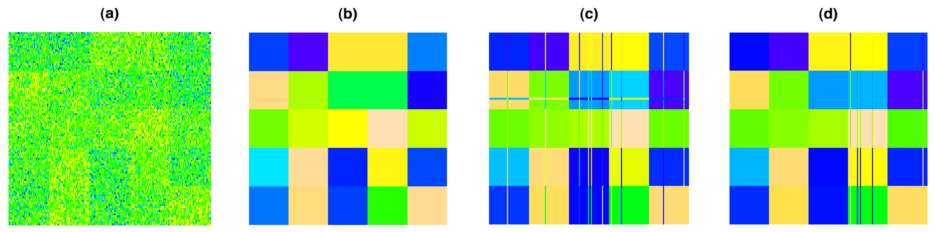

Figure 1: (a): A heatmap of a simulated 100×200 data set, with five row clusters and five column clusters. (b): True underlying mean signal within each cluster. (c): Mean signal estimated by independent 5-means clustering of the rows and 5-means clustering of the columns. (d): Mean signal estimated by biclustering, as described in Algorithm 1, with K=5, R=5, and λ=0. Biclustering results in more accurate clustering of both the rows and the columns than does independent 5-means clustering.

Gene expression data is *high-dimensional*, in the sense that $p \gg n$. In this setting, it might be reasonable to assume that most genes do not contribute much to or differ between the biological conditions being studied, and so in a sense can be considered to be noise. A number of authors have recently suggested performing *sparse* one-way clustering of the observations in gene expression data, so that just a subset of the genes are used to cluster the observations (Pan & Shen 2007, Wang & Zhu 2008, Xie et al. 2008, Witten & Tibshirani 2010). This can yield more accurate clusters, and also allows biologists to focus their research efforts on those selected genes.

In this paper, we extend sparse one-way clustering to the biclustering problem. Assume that each element of the data matrix follows a normal distribution with a bicluster-specific mean value and a common variance. We can estimate the biclusters by maximizing the corresponding log likelihood. To achieve sparse biclustering, we maximize the ℓ_1-penalized log likelihood. The proposed approach is illustrated on a toy example in Figure 1, in which it is shown that biclustering can result in more accurate cluster discovery than independent one-way clustering of the rows and columns of a data matrix. Our approach identifies constant and contiguous biclusters, as in Table 1(a).

The rest of this paper is organized as follows. In Section 2, we review the biclustering literature. Section 3 contains our proposal for sparse biclustering, and in Section 4, we motivate our biclustering proposal further by exploring its connection with the singular value decomposition. In Section 5 we present an approach for selecting the tuning parameters associated with this proposal. In Section 6 we present the results of simulation studies, and Section 7 contains an application to a gene expression data set. We propose a more general formulation for biclustering using the matrix-variate normal distribution in Section 8. The Discussion is in Section 9.

2 Past work on biclustering

In the literature, biclustering proposals have taken a number of forms, and date back to at least Hartigan (1972). For instance, some authors have independently clustered the rows and the columns of the data matrix, and others have suggested performing matrix factorization and examining the resulting singular vectors in order to identify biclusters. In addition, some biclustering proposals allow overlapping biclusters while some identify biclusters as contiguous block matrices. A detailed review of past proposals is outside of the scope of this paper, but can be found in Madeira & Oliveira (2004) and Prelic et al. (2006). Here, we briefly review three proposals for biclustering that form the basis for comparisons in the later sections of this paper. These three methods are included in comparisons because, like the proposal in this paper, they assume that most elements of the data matrix take on a common mean value. If the data matrix is centered appropriately, then this leads to a sparse estimate of the mean matrix.

Lazzeroni & Owen (2002) introduced the *plaid model* for transposable data, in which $X_{ij} = \sum_{k=1}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk}$, where ρ_{ik} and κ_{jk} are binary values that equal one if the ith observation and jth variable belong to the kth bicluster. The plaid model identifies constant biclusters when $\theta_{ijk} = \mu_k$, and additive coherent biclusters result when $\theta_{ijk} = \mu_k + \alpha_{ik} + \beta_{jk}$. The parameters are estimated by minimizing the quantity $\sum_{i=1}^{n}\sum_{j=1}^{p}(X_{ij} - \sum_{k=1}^{K}\theta_{ijk}\rho_{ik}\kappa_{jk})^2$. Turner et al. (2005) developed the *improved plaid* (IP) approach, an improved algorithm for this task, which is challenging due to the constraint that ρ_{ik} and κ_{jk} are binary.

More recently, Shabalin et al. (2009) proposed an algorithm for finding constant biclusters, termed *large average submatrices* (LAS), using the model $X_{ij} = \sum_{k=1}^{K} \mu_k \; I_{(i,j)\in B_k} + \epsilon_{ij}$, where $I_{(i,j)\in B_k}$ is an indicator function for whether the ith row and jth column belong to the kth bicluster, μ_k is a mean term, and ϵ_{ij} is a noise term. The algorithm seeks to find a bicluster that maximizes a significance score on the residual matrix obtained by subtracting out the biclusters identified in previous iterations.

An entirely different approach based on the *singular value decomposition* (SVD) is taken by Lee et al. (2010) and Hochreiter et al. (2010). They proposed to identify multiplicative biclusters using a low-rank approximation: $\mathbf{X} \approx \sum_{k=1}^{K} s_k \mathbf{u}_k \mathbf{v}_k^T$, where s_k is a scalar and $\mathbf{u}_k$ and $\mathbf{v}_k$ are vectors of lengths n and p. Lee et al. (2010) estimated the parameters subject to sparsity-inducing penalties on $\mathbf{u}_k$ and $\mathbf{v}_k$; we will refer to this as the *sparse SVD* (SSVD) approach. Hochreiter et al. (2010) imposed sparsity on the vectors $\mathbf{u}_k$ and $\mathbf{v}_k$ using a Bayesian approach. Both sets of authors declared the matrix elements corresponding to non-zero elements of $\mathbf{u}_k$ and $\mathbf{v}_k$ to make up the kth bicluster.

In this paper, we propose sparse biclustering under the assumptions that (1) each matrix element is normally distributed with a bicluster-specific mean, and (2) the biclusters partition the rows and columns of the matrix. Our proposal can be thought of as a generalization of k-means clustering to biclustering, and also a sparse and constrained version of the SVD.

3 Sparse biclustering

In what follows, $\mathbf{X}$ is a $n \times p$ matrix with n observations and p features. We assume that the n observations belong to K unknown and non-overlapping classes, $C_1, \ldots, C_K$, and the p features belong to R unknown and non-overlapping classes, $D_1, \ldots, D_R$.

3.1 An approach for biclustering

Assume that all matrix elements are independent, and that $X_{ij} \sim N(\mu_{kr}, \sigma^2)$ for $i \in C_k, j \in D_r$. We wish to estimate C_k, D_r, and μ_{kr} for $k = 1, \ldots, K$ and $r = 1, \ldots, R$. Maximizing the log likelihood of the data under this model is equivalent to

$$\underset{C_1,\ldots,C_K,D_1,\ldots,D_R,\boldsymbol{\mu}\in\mathbb{R}^{K\times R}}{\text{minimize}} \left\{ \sum_{k=1}^{K} \sum_{r=1}^{R} \sum_{i \in C_k} \sum_{j \in D_r} (X_{ij} - \mu_{kr})^2 \right\}, \tag{1}$$

which is easily seen to reduce to k-means clustering of the observations into K clusters if $R = p$, and k-means clustering of the features into R clusters if $K = n$. Note that solving (1) results in the discovery of KR biclusters, each of which consists of $|C_k||D_r|$ elements – namely, the observations in C_k and the features in D_r.

3.2 Sparse biclustering

A shortcoming of (1) is that every row cluster C_k and column cluster D_r is assigned its own mean term μ_{kr}, where $\mu_{kr} \neq 0$ in general. If the data matrix $\mathbf{X}$ is centered so that its overall mean is zero, then we may suspect that some or many biclusters have a mean term that is approximately zero. In this setting, it may be worth incurring a little bit of additional bias by estimating these mean terms to be exactly zero, in the interest of improved interpretability and reduced variance in the resulting biclusters. It is straightforward to induce sparsity on the mean elements by penalizing (1) using an ℓ_1 or *lasso* penalty (Tibshirani 1996). We arrive at

$$\underset{C_1,\ldots,C_K,D_1,\ldots,D_R,\boldsymbol{\mu}\in\mathbb{R}^{K\times R}}{\text{minimize}} \left\{ \frac{1}{2}\sum_{k=1}^{K}\sum_{r=1}^{R}\sum_{i\in C_k}\sum_{j\in D_r}(X_{ij}-\mu_{kr})^2 + \lambda\sum_{k=1}^{K}\sum_{r=1}^{R}|\mu_{kr}| \right\}, \tag{2}$$

where λ is a nonnegative tuning parameter. As λ increases, (on average) an increasing number of μ_{kr}'s will be estimated to equal zero. If $\hat{\mu}_{kr} = 0$, then this indicates a bicluster (C_k, D_r) for which the overall mean is not substantially different from zero. We note that (2) can be viewed as an extension of some recent sparse one-way clustering proposals (Pan & Shen 2007, Xie et al. 2008, Wang & Zhu 2008) to the biclustering setting, in the sense that if $R = p$ then we are performing sparse k-means clustering of the rows of the data matrix.

Algorithm 1 is a simple iterative approach for finding a local optimum of (2). It is a descent algorithm, and when $\lambda = 0$, it amounts to finding a local optimum of (1). We performed Algorithm 1 5,000 times on the same data matrix $\mathbf{X}$, generated as in Section 6.2, using random initializations of the row and column clusters. In 5,000 replications, the values of the objective function (2) were always within $\pm 0.5\%$ of the mean of the values.

We note that in the optimization problem (2), there is a complex interplay between the parameters K, R, and λ. For instance, when λ is extremely large, then $\mu_{kr} = 0$ for all $k = 1, \ldots, K$ and $r = 1, \ldots, R$, and so the values of $C_1, \ldots, C_K$ and $D_1, \ldots, D_R$ that minimize (2) are not unique. This problem can also manifest itself for more moderate values of λ. For instance, consider Step 2(a) of Algorithm 1, and suppose that $\mu_{kr} = \mu_{k'r} = 0$ for some $k \neq k'$ and for all $r = 1, \ldots, R$. Then in Step 2(b), $\sum_{r=1}^{R}\sum_{j\in D_r}(X_{ij}-\mu_{kr})^2 = \sum_{r=1}^{R}\sum_{j\in D_r}(X_{ij}-\mu_{k'r})^2$, and so C_k and $C_{k'}$ cannot be uniquely assigned. In our implementation of Algorithm 1, we address this problem when it occurs

Algorithm 1 Sparse biclustering

```
1. Initialize $D_1, \ldots, D_R$ and $C_1, \ldots, C_K$ by performing one-way $k$-means clustering on the
   columns and on the rows of the mean-centered data matrix $\mathbf{X}$.
2. Iterate until convergence:
    (a) Holding $C_1, \ldots, C_K$ and $D_1, \ldots, D_R$ fixed, solve (2) with respect to $\boldsymbol{\mu}$. That is,
            $\mu_{kr} = \frac{S(\sum_{i \in C_k} \sum_{j \in D_r} X_{ij}, \lambda)}{|C_k||D_r|}$,    (3)
        where $S$ is the soft-thresholding operator $S(a, b) = \text{sign}(a)(|a| - b)_+$, $|C_k|$ is the cardi-
        nality of $C_k$, and $|D_r|$ is the cardinality of $D_r$.
    (b) Holding $D_1, \ldots, D_R$ and $\boldsymbol{\mu}$ fixed, solve (2) with respect to $C_1, \ldots, C_K$, by assigning the
        $i$th observation to the row cluster for which $\sum_{r=1}^{R} \sum_{j \in D_r} (X_{ij} - \mu_{kr})^2$ is smallest.
    (c) Repeat Step 2(a).
    (d) Holding $C_1, \ldots, C_K$ and $\boldsymbol{\mu}$ fixed, solve (2) with respect to $D_1, \ldots, D_R$, by assigning the
        $j$th feature to the column cluster for which $\sum_{k=1}^{K} \sum_{i \in C_k} (X_{ij} - \mu_{kr})^2$ is smallest.
```

by simply merging the kth and k'th clusters, thereby reducing the total number of row clusters from K to $K - 1$. We take this approach in the interest of simplicity, though alternative procedures are possible and could lead to lower values of the objective (2).

4 A spectral interpretation for biclustering

Zha et al. (2001) established that a relaxation of k-means clustering yields principal components analysis (PCA), or equivalently, that k-means can be interpreted as a constrained version of PCA in which the kth principal component must take on values in $\{0, \frac{1}{\sqrt{n_k}}\}$. We will now show that with $K = R$ (i.e. the same number of row and column clusters), the biclustering optimization problem (1) can be relaxed in order to yield the SVD. We first present a lemma that provides an alternative characterization for the SVD.

Lemma 1. *Consider the optimization problem*

$$\underset{\mathbf{A}^T\mathbf{A}=\mathbf{I}_K, \mathbf{B}^T\mathbf{B}=\mathbf{I}_K}{\text{maximize}} ||\mathbf{A}^T\mathbf{X}\mathbf{B}||_F^2, \tag{4}$$

where $\mathbf{A}$ and $\mathbf{B}$ are $n \times K$ and $p \times K$ orthogonal matrices and $K \leq \min(n, p)$. The solution is *given by* $\mathbf{A} = \mathbf{U}_{1:K}\mathbf{Q}_1$ *and* $\mathbf{B} = \mathbf{V}_{1:K}\mathbf{Q}_2$*, where* $\mathbf{U}_{1:K}$ *and* $\mathbf{V}_{1:K}$ *are* $n \times K$ *and* $p \times K$ *matrices whose columns are the first* K *left and right singular vectors of* $\mathbf{X}$ *respectively, and* $\mathbf{Q}_1$ *and* $\mathbf{Q}_2$ *are any* $K \times K$ *orthogonal matrices.*

Finally, we present our theorem.

Theorem 4.1. *Consider the problem (4) with two additional constraints:*

1. *The elements of the kth column of* $\mathbf{A}$ *are 0 or* $\frac{1}{\sqrt{n_k}}$ *with* $n_k \in \mathbb{Z}^+$*,* $\sum_{k=1}^{K} n_k = n$*.*
2. *The elements of the kth column of* $\mathbf{B}$ *are 0 or* $\frac{1}{\sqrt{p_r}}$ *with* $p_r \in \mathbb{Z}^+$*,* $\sum_{r=1}^{K} p_r = p$*.*

This constrained version of (4) is equivalent to the biclustering optimization problem (1) with $K = R$*. Equivalently, a relaxed version of (1) yields the SVD.*

Theorem 4.1 elucidates the difference between performing independent k-means clustering on the rows and columns of a data matrix, and performing biclustering. For the relaxed problem, the two approaches are identical - that is, we know that performing PCA on the rows of a data matrix and PCA on the columns of a data matrix is equivalent to simply computing the SVD of the data matrix. However, for the constrained problem, the two approaches are different, in the sense that k-means clustering and biclustering yield different solutions. Biclustering constitutes a more symmetric and systematic approach. A result closely-related to Theorem 4.1 can be found in Cho et al. (2004).

5 Tuning parameter selection

The sparse biclustering proposal (2) involves three tuning parameters: the number of row clusters K, the number of column clusters R, and the sparsity parameter λ. Here we consider the problem of selecting these tuning parameters in an automated fashion.

5.1 Selection of K and R

In order to select K and R, we recast biclustering as a supervised learning problem, as follows. We leave out a random subset of elements from the data matrix $\mathbf{X}$, impute those left-out elements using the overall mean for the data matrix, and bicluster the resulting data matrix. We then assess the extent to which the estimated bicluster mean for the left-out elements differs from the true value of the left-out elements, using squared error loss. A related proposal appears in Witten et al. (2009). This approach, which assumes that λ is fixed, is described in greater detail in Algorithm 2.

Algorithm 2 Selecting number of row clusters K and column clusters R

```
1. Repeat the following procedure $T$ times:
   (a) Let $\mathcal{M}$ denote a set containing $np/T$ elements of the form $(i, j)$, where $(i, j)$ is drawn
       uniformly at random from $\{(1, 1), (1, 2), \dots, (n, p)\}$.
   (b) Construct a new $n \times p$ matrix, $\mathbf{X}^*$, for which the elements in $\mathcal{M}$ are "missing" and are
       imputed using the mean of the non-missing values:
       $$X^*_{ij} = \begin{cases} X_{ij} & \text{if} \quad (i,j) \in \mathcal{M}^c \\ \sum_{(i,j)\in\mathcal{M}^c} X_{ij}/|\mathcal{M}^c| & \text{if} \quad (i,j) \in \mathcal{M} \end{cases}. \qquad (5)$$
   (c) For each pair of values $(K, R)$ of interest:
       i. Perform sparse biclustering of $\mathbf{X}^*$ with $K$ row and $R$ column clusters.
       ii. Construct a $n \times p$ matrix $\mathbf{A}$ whose $(i, j)$th element equals the estimated value of
           $\mu_{kr}$, where $i \in C_k$ and $j \in D_r$.
       iii. Calculate the mean squared error that results from estimating the "missing" ele-
            ments using the corresponding bicluster means,
            $$\sum_{(i,j)\in\mathcal{M}} (X_{ij} - A_{ij})^2/|\mathcal{M}|. \qquad (6)$$
2. For each pair of values $(K, R)$ that was considered in Step 1(c), compute $m_{K,R}$, the mean of
   the quantity (6) across all $T$ iterations, as well as $s_{K,R}$, its standard error.
3. Identify the pairs $(K, R)$ for which $m_{K,R} \le m_{K+1,R+1} + s_{K+1,R+1}$.
4. Select the $(K, R)$ from Step 3 for which $K + R$ is smallest.
```

In order to explore the performance of this approach for selecting K and R, we conducted a small simulation study with various values of n, p, K, and R. First, each row was randomly assigned into one of the row clusters with uniform probability, and each column was randomly assigned to one of the column clusters with uniform probability. Then, the elements of the matrix $\mathbf{X}$ were generated independently, $X_{ij} \sim_{i.i.d.} N(\mu_{kr}, 2^2)$ for $i \in C_k, j \in D_r$ where $\mu_{kr} \sim$ Unif(-3, 3). We quantified the extent to which Algorithm 2 correctly identified the values of K and R. Occasionally, Algorithm 2 may return multiple results – for instance, two results will be returned if both $(K = 3, R = 4)$ and $(K = 4, R = 3)$ satisfy the criterion in Step 3, and no pair of (K, R)

for which $K + R < 7$ satisfies the criterion. In this case, we gave the algorithm "partial credit" according to the fraction of returned (K, R) pairs that are correct. Results are in Table 2.

Table 2: Simulation study to evaluate the performance of Algorithm 2 for tuning parameter selection. Results are reported over 50 simulated data sets. We report the overall accuracy, i.e. the proportion of the data sets for which the correct values of both K and R were identified. We also report the mean (and standard errors) of the K and R values obtained.

True value of (K, R)	n	p	Overall Accuracy	Selected K	Selected R
$(K = 2, R = 4)$	100	100	56%	2 (0)	3.48 (0.0914)
		500	66%	2 (0)	3.60 (0.0857)
	500	100	70%	2 (0)	3.68 (0.0725)
		500	94%	2 (0)	3.94 (0.0339)
$(K = 6, R = 3)$	100	100	44%	5.26 (0.1100)	3 (0.0286)
		500	74%	5.7 (0.0769)	3 (0)
	500	100	68%	5.68 (0.0666)	3 (0)
		500	94%	5.92 (0.0481)	3 (0)

5.2 Selection of λ

We now assume that K and R are known, or else were already selected using Algorithm 2 with $\lambda = 0$. We select λ using an approach motivated by BIC. For a given value of λ, we perform sparse biclustering, and create a $(np) \times (q + 1)$ design matrix, where q is equal to the number of non-zero $\hat{\mu}_{kr}$'s in the sparse biclustering output. The first column is a vector of 1's corresponding to an intercept, and the remaining columns contain 1's and 0's, indicating whether a given element of the matrix is part of the corresponding non-zero-mean bicluster in the sparse biclustering output. We fit a least squares regression model that uses this design matrix to predict the matrix elements, and compute BIC using the formula

$$\text{BIC} = np \times \log(\text{RSS}) + np \log(q)$$

where RSS is the usual residual sum of squares. We then select the value of λ that leads to the smallest value of BIC.

6 A simulation study

We compared the performance of our biclustering proposal to independent one-way k-means clustering of the rows and columns in a simulation setting with constant and contiguous non-zero biclusters (Simulation 1). In addition, we compared our biclustering proposal to a number of competitors under three simulation settings: in Simulation 2 there are constant and contiguous biclusters with some of the bicluster means exactly equal to zero, in Simulation 3 there are multiplicative biclusters, and in Simulation 4 there are overlapping biclusters.

6.1 Biclustering methods used in our comparisons

We compared the following biclustering methods, which were discussed in Sections 2 and 3.

1. Independent one-way k-means clustering of the rows and of the columns.

2. Sparse biclustering using Algorithm 1, with several values of λ.

3. IP (Turner et al. 2005), which is a variant of the plaid model (Lazzeroni & Owen 2002), using the `R` package `biclust` available on `CRAN` (Kaiser et al. 2011).

4. SSVD (Lee et al. 2010), using the `R` package `s4vd`, available on `CRAN` (Sill & Kaiser 2011).

5. LAS (Shabalin et al. 2009), using Matlab code available at `https://genome.unc.edu/las/`.

6.2 Simulation 1: No bicluster means exactly equal zero

We created $K = 4$ row clusters and $R = 5$ column clusters by randomly assigning each row to a row cluster and each column to a column cluster with uniform probability. We generated a $n \times p$ data matrix $\mathbf{X}$, according to $X_{ij} \sim_{i.i.d.} N(\mu_{kr}, 4^2)$ for $i \in C_k, j \in D_r$, where $\mu_{kr} \sim$ Unif$(-2, 2)$. Then, we mean-centered the matrix $\mathbf{X}$. We performed independent one-way k-means clustering on the rows and on the columns of the matrix, as well as sparse biclustering with various values of λ, as well as with λ selected automatically as described in Section 5.2.

The *clustering error rate* (CER; see e.g. Chipman & Tibshirani 2005, Witten & Tibshirani 2010) measures the disagreement between the true and estimated cluster labels. It is one minus the Rand index (Rand 1971). A high value of CER indicates disagreement between the true and estimated clusters, and a value of zero indicates perfect agreement. We used the CER to compare the estimated row and column clusters to the true row and column clusters. We defined the *sparsity* *rate* to be the fraction of the $\hat{\mu}_{kr}$'s that exactly equal zero, and we defined the *sparsity error rate* to be the proportion of $\hat{\mu}_{kr}$'s that were incorrectly set to zero or incorrectly set to be non-zero.

Results are reported in Table 3. We see that biclustering with $\lambda = 0$ leads to consistently better results than independent clustering of the rows and columns.

Table 3: Results from one-way k-means clustering and sparse biclustering for Simulation 1 with $n = 200$, over 50 simulated data sets. We report the mean (and standard error) of the CER of the rows and columns, and the mean (and standard error) of the sparsity rate. Note that $\bar{\lambda}$ is the mean of λ selected across 50 simulations using the approach of Section 5.2. The correct values of K and R were used, since CER is not comparable across different numbers of clusters.

p	Method	Row CER	Column CER	Sparsity Rate
200	k-means	0.0873 (0.0079)	0.1055 (0.0078)	-
	Bicluster λ=0	0.0547 (0.0066)	0.0559 (0.0056)	-
	Bicluster λ=200	0.0520 (0.0053)	0.0575 (0.0057)	0.0779 (0.0071)
	Bicluster λ=400	0.0589 (0.0063)	0.0699 (0.0065)	0.1665 (0.0111)
	Bicluster λ=800	0.0865 (0.0091)	0.0971 (0.0078)	0.2588 (0.0127)
	Bicluster $\bar{\lambda} = 320$	0.0534 (0.0057)	0.0644 (0.0063)	0.1338 (0.0110)
500	k-means	0.0254 (0.0048)	0.0755 (0.0061)	-
	Bicluster λ=0	0.0108 (0.0034)	0.0474 (0.0043)	-
	Bicluster λ=200	0.0109 (0.0032)	0.0475 (0.0044)	0.0237 (0.0052)
	Bicluster λ=400	0.0095 (0.0031)	0.0478 (0.0042)	0.0560 (0.0061)
	Bicluster λ=800	0.0122 (0.0034)	0.0557 (0.0051)	0.1158 (0.0089)
	Bicluster $\bar{\lambda} = 442$	0.0100 (0.0032)	0.0480 (0.0043)	0.0891 (0.009)

6.3 Simulation 2: Some bicluster means exactly equal zero

We modified Simulation 1 so that $\mu_{kr} \sim \text{Unif}[(-2.5, -1.5) \cup (1.5, 2.5)]$ or $\mu_{kr} = 0$ with equal probability. We compared sparse biclustering with several competitors as described in Section 6.1:

- For IP, we used the `R` package `biclust` to identify constant biclusters, with a background layer, and with *row* and *column release* parameters set to 0.5 as in Turner et al. (2005).
- For LAS, we used the default settings in the Matlab code. We discarded biclusters with a significance-based score below one, as those tend to contain the entire matrix.
- For SSVD, we obtained a rank-1 through rank-4 approximation using the `R` package `sv4d`; note that in our simulation set-up, the rank of the true underlying mean matrix is four. Sparsity parameters were selected using BIC. The adaptive weight parameters were set to two as in Lee et al. (2010). Only the best results obtained are reported.

We quantify the success of the approaches via the proportion of zero elements in the underlying mean matrix that are correctly identified (correct zeros), and the proportion of non-zero elements in the underlying mean matrix that are correctly identified (correct non-zeros). We also report sparsity rate and sparsity error rate as defined in Section 6.2. Finally, for one-way k-means clustering and for our sparse biclustering proposal, we report row and column CER; we do not report this for the other competitors, since they do not provide a partition of the rows and columns, and instead simply identify (possibly overlapping) hotspots in the matrix.

The results are presented in Table 4. We see that a substantial benefit is obtained by performing sparse biclustering rather than one-way k-means clustering, in terms of CER. Now, we discuss the performance of various biclustering methods in terms of proportion of correctly identified zeros and non-zeros, and also the sparsity error rate. We see from Table 4 that IP fails to identify any biclusters in this simulation set-up. This is due to the fact that the signal-to-noise ratio in this setting is too low; in related simulation set-ups with a higher signal-to-noise ratio, IP's performance is improved. SSVD and LAS perform comparably in this setting, and by far the best overall performance is achieved by our sparse biclustering proposal with a large value of λ. For instance, when $\lambda = 1000$ and $p = 200$, the sparsity error rate is only 14.2%.

6.4 Simulation 3: Multiplicative biclusters

This simulation study is adapted from Lee et al. (2010). Let $\mathbf{M} = d\mathbf{u}_1\mathbf{v}_1^T$ be a 100×50 matrix with $d = 50$, $\tilde{\mathbf{u}}_1 = [10, 9, 8, 7, 6, 5, 4, 3, r(2, 17), r(0, 75)]^T$, $\tilde{\mathbf{v}}_1 = [10, -10, 8, -8, 5, -5, r(3, 5), r(-3, 5), r(0, 34)]^T$, $\mathbf{u}_1 = \tilde{\mathbf{u}}_1/\|\tilde{\mathbf{u}}_1\|_2$, and $\mathbf{v}_1 = \tilde{\mathbf{v}}_1/\|\tilde{\mathbf{v}}_1\|_2$, where $r(a, b)$ denotes a vector of length b with all entries equal a. Then, let $\mathbf{X} = \mathbf{M} + \epsilon$ where $\epsilon_{ij} \sim_{i.i.d.} N(0, 1)$. Figures 2(a)-(b) display the data matrix $\mathbf{X}$ and the underlying mean matrix $\mathbf{M}$. As mentioned in Lee et al. (2010), this is a challenging biclustering problem since some non-zero entries in $\mathbf{M}$ are small relative to the noise. In particular, this setting is challenging for our sparse biclustering proposal, due to the presence of multiplicative biclusters, as opposed to the contiguous constant bicluster setting for which our proposal is intended.

We performed sparse biclustering with K, R automatically selected using Algorithm 2, and with various values of λ. For IP, LAS, and SSVD, the tuning parameters used are as in Section 6.3 unless specified otherwise. For IP, we set the `R` package `biclust` to identify the most flexible model discussed in Lazzeroni & Owen (2002), and ran the algorithm without the background layer. For

SSVD, we set the parameters in the `R` package `s4vd` such that one bicluster is identified.

The results (averaged over 100 simulations) are summarized in Table 5. It is not surprising that SSVD has the best results in this simulation set-up, as in this set-up there are multiplicative biclusters. Though they have low sparsity error rates, both IP and LAS fail to correctly identify most of the non-zero elements in the underlying mean matrix. It is not surprising that LAS performs poorly in this simulation set-up, as LAS was developed to identify constant biclusters.

How does sparse biclustering perform in this setting, which clearly violates the constant and contiguous bicluster model? Sparse biclustering with $\lambda = 0$ has a sparsity error rate of 0.92, due to the fact that when $\lambda = 0$, all elements in the estimated mean matrix are non-zero. However, for a moderate value of λ, sparse biclustering performs well, even though it is designed to identify contiguous constant biclusters. This is because the multiplicative bicluster in Figure 2(b) can be approximated as the *union of a number of constant biclusters.* Therefore, sparse biclustering leads to Figure 2(c), which is a very accurate approximation of Figure 2(b). In particular, Figure 2(c) resulted from our sparse biclustering proposal with $K = 3$ and $R = 5$; note that these values were selected automatically by Algorithm 2.

6.5 Simulation 4: Overlapping multiplicative biclusters

In this section, we investigate an example with overlapping multiplicative biclusters. Let $\mathbf{M} = d\mathbf{u}_1\mathbf{v}_1^T + d\mathbf{u}_2\mathbf{v}_2^T$ be a 100×50 matrix with $d = 50$, $\mathbf{u}_1, \mathbf{v}_1$ as defined in Section 6.4, $\tilde{\mathbf{u}}_2 = [r(0,13), 10, 9, 8, 7, 6, 5, 4, 3, r(2,17), r(0,62)]^T$, $\tilde{\mathbf{v}}_2 = [r(0,9), 10, -9, 8, -7, 6, -5, r(4,5), r(-3,5), r(0,25)]^T$, $\mathbf{u}_2 = \tilde{\mathbf{u}}_2/\|\tilde{\mathbf{u}}_2\|$, and $\mathbf{v}_2 = \tilde{\mathbf{v}}_2/\|\tilde{\mathbf{v}}_2\|$. Then, let $\mathbf{X} = \mathbf{M} + \epsilon$ where $\epsilon_{ij} \sim_{i.i.d.} N(0,1)$. Heatmaps of $\mathbf{X}$ and $\mathbf{M}$ are shown in Figures 3(a)-(b).

We performed the biclustering methods described in the previous section, with the SSVD parameters set to identify two biclusters. We expect SSVD to perform well in this set-up, since there are multiplicative overlapping biclusters. In contrast, sparse biclustering's assumption of constant and non-overlapping biclusters is clearly violated. Nonetheless, sparse biclustering performs competitively (Table 6), since the multiplicative and overlapping biclusters can be very accurately approximated using sparse biclustering using a sufficiently large value of K and R (Figure 3(c)). A similar fact was noted in Gu & Liu (2008).

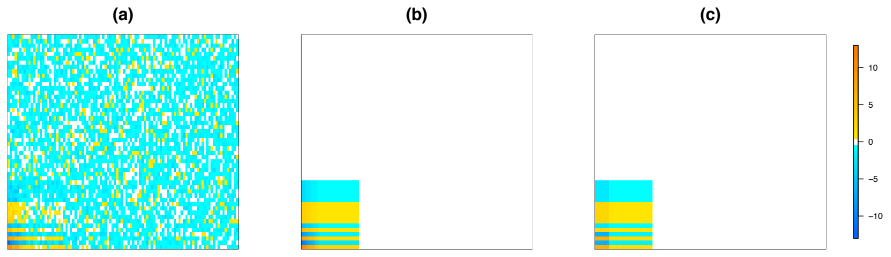

Figure 2: Heatmaps of (a): data matrix, generated according to Simulation 3. (b) Underlying means used to generate data. (c) Mean matrix estimated by sparse biclustering, with K and R automatically chosen ($K = 3$, $R = 5$) and $\lambda = 10$; 84% of the elements are estimated to equal zero.

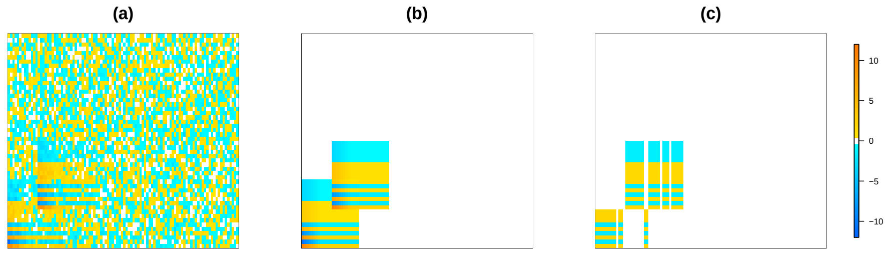

Figure 3: Heatmaps of (a): data matrix, generated according to Simulation 4. (b) Underlying means used to generate data. (c) Mean matrix estimated by sparse biclustering, with K and R automatically chosen ($K = 3$, $R = 6$) and $\lambda = 70$; 88% of the elements are exactly equal to zero.

7 Application to a gene expression data set

In this section, we consider a lung cancer gene expression data set previously analyzed by Lee et al. (2010) and Liu et al. (2008), consisting of measurements for 56 samples and 12,625 genes. 17 samples correspond to normal subjects, 20 correspond to subjects with pulmonary carnicoid tumors, 13 correspond to colon metastases, and six correspond to small cell carnicomas. We selected 5,000 genes with largest variance, and we mean-centered the 56×5000 data matrix. The goal is to discover sets of genes whose expression differs from the baseline in a subset of the patients.

We performed sparse biclustering using $K = 4$ (which we know to be the true number of row clusters), $R = 10$, and $\lambda = 1500$. A heatmap of the resulting estimated mean matrix is shown in Figure 4. For visualization purposes, we reordered the genes based on the estimated clusters to which they belong. From Figure 4, we see that one subject with small cell carnicoma is assigned to a cluster of pulmonary carcinoid tumors via sparse biclustering. Imposing sparsity in estimating the bicluster means provides substantial benefits in interpretation of the image plot, as $\hat{\mu}_{kr} = 0$ for many values of k and r. Furthermore, we see from Figure 4 that there is substantial variation among the estimated bicluster means. For instance, the genes in the second column cluster have a very large mean value in normal patients and a very small mean value in carcinoid patients.

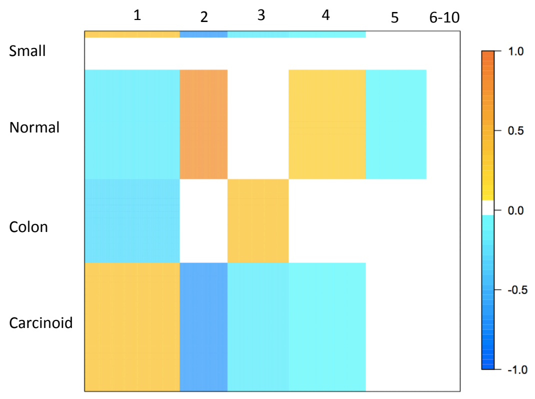

Figure 4: Heatmap of the estimated mean matrix from sparse biclustering using $K = 4$, $R = 10$, and $\lambda = 1500$ on a subset of the lung cancer data set consisting of the 5,000 genes with highest variance. The rows are ordered by true cancer subtype. The genes are reordered based on the estimated clusters for visualization purposes. The column labels are the gene clusters. Note that all elements in column clusters 6-10 are estimated to equal zero.

The estimated mean matrix shown in Figure 4 is similar to the three image plots obtained using SSVD in Lee et al. (2010). This is not surprising, since our biclustering proposal can be interpreted as a constrained version of the SVD (see Section 4). However, SSVD has a major interpretational disadvantage relative to our proposal: whereas sparse biclustering explicitly returns cluster labels for both the rows and columns of the data matrix, the SSVD instead returns a series of sparse singular vectors. The analyst must then take a *post hoc* approach to interpret these singular vectors in order to determine the row and column clusters. In other words, SSVD does not directly output a single interpretable figure as in Figure 4.

We note that Algorithm 2 led to selection of $K = 5$ and $R = 25$ on this example. One of these row clusters contains just a single subject, and the others correspond perfectly to the subjects' cancer types. Here we reported results using $R = 10$ instead of $R = 25$ for simplicity; however, using $R = 25$, a figure that is qualitatively very similar to Figure 4 emerges.

8 Matrix-variate normal biclustering

Recently, proposals have emerged to use the matrix-variate normal distribution to model high-dimensional transposable data (Gupta & Nagar 1999, Allen & Tibshirani 2010). To indicate that a $n \times p$ data matrix $\mathbf{X}$ has a matrix-variate normal distribution, we write

$$\mathbf{X} \sim MVN(\mathbf{A}, \mathbf{\Sigma}, \mathbf{\Delta}), \tag{7}$$

where $\mathbf{A}$ is a $n \times p$ matrix containing the mean for each element of $\mathbf{X}$, $\mathbf{\Sigma}$ is a $n \times n$ covariance matrix for the rows of $\mathbf{X}$, and $\mathbf{\Delta}$ is a $p \times p$ covariance matrix for the columns of $\mathbf{X}$. A consequence of the matrix-variate normal model (7) is that the rows and columns of $\mathbf{X}$ are marginally multivariate normal. For instance, letting $\mathbf{X}_i$ and $\mathbf{A}_i$ be the ith rows of $\mathbf{X}$ and $\mathbf{A}$, respectively, then

$$\mathbf{X}_i \sim N(\mathbf{A}_i, \Sigma_{ii}\mathbf{\Delta}). \tag{8}$$

We note that in the case $\mathbf{\Sigma} = \mathbf{\Delta} = \mathbf{I}$, this model reduces to $X_{ij} \sim_{i.i.d.} N(0, 1)$.

8.1 General formulation of matrix-variate normal biclustering

Now assume that the $n \times p$ data matrix $\mathbf{X}$ is drawn from a matrix-variate normal distribution of the form (7) and that $\mathbf{A}$ has *constant biclusters*: that is, for all $i \in C_k$ and $j \in D_r$, $A_{ij} = \mu_{kr}$. Without loss of generality, suppose that the rows and columns are ordered such that $k < k'$, $i \in C_k$, and $i' \in C_{k'}$ implies that $i < i'$, and similarly $r < r'$, $j \in D_r$, and $j' \in D_{r'}$ implies that $j < j'$. In other words, we use the model

$$\mathbf{X} \sim MVN\left(\begin{pmatrix} (\mu_{11}) & \dots & (\mu_{1R}) \\ \vdots & \ddots & \vdots \\ (\mu_{K1}) & \dots & (\mu_{KR}) \end{pmatrix}, \mathbf{\Sigma}, \mathbf{\Delta}\right), \tag{9}$$

where (μ_{kr}) is a $|C_k| \times |D_r|$ matrix, all of whose elements equal μ_{kr}. This is a natural formulation for biclustering since it easily accommodates constant biclusters as well as arbitrary row and column covariances. Fitting the model (9) requires estimating the $n \times n$ matrix $\mathbf{\Sigma}$ and the $p \times p$ matrix $\mathbf{\Delta}$ using the $n \times p$ matrix $\mathbf{X}$; a proposal to do this using ℓ_1 or ℓ_2 penalties is presented in Allen &

Tibshirani (2010).

A further simplification to the model (9) is natural. Though we might expect correlation between observations within a row cluster, or between features within a column cluster, correlations between observations in two different row clusters or between features in two different column clusters are less easily interpreted. This leads to the model

$$\mathbf{X} \sim MVN\left(\begin{pmatrix} (\mu_{11}) & \dots & (\mu_{1R}) \\ \vdots & \ddots & \vdots \\ (\mu_{K1}) & \dots & (\mu_{KR}) \end{pmatrix}, \begin{pmatrix} \mathbf{\Sigma}_1 & & \\ & \ddots & \\ & & \mathbf{\Sigma}_K \end{pmatrix}, \begin{pmatrix} \mathbf{\Delta}_1 & & \\ & \ddots & \\ & & \mathbf{\Delta}_R \end{pmatrix}\right), \tag{10}$$

where $\mathbf{\Sigma}$ and $\mathbf{\Delta}$ are now block diagonal with blocks of dimension $|C_1| \times |C_1|, \dots, |C_K| \times |C_K|$ and $|D_1| \times |D_1|, \dots, |D_R| \times |D_R|$, respectively. The formulation (10) is attractive not only because it provides a natural model for biclustering, but also because it has as special cases some well-known formulations for one-way clustering. In particular, consider (10) with $R = p$ and $\mathbf{\Sigma}_k = \mathbf{I}$ for $k = 1, \dots, K$. Then (10) amounts to a simple and well-studied model in which all observations come from a multivariate normal distribution with a common diagonal covariance matrix and a cluster-specific mean vector (Fraley & Raftery 2002). If furthermore $\mathbf{\Delta} = \sigma^2\mathbf{I}$, then this amounts to the usual formulation for one-way k-means clustering. By symmetry of the matrix normal distribution, (10) also reduces to model-based clustering or k-means clustering of the columns. Note that if we assume that $\mathbf{\Sigma} = \sigma^2\mathbf{I}$ and $\mathbf{\Delta} = \mathbf{I}$, then this corresponds to our proposal in Section 3.

8.2 Sparse matrix-variate normal biclustering

The log likelihood corresponding to (10) takes the form

$$l(\boldsymbol{\mu}, \mathbf{\Sigma}, \mathbf{\Delta}) = \frac{p}{2}\sum_{k=1}^{K} \log|\mathbf{\Sigma}_k^{-1}| + \frac{n}{2}\sum_{r=1}^{R}\log|\mathbf{\Delta}_r^{-1}| - \frac{1}{2}\sum_{k=1}^{K}\sum_{r=1}^{R} \mathrm{tr}(\mathbf{\Sigma}_k^{-1}(\mathbf{X}_{k,r} - \mu_{kr})\mathbf{\Delta}_r^{-1}(\mathbf{X}_{k,r} - \mu_{kr})^T), \tag{11}$$

where $\mathbf{X}_{k,r}$ is a $|C_k| \times |D_r|$ submatrix of $\mathbf{X}$ that consists of the elements X_{ij} for $i \in C_k$ and $j \in D_r$. We would like to fit the model (10) by maximizing (11). However, two problems arise. First, the maximum likelihood estimates of $\mathbf{\Sigma}_k$ and $\mathbf{\Delta}_r$ may be singular. Second, we may want to encourage sparsity in μ_{kr}. To address these two points, we propose to maximize the penalized log likelihood

$$\begin{aligned} l_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\Delta}) &= \frac{p}{2}\sum_{k=1}^{K} \log|\boldsymbol{\Sigma}_k^{-1}| + \frac{n}{2}\sum_{r=1}^{R} \log|\boldsymbol{\Delta}_r^{-1}| - \frac{1}{2}\sum_{k=1}^{K}\sum_{r=1}^{R} \text{tr}\left(\boldsymbol{\Sigma}_k^{-1}(\mathbf{X}_{k,r} - \mu_{kr})\boldsymbol{\Delta}_r^{-1}(\mathbf{X}_{k,r} - \mu_{kr})^T\right) \\ &\quad -\lambda\sum_{k=1}^{K}\sum_{r=1}^{R}|\mu_{kr}| - \alpha\sum_{k=1}^{K}||\boldsymbol{\Sigma}_k^{-1}||^d - \beta\sum_{r=1}^{R}||\boldsymbol{\Delta}_r^{-1}||^d. \end{aligned} \quad (12)$$

Here, α, β, and λ are nonnegative parameters that determine the extent of penalization. We take $d = 1$ or $d = 2$. The last two terms in (12) can be understood as $||\mathbf{W}||^d = \sum_{i,j}|W_{ij}|^d$.

To maximize (12), we take an iterative approach in which we update the parameters $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, $\boldsymbol{\Delta}$, $C_1, \ldots, C_K$, $D_1, \ldots, D_R$ sequentially, holding all other parameters fixed as we update the current set of parameters. We begin with two simple lemmas.

Lemma 2. *With $\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1}$, $\boldsymbol{\Delta}_1^{-1}, \ldots, \boldsymbol{\Delta}_R^{-1}$, $C_1, \ldots, C_K$, and $D_1, \ldots, D_R$ held fixed, then maximizing (12) with respect to $\boldsymbol{\mu}$ results in the update*

$$\mu_{kr} = S\left(\frac{\text{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{1}\boldsymbol{\Delta}_r^{-1}\mathbf{X}_{k,r}^T)}{\text{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{1}\boldsymbol{\Delta}_r^{-1}\mathbf{1}^T)}, \frac{\lambda}{\text{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{1}\boldsymbol{\Delta}_r^{-1}\mathbf{1}^T)}\right), \quad (13)$$

where $\mathbf{1}$ is a $|C_k| \times |D_r|$ matrix comprised solely of 1's, and S is the soft-thresholding operator.

Lemma 3. *With $\boldsymbol{\mu}$, $\boldsymbol{\Delta}_1^{-1}, \ldots, \boldsymbol{\Delta}_R^{-1}$, $C_1, \ldots, C_K$, and $D_1, \ldots, D_R$ held fixed, maximizing (12) with respect to $\boldsymbol{\Sigma}_k^{-1}$ reduces to*

$$\underset{\boldsymbol{\Sigma}_k^{-1}}{\text{maximize}}\left\{\log|\boldsymbol{\Sigma}_k^{-1}| - \text{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{S}_k) - (2\alpha/p)||\boldsymbol{\Sigma}_k^{-1}||^d\right\} \quad (14)$$

where $\mathbf{S}_k = \frac{1}{p}\sum_{r=1}^{R}(\mathbf{X}_{k,r} - \mu_{kr})\boldsymbol{\Delta}_r^{-1}(\mathbf{X}_{k,r} - \mu_{kr})^T$.

Note that if $d = 1$, the graphical lasso algorithm (Friedman et al. 2007) can be used to solve (14), and the estimate for $\boldsymbol{\Sigma}_k^{-1}$ will be sparse if the tuning parameter α is sufficiently large. When $d = 2$, then a simple analytical solution in terms of the eigenvectors and eigenvalues of $\mathbf{S}_k$ is available (Witten & Tibshirani 2009). A similar approach can be used to solve (12) with respect to $\boldsymbol{\Delta}_r^{-1}$, with $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1}$ held fixed.

In order to update $C_1, \ldots, C_K$ with $D_1, \ldots, D_R$, $\boldsymbol{\Delta}^{-1}$, $\boldsymbol{\Sigma}^{-1}$, and $\boldsymbol{\mu}$ held fixed, we note that by

(8), the ith row of $\mathbf{X}$ has a multivariate normal distribution given by

$$\mathbf{X}_i \sim N(\boldsymbol{\mu}_k, \Sigma_{ii}\boldsymbol{\Delta}) \tag{15}$$

if that observation belongs to the kth cluster. In (15), $\boldsymbol{\mu}_k$ is a p-vector whose jth element equals μ_{kr} if $j \in D_r$. So we update the row cluster of the ith observation by assigning that observation to the class for which the log likelihood resulting from (15) is largest. We note that this approach for updating the row clusters is not completely rigorous, since we are assigning each observation to a new row cluster without regard to the covariance structure among the rows. In particular, this approach is not guaranteed to increase the log likelihood, but performs well empirically. A similar approach is taken to update the column clusters.

The steps just described for maximizing (12) are summarized in Algorithm 3. Although Steps 2(b) and 2(d) in Algorithm 3 could potentially lead to a decrease in (12), in our experience, the algorithm tends to converge within 35 iterations in the simulation set-up of Section 8.3.

Algorithm 3 Matrix-variate normal biclustering

```
1. Initialize $C_1, \ldots, C_K$, $D_1, \ldots, D_R$, $\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1}$, $\boldsymbol{\Delta}_1^{-1}, \ldots, \boldsymbol{\Delta}_R^{-1}$, $\boldsymbol{\mu}$.
2. Iterate until convergence or until a fixed number of iterations is reached:
    (a) Holding $C_1, \ldots, C_K$ and $D_1, \ldots, D_R$ fixed, perform the following updates:
        i. Holding $\boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{\Delta}^{-1}$ fixed, update $\boldsymbol{\mu}$ using (13).
        ii. Holding $\boldsymbol{\mu}$ and $\boldsymbol{\Delta}^{-1}$ fixed, update $\boldsymbol{\Sigma}_k^{-1}$ as in Lemma 3 for $k = 1, \ldots, K$.
        iii. Holding $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}^{-1}$ fixed, update $\boldsymbol{\Delta}_r^{-1}$ as in Lemma 3 for $r = 1, \ldots, R$.
    (b) Holding $\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1}$, $\boldsymbol{\Delta}_1^{-1}, \ldots, \boldsymbol{\Delta}_R^{-1}$, $\boldsymbol{\mu}$, and $D_1, \ldots, D_R$ fixed, update the row clustering. To do this, iterate through the rows and assign each row to the row cluster for which the log likelihood resulting from (15) is largest.
    (c) Repeat Step 2(a).
    (d) Holding $\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1}$, $\boldsymbol{\Delta}_1^{-1}, \ldots, \boldsymbol{\Delta}_R^{-1}$, $\boldsymbol{\mu}$, and $C_1, \ldots, C_K$ fixed, update the column clustering, as in Step 2(b), with the roles of the rows and columns reversed.
```

8.3 A simulation study

We created $K = 4$ row and $R = 5$ column clusters by randomly assigning each row to a row cluster with uniform probability, and each column to a column cluster with uniform probability.

We generated a $n \times p$ mean matrix $\mathbf{A}$ as follows: for each $i \in C_k$ and $j \in D_r$, $A_{ij} = \mu_{kr}$, where $\mu_{kr} \sim \text{Unif}[(-2.5, -1.5) \cup (1.5, 2.5)]$ or $\mu_{kr} = 0$ with equal probability. Then, the $n \times p$ matrix $\mathbf{X}$ is generated according to $\mathbf{X} \sim MVN(\mathbf{A}, \mathbf{\Sigma}, \mathbf{\Delta})$, where $\mathbf{\Sigma}$ and $\mathbf{\Delta}$ are block diagonal covariance matrices with blocks corresponding to the row and column cluster memberships, respectively.

We performed one-way k-means clustering on the rows and on the columns, sparse biclustering, and matrix-variate normal biclustering with $d = 1$. We considered the cases when $\mathbf{\Sigma}^{-1}$ and $\mathbf{\Delta}^{-1}$ are known and unknown. We set the tuning parameters α and β in (12) to equal 0.05. In addition, we considered IP, LAS, and SSVD, where the tuning parameters were chosen as described in Section 6.3. The same evaluation criteria as in Section 6.3 were used to evaluate the performance of various biclustering methods. Results are reported in Table 7.

We see that matrix-variate normal biclustering leads to consistently better results than sparse biclustering and one-way clustering of the rows and columns via k-means. When both $\mathbf{\Sigma}^{-1}$ and $\mathbf{\Delta}^{-1}$ are known, matrix-variate normal biclustering results in the lowest CER.

8.4 Application to real data

We again consider the lung cancer data set described in Section 7. Once again, we selected 5,000 genes with largest variance, and mean-centered the data matrix. We performed MVN biclustering with $K = 4$, $R = 10$, $\lambda = 1500$, $\alpha = 0.35$, $\beta = 0.35$, and $d = 1$, where α, β, and d are given in (12). A heatmap of the estimated mean matrix resulting from MVN biclustering is shown in Figure 5.

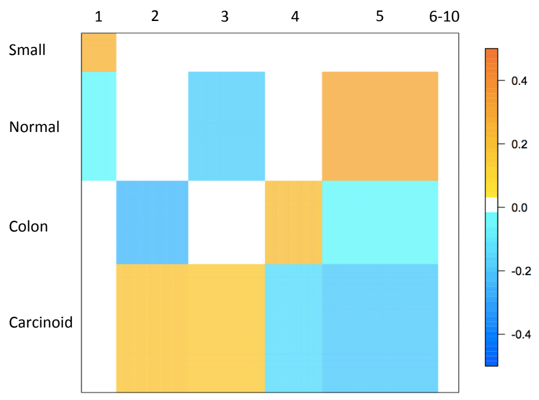

Figure 5: Heatmap of the estimated mean matrix from MVN biclustering using $K = 4$, $R = 10$, $\lambda = 1500$, $\alpha = 0.35$, and $\beta = 0.35$ on a subset of the lung cancer data set consisting of the 5,000 genes with highest variance. Details are as in Figure 4.

We see from Figure 5 that MVN biclustering perfectly identifies the four types of subjects. On this data set, since α is large and n is small, the estimate for $\mathbf{\Sigma}^{-1}$ obtained is diagonal – in other words, here our MVN biclustering does not model conditional dependencies among the samples. In contrast, the estimate obtained for $\mathbf{\Delta}^{-1}$ has many non-zero elements within each of the blocks. In particular, 13.45% of the partial correlations in cluster 1, 73% of the partial correlations in cluster 2, 58.23% of the partial correlations in cluster 3, 40.96% of the partial correlations in cluster 4, 73.22% of the partial correlations in cluster 5, and 0.057% of the partial correlations in clusters 6-10 are non-zero. By inspection of Figure 5, we see that the gene clusters with expression levels that differ substantially among cancer subtypes tend to contain genes that are conditionally dependent. This is scientifically plausible, since we believe that genes that participate in the same pathways tend to be conditionally dependent, and may have similar expression levels in each biological condition.

9 Discussion

In this paper, we have proposed a novel approach for biclustering. Sparsity in the bicluster means is achieved using an ℓ_1 penalty, and our biclustering proposal is extended to a more general setting using the matrix-variate normal distribution. We have shown that k-means clustering can be seen as a special case of our biclustering proposal. Just as a relaxation of k-means clustering yields PCA, a relaxation of our biclustering approach yields the SVD.

A possible drawback of our sparse biclustering proposal is that it does not allow for overlapping biclusters — that is, it assigns each element of the data matrix to exactly one bicluster. While allowing for overlapping biclusters can be beneficial in certain contexts (Madeira & Oliveira 2004), we argue that it results in too much complexity as well as challenges in interpretation. Furthermore, we demonstrate in Sections 6.4 and 6.5 that even though our sparse biclustering proposal assumes constant and contiguous biclusters, it performs competitively when there are multiplicative biclusters and overlapping biclusters.

The `R` package `sparseBC`, available on `CRAN`, implements the methods proposed.

Supplementary materials

R package for sparse biclustering and MVN biclustering: R package `sparseBC` available on CRAN.

R script for Figures 1-5 R script to reproduce Figures 1-5 in the manuscript. (Figures-code.zip)

R script for Tables 2-7: R script to reproduce the results in Tables 2-7. (Tables-code.zip)

Acknowledgments

We thank the editor, two associate editors, and two reviewers for helpful comments that improved the quality of this manuscript. The authors were supported by NIH Grant DP5OD009145 and NSF CAREER Award DMS-1252624.

References

Allen, G. & Tibshirani, R. (2010), 'Transposable regularized covariance models with an application to missing data imputation', *Annals of Applied Statistics* **4(2)**, 764–790.

Cheng, Y. & Church, G. (2000), 'Biclustering of gene expression data', *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **8**, 93–103.

Chipman, H. & Tibshirani, R. (2005), 'Hybrid hierarchical clustering with applications to microarray data', *Biostatistics* **7**, 286–301.

Cho, H. & Dhillon, I. S. (2008), 'Coclustering of human cancer microarrays using minimum sum-squared residue coclustering', *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* **5**(3), 385–400.

Cho, H., Dhillon, I. S., Guan, Y. & Sra, S. (2004), Minimum sum-squared residue co-clustering of gene expression data., *in* 'Proceedings of the Fourth SIAM International Conference on Data Mining', pp. 114–125.

Eisen, M., Spellman, P., Brown, P. & Botstein, D. (1998), 'Cluster analysis and display of genome-wide expression patterns', *Proc. Natl. Acad. Sci., USA.* **95**, 14863–14868.

Fraley, C. & Raftery, A. (2002), 'Model-based clustering, discriminant analysis, and density estimation', *J. Amer. Statist. Assoc.* **97**, 611–631.

Friedman, J., Hastie, T. & Tibshirani, R. (2007), 'Sparse inverse covariance estimation with the graphical lasso', *Biostatistics* **9**, 432–441.

Getz, G., Levine, E. & Domany, E. (2000), 'Coupled two-way clustering of gene microarray data', *PNAS* **97**, 12079–12084.

Gu, J. & Liu, J. (2008), 'Bayesian biclustering of gene expression data', *BMC Genomics* **9**, S4.

Gupta, A. & Nagar, D. (1999), *Matrix variate distributions*, CRC Press, Boca Raton, FL.

Hartigan, J. A. (1972), 'Direct clustering of a data matrix', *J. Amer. Statis. Assoc.* **6**, 123–129.

Hastie, T., Tibshirani, R. & Friedman, J. (2009), *The Elements of Statistical Learning; Data Mining, Inference and Prediction*, Springer Verlag, New York.

Hochreiter, S., Bodenhofer, U., Heusel, M., Mayr, A., Mitterecker, A., Kasim, A., Khamiakova, T., Sanden, S., Lin, D., Talloen, W., Bijnens, L., Gohlmann, H., Shkedy, Z. & Clevert, D. (2010), 'Fabia: factor analysis for bicluster acquisition', *Bioinformatics* **26**(12), 1520–1527.

Kaiser, S., Santamaria, R., Khamiakova, T., Sill, M., Theron, R., Quintales, L. & Leisch, F. (2011), *biclust: BiCluster algorithms.* R package version 1.0.1.
URL: *cran.r-project.org/package=biclust*

Lazzeroni, L. & Owen, A. (2002), 'Plaid models for gene expression data', *Statistica Sinica* **12**, 61–86.

Lee, M., Shen, H., Huang, J. & Marron, J. (2010), 'Biclustering via sparse singular value decomposition', *Biometrics* **66**(4), 1087–1095.

Liu, Y., Hayes, D., Nobel, A. & Marron, J. (2008), 'Statistical significance of clustering for high-dimension, low-sample size data', *Journal of the American Statistical Association* **103(483)**, 1281–1293.

Madeira, S. & Oliveira, A. (2004), 'Biclustering algorithms for biological data analysis: A survey', *IEEE Transactions on Computational Biology and Bioinformatics* **1**(1), 24–45.

Pan, W. & Shen, X. (2007), 'Penalized model-based clustering with application to variable selection', *Journal of Machine Learning Research* **8**, 1145–1164.

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

Rand, W. M. (1971), 'Objective criteria for the evaluation of clustering methods', *Journal of the American Statistical Association* **66**, 846–850.

Shabalin, A., Weigman, V., Perou, C. & Nobel, A. (2009), 'Finding large average submatrices in high dimensional data', *Annals of Applied Statistics* **3(3)**, 985–1012.

Sill, M. & Kaiser, S. (2011), *s4vd: Biclustering via sparse singular value decomposition incorporating stability selection.* R package version 1.0.
URL: *cran.r-project.org/web/packages/s4vd*

Tang, C., Zhang, L., Zhang, A. & Ramanathan, M. (2001), Interrelated two-way clustering: An unsupervised approach for gene expression data analysis, *in* 'Proc. of 2nd IEEE International Symposium on Bioinformatics and Bioengineering, Bethesda'.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *J. Royal. Statist. Soc. B.* **58**, 267–288.

Turner, H., Bailey, T. & Krzanowski, W. (2005), 'Improved biclustering of microarray data demonstrated through systematic performance tests', *Computational Statistics and Data Analysis* **48**, 235–254.

Wang, S. & Zhu, J. (2008), 'Variable selection for model-based high-dimensional clustering and its application to microarray data', *Biometrics* **64**, 440–448.

Witten, D. & Tibshirani, R. (2009), 'Covariance-regularized regression and classification for high-dimensional problems', *J. Royal. Stat. Soc. B.* **71(3)**, 615–636, PMCID:PMC2806603.

Witten, D. & Tibshirani, R. (2010), 'A framework for feature selection in clustering', *Journal of the American Statistical Association* **105(490)**, 713–726.

Witten, D., Tibshirani, R. & Hastie, T. (2009), 'A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis', *Biostatistics* **10(3)**, 515–534.

Xie, B., Pan, W. & Shen, X. (2008), 'Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables', *Electronic Journal of Statistics* **2**, 168–212.

Zha, H., He, X., Ding, G., Simon, H. & Gu, M. (2001), 'Spectral relaxation for k-means clustering', *Neural Information Processing Systems* **14**, 1057–1064.

Appendix: Proofs

Proof of Lemma 1

Proof. Let $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ denote the SVD of $\mathbf{X}$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal $n \times n$ and $p \times p$ matrices and $\mathbf{D}$ is a $n \times p$ matrix with decreasing nonnegative diagonal elements. Note that any $n \times K$ orthogonal matrix $\mathbf{A}$ can be written as $\mathbf{A} = \mathbf{U}\boldsymbol{\alpha}$ for some orthogonal $n \times K$ matrix $\boldsymbol{\alpha}$, and any orthogonal $p \times K$ matrix $\mathbf{B}$ can be written as $\mathbf{B} = \mathbf{V}\boldsymbol{\beta}$ for some orthogonal $p \times K$ matrix $\boldsymbol{\beta}$. Thus, instead of solving (4), we can solve

$$\underset{\boldsymbol{\alpha}^T\boldsymbol{\alpha}=\mathbf{I}_K,\boldsymbol{\beta}^T\boldsymbol{\beta}=\mathbf{I}_K}{\text{maximize}} ||\boldsymbol{\alpha}^T\mathbf{D}\boldsymbol{\beta}||_F^2. \tag{16}$$

By inspection, (16) is solved by $\boldsymbol{\alpha} = \mathbf{I}_{n\times K}\mathbf{Q}_1$ and $\boldsymbol{\beta} = \mathbf{I}_{p\times K}\mathbf{Q}_2$, where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are any $K\times K$ orthogonal matrix, and where $\mathbf{I}_{n\times K}$ and $\mathbf{I}_{p\times K}$ and $n\times K$ are $p\times K$ identity matrices. Therefore, the solution to (4) takes the form $\mathbf{A} = \mathbf{U}\mathbf{I}_{n\times K}\mathbf{Q}_1 = \mathbf{U}_{1:K}\mathbf{Q}_1$, and $\mathbf{B} = \mathbf{V}\mathbf{I}_{p\times K}\mathbf{Q}_2 = \mathbf{V}_{1:K}\mathbf{Q}_2$.

□

Proof of Lemma 2

Proof. We must minimize the quantity

$$\mathrm{tr}(\boldsymbol{\Sigma}_k^{-1}(\mathbf{X}_{k,r} - \mu_{kr})\boldsymbol{\Delta}_r^{-1}(\mathbf{X}_{k,r} - \mu_{kr})^T) + 2\lambda|\mu_{kr}|$$

with respect to μ_{kr}. This amounts to minimizing

$$\mu_{kr}^2\mathrm{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{1}\boldsymbol{\Delta}_r^{-1}\mathbf{1}^T) - 2\mu_{kr}\mathrm{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{1}\boldsymbol{\Delta}_r^{-1}\mathbf{X}_{k,r}^T) + 2\lambda|\mu_{kr}|,$$

where $\mathbf{1}$ is a $|C_k| \times |D_r|$ matrix with all entries equal to 1. Completing the square, we see that this is equivalent to minimizing

$$\left(\mu_{kr}\sqrt{\mathrm{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{1}\boldsymbol{\Delta}_r^{-1}\mathbf{1}^T)} - \frac{\mathrm{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{1}\boldsymbol{\Delta}_r^{-1}\mathbf{X}_{k,r}^T)}{\sqrt{\mathrm{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{1}\boldsymbol{\Delta}_r^{-1}\mathbf{1}^T)}}\right)^2 + 2\lambda|\mu_{kr}|$$

with respect to μ_{kr}. The result follows directly.

□

Proof of Theorem 4.1

Before we prove Theorem 4.1, we present a simple lemma.

Lemma 4. *Let $\bar{X}$ denote the mean of the elements in $\mathbf{X}$. Then,*

$$\sum_{i=1}^{n}\sum_{j=1}^{p}(X_{ij} - \bar{X})^2 = \sum_{i=1}^{n}\sum_{j=1}^{p}X_{ij}^2 - np(\bar{X})^2 = \frac{1}{2np}\sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{i'=1}^{n}\sum_{j'=1}^{p}(X_{ij} - X_{i'j'})^2. \tag{17}$$

Now we proceed with a proof of Theorem 4.1.

Proof. Problem (4) is equivalent to the problem

$$\underset{\mathbf{A}^T\mathbf{A}=\mathbf{I}_K,\mathbf{B}^T\mathbf{B}=\mathbf{I}_K}{\text{minimize}} \left\{||\mathbf{X}||_F^2 - ||\mathbf{A}^T\mathbf{X}\mathbf{B}||_F^2\right\}, \tag{18}$$

which is equivalent to

$$\underset{\mathbf{A}^T\mathbf{A}=\mathbf{I}_K,\mathbf{B}^T\mathbf{B}=\mathbf{I}_K}{\text{minimize}} \left\{\sum_{i=1}^{n}\sum_{j=1}^{p}X_{ij}^2 - \sum_{k=1}^{K}\sum_{r=1}^{K}(\sum_{i=1}^{n}\sum_{j=1}^{p}A_{ik}X_{ij}B_{jr})^2\right\}. \tag{19}$$

Since (4) constrains $\mathbf{A}$ to be orthogonal, the two additional constraints in the theorem statement imply that the kth column of $\mathbf{A}$ contains exactly n_k elements that are equal to $\frac{1}{\sqrt{n_k}}$, and $n - n_k$ elements that equal zero. Moreover, the non-zero elements of each column of $\mathbf{A}$ are non-overlapping. A similar claim holds for $\mathbf{B}$. Let C_k denote the indices of the non-zero elements in the kth column of $\mathbf{A}$, and similarly let D_r denote the indices of the non-zero elements in the rth column of $\mathbf{B}$. Then (19) leads to

$$\underset{C_1,\ldots,C_K,D_1,\ldots,D_K}{\text{minimize}} \left\{\sum_{k=1}^{K}\sum_{r=1}^{K}\left(\sum_{i\in C_k}\sum_{j\in D_r}X_{ij}^2 - n_kp_r(\frac{1}{n_kp_r}\sum_{i\in C_k}\sum_{j\in D_r}X_{ij})^2\right)\right\}. \tag{20}$$

Finally, applying Lemma 4 reveals that this is equivalent to

$$\underset{C_1,\ldots,C_K,D_1,\ldots,D_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \sum_{r=1}^{K} \sum_{i \in C_k} \sum_{j \in D_r} (X_{ij} - \bar{X}_{kr})^2 \right\}. \tag{21}$$

Now one can easily show that this is equivalent to the biclustering optimization problem in equation 1 in the case that $K = R$. $\square$

Table 4: Results of various competitors in Simulation 2 with $n = 200$. We report the mean (and standard error) over 50 simulated data sets of the CER of the rows and columns, proportion of correctly identified zeros and non-zeros, sparsity rate, and sparsity error rate. Note that $\bar{\lambda}$ is the mean of λ selected across 50 simulations using the approach of Section 5.2.

p	Method	Row CER	Column CER	C. Zeros	C. Non-zeros	Sparsity Rate	Sparsity Error Rate
	k-means	0.0460 (0.009)	0.0725 (0.008)	-	-	-	-
	Bicluster λ=0	0.0306 (0.008)	0.0434 (0.007)	-	-	-	-
	Bicluster λ=200	0.0289 (0.007)	0.0425 (0.007)	0.264 (0.035)	0.994 (0.002)	0.135 (0.018)	0.372 (0.021)
	Bicluster λ=500	0.0313 (0.008)	0.0482 (0.007)	0.574 (0.053)	0.985 (0.004)	0.295 (0.028)	0.217 (0.025)
200	Bicluster λ=1000	0.0552 (0.010)	0.0723 (0.009)	0.749 (0.042)	0.962 (0.007)	0.392 (0.238)	0.142 (0.022)
	Bicluster $\bar{\lambda}$=475	0.0292 (0.007)	0.0456 (0.007)	0.684 (0.053)	0.987 (0.002)	0.345 (0.028)	0.166 (0.026)
	IP	-	-	1.000 (0.000)	0.000 (0.000)	1.000 (0.000)	0.498 (0.020)
	SSVD rank-2	-	-	0.683 (0.047)	0.489 (0.052)	0.609 (0.048)	0.388 (0.017)
	LAS	-	-	0.366 (0.008)	0.932 (0.004)	0.217 (0.007)	0.353 (0.012)
	k-means	0.0168 (0.005)	0.0494 (0.007)	-	-	-	-
	Bicluster λ=0	0.0100 (0.004)	0.0375 (0.006)	-	-	-	-
	Bicluster λ=200	0.0097 (0.004)	0.0374 (0.006)	0.127 (0.028)	0.998 (0.001)	0.063 (0.013)	0.440 (0.021)
	Bicluster λ=500	0.0103 (0.004)	0.0379 (0.006)	0.287 (0.045)	0.995 (0.001)	0.151 (0.025)	0.354 (0.024)
500	Bicluster λ=1000	0.0112 (0.004)	0.0401 (0.007)	0.511 (0.058)	0.994 (0.001)	0.261 (0.032)	0.244 (0.028)
	Bicluster $\bar{\lambda}$=663	0.0098 (0.004)	0.0383 (0.006)	0.530 (0.059)	0.994 (0.0013)	0.264 (0.029)	0.242 (0.029)
	IP	-	-	1.000 (0.000)	0.000 (0.000)	1.000 (0.000)	0.498 (0.020)
	SSVD rank-2	-	-	0.594 (0.045)	0.623 (0.043)	0.503 (0.044)	0.373 (0.016)
	LAS	-	-	0.443 (0.011)	0.953 (0.004)	0.244 (0.008)	0.305 (0.013)

Table 5: Results for Simulation 3, averaged over 100 simulated data sets. For sparse biclustering, K and R were automatically chosen using Algorithm 2. Note that $\bar{\lambda}$ is the mean of λ selected across 100 simulations using the approach of Section 5.2. Standard errors are in parentheses.

Method	Sparsity Rate	C. Zeros	C. Non-zeros	Sparsity Error Rate
Bicluster λ=0	0.000 (0.000)	0.000 (0.000)	1.000 (0.000)	0.920 (0.000)
Bicluster λ=80	0.829 (0.012)	0.895 (0.013)	0.940 (0.005)	0.101 (0.012)
Bicluster λ=90	0.872 (0.009)	0.944 (0.010)	0.951 (0.005)	0.056 (0.009)
Bicluster λ=100	0.878 (0.014)	0.950 (0.015)	0.955 (0.005)	0.050 (0.013)
Bicluster λ=110	0.804 (0.024)	0.871 (0.025)	0.963 (0.004)	0.122 (0.023)
Bicluster $\bar{\lambda} = 11.6$	0.310 (0.029)	0.336 (0.032)	0.986 (0.004)	0.612 (0.029)
SSVD	0.886 (0.002)	0.963 (0.002)	0.997 (0.001)	0.034 (0.002)
IP	0.972 (0.001)	0.997 (0.001)	0.307 (0.008)	0.059 (0.001)
LAS	0.920 (0.002)	0.963 (0.002)	0.575 (0.009)	0.068 (0.002)

Table 6: Results for Simulation 4. Details are as in Table 5.

Method	Sparsity Rate	C. Zeros	C. Non-zeros	Sparsity Error Rate
Bicluster λ=40	0.648 (0.020)	0.718 (0.023)	0.775 (0.007)	0.274 (0.019)
Bicluster λ=60	0.770 (0.018)	0.849 (0.021)	0.706 (0.007)	0.171 (0.017)
Bicluster λ=80	0.813 (0.016)	0.895 (0.017)	0.679 (0.007)	0.136 (0.015)
Bicluster λ=100	0.859 (0.012)	0.950 (0.014)	0.687 (0.004)	0.088 (0.011)
Bicluster λ=120	0.823 (0.009)	0.915 (0.010)	0.727 (0.006)	0.112 (0.009)
Bicluster $\bar{\lambda} = 12.2$	0.262 (0.021)	0.294 (0.024)	0.928 (0.006)	0.616 (0.020)
SSVD	0.792 (0.008)	0.897 (0.006)	0.834 (0.028)	0.112 (0.004)
IP	0.944 (0.012)	0.995 (0.001)	0.358 (0.007)	0.097 (0.001)
LAS	0.877 (0.002)	0.963 (0.002)	0.634 (0.005)	0.084 (0.002)

Table 7: Results for simulation study with $n = p = 200$ as described in Section 8.3. Sparse biclustering and MVN biclustering were performed, with various values of λ, and with λ chosen automatically ($\bar{\lambda}$). MVN biclustering was performed with $\mathbf{\Sigma}^{-1}$ and $\mathbf{\Delta}^{-1}$ known (MVN bicluster known) and unknown (MVN bicluster).

Method	Row CER	Column CER	C. Zeros	C. Non-zeros	Sparsity Rate	Sparsity Error Rate
k-means	0.124 (0.013)	0.145 (0.008)	-	-	-	-
Bicluster $\lambda = 0$	0.075 (0.013)	0.081 (0.010)	-	-	-	-
Bicluster $\lambda = 200$	0.068 (0.012)	0.078 (0.009)	0.556 (0.031)	0.978 (0.003)	0.272 (0.014)	0.248 (0.023)
Bicluster $\lambda = 400$	0.065 (0.012)	0.079 (0.009)	0.782 (0.029)	0.960 (0.006)	0.394 (0.015)	0.139 (0.020)
Bicluster $\bar{\lambda} = 430$	0.066 (0.012)	0.078 (0.009)	0.791 (0.033)	0.962 (0.007)	0.398 (0.019)	0.137 (0.023)
MVN bicluster $\lambda = 0$	0.071 (0.013)	0.081 (0.010)	-	-	-	-
MVN bicluster $\lambda = 15$	0.060 (0.012)	0.073 (0.009)	0.649 (0.028)	0.975 (0.005)	0.323 (0.013)	0.199 (0.020)
MVN bicluster $\lambda = 30$	0.087 (0.014)	0.095 (0.011)	0.809 (0.025)	0.922 (0.013)	0.432 (0.015)	0.141 (0.018)
MVN bicluster $\bar{\lambda} = 18.8$	0.060 (0.012)	0.073 (0.010)	0.716 (0.039)	0.969 (0.009)	0.354 (0.019)	0.169 (0.025)
MVN bicluster known, $\lambda = 0$	0.027 (0.008)	0.044 (0.007)	-	-	-	-
MVN bicluster known, $\lambda = 100$	0.025 (0.008)	0.041 (0.007)	0.475 (0.027)	0.997 (0.001)	0.245 (0.018)	0.258 (0.016)
MVN bicluster known, $\lambda = 250$	0.034 (0.008)	0.053 (0.009)	0.693 (0.027)	0.987 (0.006)	0.358 (0.020)	0.155 (0.014)
MVN bicluster known, $\bar{\lambda} = 257.5$	0.057 (0.017)	0.048 (0.009)	0.712 (0.039)	0.993 (0.002)	0.344 (0.020)	0.163 (0.026)
IP	-	-	1.000 (0.000)	0.000 (0.000)	1.000 (0.000)	0.500 (0.020)
SSVD rank-2	-	-	0.716 (0.040)	0.449 (0.051)	0.640 (0.044)	0.387 (0.014)
LAS	-	-	0.334 (0.006)	0.917 (0.004)	0.208 (0.005)	0.376 (0.012)