Robust integrative biclustering for multi-view data

Weijie Zhang¹, Christine Wendt², Russel Bowler³, Craig P Hersh⁴, and Sandra E Safo¹

Abstract

In many biomedical research, multiple views of data (e.g. genomics, proteomics) are available, and a particular interest might be the detection of sample subgroups characterized by specific groups of variables. Biclustering methods are well-suited for this problem as they assume that specific groups of variables might be relevant only to specific groups of samples. Many biclustering methods exist for detecting row–column clusters in a view but few methods exist for data from multiple views. The few existing algorithms are heavily dependent on regularization parameters for getting row–column clusters, and they impose unnecessary burden on users thus limiting their use in practice. We extend an existing biclustering method based on sparse singular value decomposition for single-view data to data from multiple views. Our method, integrative sparse singular value decomposition (iSSVD), incorporates stability selection to control Type I error rates, estimates the probability of samples and variables to belong to a bicluster, finds stable biclusters, and results in interpretable row–column associations. Simulations and real data analyses show that integrative sparse singular value decomposition outperforms several other single- and multi-view biclustering methods and is able to detect meaningful biclusters. iSSVD is a user-friendly, computationally efficient algorithm that will be useful in many disease subtyping applications.

Keywords

Multi-view biclustering, biclustering, stability selection, multiomics, co-clustering, integrative biclustering

1 Introduction

Biclustering (or two-way clustering, co-clustering, two-mode clustering) is a popular statistical method for simultaneously detecting groups of samples (rows) and groups of variables (columns) characterizing the different sample groups. These clusters of rows and columns are known as biclusters. Biclustering methods are especially appealing for complex disease subtyping as they seek to detect homogeneous subgroups of people characterized by highly specific groups of biological features. A main limitation of one-way clustering algorithms such as hierarchical or k-means clustering when applied to high-dimensional molecular data for disease subtyping is that cluster assignment of samples is based on the assumption that all molecular features are relevant to the sample groups or disease subtypes. But specific groups of genes, for instance, may be co-regulated within one disease subtype, and not another subtype. In such cases, biclustering methods are well-suited.

Generally speaking, a biclustering algorithm finds the associations between observations (rows) and attributes (columns) in a data matrix. More recently, because of the availability of three-dimensional (3D) data such as gene-sample-time data in biomedical research, a number of tri-clustering methods have been proposed to detect homogeneous 3D subspaces in a given 3D data set. As noted by Henriques and Madeira,¹ tri-clustering algorithms face a number of major challenges such as robustness and efficiency. In this paper, we focus on biclustering, which aims to detect row-column associations in a two-dimensional (2D) data matrix.

¹Division of Biostatistics, University of Minnesota, MN, USA
²Division of Pulmonary, Allergy and Critical Care, University of Minnesota, MN, USA
³Division of Pulmonary, Critical Care and Sleep Medicine, Department of Medicine, National Jewish Health, Denver, USA
⁴Channing Division of Network Medicine, Brigham and Women’s Hospital, Harvard Medical School, USA

Corresponding author:
Sandra E Safo, Division of Biostatistics, University of Minnesota, Minneapolis, MN 55455, USA.
Email: ssafo@umn.edu
A number of biclustering methods have been proposed over the past two decades and they can be broadly categorized into four groups: (a) combinatorial methods such as CTWC,2 OPSM,3 BIMAX,4 association analysis-based RAP,5 COALESCE,6 QUBIC,7 and QUBIC28; (b) probabilistic and generative methods such as SAMBA,9 FABIA,10 BicMix,11 COBRA,12 GBC,13 and plaid models14; (c) matrix factorization approaches that include SSVD,15 S4VD,16 biclustering via non-negative matrix factorization,17 and BEM18; and (d) deep learning with neural networks such as AutoDecoder.19 Other biclustering methods have been proposed to tackle specific problems such as missing data,20 heterogeneous and temporal medical data,21 discrete or binary data,22 and heterogeneous data from multiple views.23

So far, most existing biclustering methods aim to detect row–column clusters in a single data matrix (i.e. single-view or data from one view). Due to technological advancements, for the same sets of samples, multiple types of data or views (e.g. genomics, proteomics) are oftentimes generated in biomedical research. There is continued interest in statistical methods and/or applications that integrate these diverse but related data in a meaningful way to exploit the strengths in each data as well as the overall dependency structure among the multiple views. Such methods and/or applications have the potential to better model the complexity in biomedical data and to yield greater insight into disease pathobiology than individual analysis of the views. Biclustering methods that leverage the rich information in these multi-faceted data have the potential to detect multidimensional view-specific features characterizing sample subgroups common to all views. Figure 1 is a pictorial illustration of biclustering for two views. The two data matrices in the figure describe two distinct feature sets from the same samples. For implementing biclustering approaches on multi-view data, we consider an integrative bicluster to be a subset of samples and their associated features across multiple views. We then can detect underlying sample groups and the different features that describe them. For instance, by integrating genetic (View 1) and clinical data (View 2), we leverage the strengths in molecular data and the advantages of clinical data to define homogeneous groups of people characterized by common molecular features and clinical factors. Yet, there are only few biclustering methods that have been developed for data from multiple views.24,25,23 For data from two views, existing tri-clustering algorithms could be used to detect homogeneous subspaces. Bunte et al.25 developed a Bayesian approach for joint biclustering of data from multiple views that is based on group factor analysis. In Sun et al.,23 a multi-view biclustering method based on sparse singular decomposition15 was proposed. The authors in Sun et al.24 proposed a multi-view biclustering method that is based on low-rank matrix approximation and a proximal alternating linearized minimization algorithm was developed to solve their optimization problem. Recently, a Bayesian biclustering method for integrating data from multiple views that allow for each data to have a different distribution has been proposed.13

The sparse singular value decomposition (SSVD) biclustering method for single-view data15 obtains a sparse rank-one approximation of the data. To obtain the first bicluster, the authors minimize the Frobenius norm between the data and the rank one approximation of the data while regularizing both the left and right singular vectors using adaptive lasso penalties.26 The degree of sparsity of the singular vectors depends strongly on the choice of the regularization parameters. Several techniques including Bayesian information criterion (BIC), Akaike Information criterion (AIC), and cross-
validation have been proposed in the literature to select regularization parameters. The BIC was used to choose the optimal tuning parameters in Lee et al.\textsuperscript{15} However, the authors in Sill et al.\textsuperscript{16} observed that the BIC oftentimes resulted in low degree of sparsity and they proposed to use stability selection\textsuperscript{27} to choose the regularization parameters and to determine stable biclusters. Stability selection is a subsampling procedure that was originally proposed to select stable variables for penalized regression models and since then it has been used successfully in other settings. This approach allows for choosing penalization parameters and further controlling Type I error rates.

In this article, we extend the SSVD method\textsuperscript{15} and the SSVD method with stability selection\textsuperscript{16} to detect biclusters in multi-view data. Our main contribution, compared to other multi-view biclustering methods\textsuperscript{24,23,13} is that we use stability selection to detect stable and robust subject and variable clusters. We note two other contributions to existing biclustering methods that we believe are important. Compared to the methods in Sill et al.,\textsuperscript{16} Sun et al.,\textsuperscript{23} and Sun et al.,\textsuperscript{24} we estimate subsequent biclusters using the whole data. To ensure that each sample belongs to only one subject cluster, the aforementioned methods use only unclustered samples when deriving subsequent sample and column clusters. This is concerning since smaller sample sizes are used to estimate subsequent biclusters. We track samples clustered and assign weights that ensure that clustered samples have zero coefficients in subsequent biclusters. Also, we develop efficient and user-friendly algorithms for the proposed method. We use extensive simulations to compare the performance of our proposed method in both biclusters detection and computational time. We apply the method to RNA sequencing and proteomics data from the Genetic Epidemiology of Chronic Obstructive Pulmonary Disease Study (COPDGene)\textsuperscript{28} to detect subject and molecular clusters. The sample groups identified are compared across several demographic, clinical, and lung function variables.

The rest of the paper is organized as follows. In Section 2, we introduce the proposed method. In Section 3, we conduct simulation studies to assess the performance of our method in comparison with other methods in the literature. In Section 4, we apply our proposed method to a real data. We end with some brief discussion in Section 5.

2 Methods

Our primary goal is to define a biclustering method that leverages the wealth of information from diverse but related data to detect stable and robust sample clusters characterized by cluster-specific variables. In this section, we first briefly summarize previous biclustering methods for one view that are based on singular value decomposition (SVD) which motivates our proposed work. Then, we extend previous work to multiple views. Finally, we incorporate the concept of stability selection to detect robust biclusters and control Type I error rates of falsely selecting samples and variables in a bicluster.

2.1 Biclustering for a single view

Let $X$ be a $n \times p$ data matrix where $n$ represents samples and $p$ represents variables. We assume that the data can be approximated by sparse rank $k$ ($k = 1, \ldots, K$) left and right singular vectors $u_k$ and $v_k$ and corresponding singular value $s_k$, that is, $X \approx \sum_{k=1}^{K} s_k u_k v_k^T$. Then, the non-zero entries in $u_k$ represent a sample subgroup in the $k$th left singular vector. Similarly, the non-zero entries in $v_k$ represent a variable subgroup in the $k$th right singular vector. Together, these represent the $k$th bicluster. Lee et al.\textsuperscript{15} proposed to obtain the best sparse rank one approximation of the data by solving the following optimization problem:

$$
(\hat{s}_1, \hat{u}_1, \hat{v}_1) = \min_{s_1, u_1, v_1} \|X - s_1 u_1 v_1^T\|_F^2 + \lambda_{u_1} \mathcal{P}_u(s_1 u_1) + \lambda_{v_1} \mathcal{P}_v(s_1 v_1)
$$

subject to $\|u_1\|_2 = 1$, $\|v_1\|_2 = 1$. 

(1)

Here, the first term measures the approximation error using Frobenius norm of the difference; $\mathcal{P}_u(\cdot)$, $\mathcal{P}_v(\cdot)$ are sparsity-inducing terms, and $\lambda_{u_1}$ and $\lambda_{v_1}$ control the level of sparsity. For fixed $u_1$, the authors minimize (1) with respect to $\hat{v}_1 = s_1 v_1$. Similarly, for fixed $v_1$ they minimize (1) with respect to $\hat{u}_1 = s_1 u_1$. Adaptive lasso penalties\textsuperscript{26} were used to induce sparsity on the vectors, that is,

$$
\mathcal{P}_u(s_1 u_1) = s_1 \sum_{i=1}^{n} w_{1,i} |u_{1,i}| \quad \text{and} \quad \mathcal{P}_v(s_1 v_1) = s_1 \sum_{j=1}^{p} w_{2,j} |v_{1,j}|,
$$

(2)

where $w_{1,i}$’s and $w_{2,j}$’s are adaptive weights (obtained from the data itself). The adaptive lasso penalties become lasso penalties when $w_{1,i} = w_{2,j} = 1$, $\forall i, j$. To obtain $u_1$ and $v_1$, Lee et al.\textsuperscript{15} iteratively solved the optimization problem (1) until convergence. The estimated singular value is then $\hat{s}_1 = \hat{u}_1 X \hat{v}_1$. The reconstructed matrix, after convergence, given by $\hat{s}_1 \hat{u}_1 \hat{v}_1^T$ is one sparse SVD layer. The non-zero entries in $\hat{u}_1$ and $\hat{v}_1$ form the first bicluster. Subsequent biclusters may
be obtained by a sparse rank one approximation of the deflated data \((X - \tilde{s}_1 \tilde{u}_1 \tilde{v}_1^T)\), that is, by repeatedly solving problem (1) using deflated data.

### 2.2 Biclustering for multiple views

We extend the biclustering problem to data from multiple views. Suppose that data are available from \(D\) different views, and each view is arranged in an \(n \times p^{(d)}\) matrix \(X^{(d)}\), where the superscript \(d\) correspond to the \(d\)th view. For instance, for the same set of \(n\) samples, matrix \(X^{(1)}\) consists of RNA sequencing data and \(X^{(2)}\) consists of proteomics data, for \(D = 2\) views. We wish to cluster the singular vectors. For each view, we posit unobserved common views. We extend the biclustering problem to data from multiple views. Suppose that data are available from \(D\) different \(n \times d\) views, that is, \(X^{(d)} \approx \sum_{k=1}^{K} s_k^{(d)} u_k \tilde{v}_k^{(d)T}\). Here, \(U = [u_1, \ldots, u_K]\) is an \(n \times K\) matrix of common latent components that connects the \(D\) views and induces dependencies across the views. Following problem (1), the best sparse rank one approximation for the \(D\) views may be obtained by solving the optimization problem:

\[
\begin{align*}
\min_{u_1, v_1^{(d)}, d = 1, 2, \ldots, D} & \quad \sum_{d=1}^{D} \|X^{(d)} - s_1^{(d)} u_1 v_1^{(d)T}\|_F^2 + \lambda_{u_1} \|s_1 u_1\|_2 + \lambda_{v_1} \|v_1^{(d)}\|_2 \\
\text{subject to} & \quad \|u_1\|_2 = 1, \quad \|v_1^{(d)}\|_2 = 1.
\end{align*}
\]

(3)

In order to obtain sparse approximations, we use lasso penalties\(^{30}\) on both \(u_1\) and \(v_1^{(d)}\) for \(d \in \{1, 2, \ldots, D\}\). That is

\[
\begin{align*}
\mathcal{P}_u(s_1 u_1) = \lambda_{u_1} \sum_{i=1}^{n} s_1 |u_{1i}| \quad \text{and} \quad \mathcal{P}_v^{(d)}(s_1^{(d)} v_1^{(d)}) = \sum_{j=1}^{p^{(d)}} \lambda_{v_1^{(d)}} |v_{1j}^{(d)}|,
\end{align*}
\]

(4)

where \(\lambda_{u_1}\) and \(\lambda_{v_1^{(d)}}\) are regularization parameters. The penalties are sums of absolute values of the elements in the first singular vectors. For fixed \(u_1\), minimizing (3) is equivalent to minimizing \(D\) equations with similar forms, with respect to \(\tilde{v}_1^{(d)} = s_1^{(d)} v_1^{(d)}\):

\[
\begin{align*}
\tilde{v}_1^{(d)} = \min_{v_1^{(d)}} & \quad \|X^{(d)} - u_1 v_1^{(d)T}\|_F^2 + \lambda_{v_1^{(d)}} \sum_{j=1}^{p^{(d)}} |v_{1j}^{(d)}| \\
& \quad \text{subject to} \quad \|v_{1j}^{(d)}\|_2 = 1.
\end{align*}
\]

(5)

Then, \(\tilde{v}_1^{(d)} = \tilde{v}_1^{(1)} + \tilde{v}_1^{(2)} + \ldots + \tilde{v}_1^{(D)}\) is an estimate for the product of the first right singular vector and the first singular value. As in Lee et al.,\(^{15}\) we obtain an estimated sparse rank one right singular vector for the \(d\)th view as \(\tilde{v}_1^{(d)} = \tilde{v}_1^{(d)} / \|\tilde{v}_1^{(d)}\|_2\). The rank one sparse estimate for the left singular vector \(u_1\) can be obtained in a similar way by concatenating the \(D\) views. Let \(X = [X^{(1)}, X^{(2)}, \ldots, X^{(D)}] \in \mathbb{R}^{n \times p^{(1)} + p^{(2)} + \ldots + p^{(D)}}\) be the concatenated data. Also, let \(v_1\) be a collection of the rank one right singular vectors for all \(D\) views, that is, \(v_1 = (v_1^{(1)T}, \ldots, v_1^{(D)T})^T \in \mathbb{R}^{(p^{(1)} + p^{(2)} + \ldots + p^{(D)})}.\) For \(v_1\) fixed, we solve optimization problem (3) for \(u_1\):

\[
\begin{align*}
\tilde{u}_1 = \min_{u_1} & \quad \|X - \tilde{u}_1 v_1^{(d)T}\|_F^2 + \lambda_{u_1} \sum_{i=1}^{n} |\tilde{u}_{1i}| \\
& \quad \text{subject to} \quad \|\tilde{u}_{1i}\|_2 = 1.
\end{align*}
\]

(7)

where \(\tilde{u}_1 = s_1 u_1\) and \(X\) is the concatenated data set. Using soft-thresholding again, the component-wise solution of (7) is

\[
\tilde{u}_{1i} = \text{sign}[(X v_1)_i]((X v_1)_i - \lambda_{u_1}/2)_+.
\]

(8)
As before, the corresponding sparse rank one left singular vector is \( \widehat{u}_1 = \frac{\widehat{u}_1}{\| \widehat{u}_1 \|_2} \). Subsequent update of \( \widehat{d}_1 \) is given by \( \widehat{d}_{1} = \widehat{u}_1^T X_{1}(\cdot) \widehat{v}_{1}^{(d)} \). Then we deflate the \( d \)th data as \( X_{1}(\cdot) - \widehat{d}_{1} \widehat{u}_1 \widehat{v}_{1}^{(d)} \) for \( d \in \{1, 2, \ldots, D\} \). For subsequent biclusters, we repeatedly solve problems (5) and (7) using deflated data, that is, we find sparse rank one approximations of the deflated data.

The regularization parameters, \( \lambda_{u_1} \) and \( \lambda_{v_1}^{(d)} \), control the degree of sparsity (i.e., the number of non-zero elements in \( u_1 \) and \( v_1^{(d)} \)), and must be chosen. Lee et al.\(^{15} \) proposed a BIC\(^{31} \) to obtain the optimal regularization parameters. Sill et al.\(^{16} \) proposed to use stability selection techniques for robust biclusters. Similar to Sill et al.,\(^{16} \) we choose the regularization parameters using stability selection.\(^{27} \)

### 2.3 Multi-view biclustering with stability selection

Stability selection method, proposed by Meinshausen and Bühlmann,\(^{27} \) has been used for variable selection problems such as regularized regression and even for sparse SVD.\(^{14} \) Stability selection essentially combines resampling with variable selection so that the probability that a variable is selected is based on its relative frequency. Meinshausen and Bühlmann\(^{27} \) provide a theoretical justification to show that by selecting variables based on the maximum of these probabilities, we control for Type I error rates of falsely selecting variables. For completeness sake, we briefly summarize the stability selection method and how we use it in our application.

We consider estimating the left singular vector \( u_1 \) and inferring the non-zero coefficients or detecting samples that form a sample cluster. We subsample variables in each view \( I \) times without replacement, while ensuring that each view contains the same set of samples. For each regularization parameter \( \lambda_{u_1} \), we solve the optimization problem (7) for each subsampled data set. Each \( \lambda_{u_1} \in \Lambda_{u_1} \) leads to a different set of non-zero coefficients. We estimate the selection probability for each sample \( i = 1, \ldots, n \) as the number of times sample \( i \) is selected from \( I \) applications of equation (7) for a fixed \( \lambda_{u_1} \). Denote the selection probability corresponding to \( \lambda_{u_1} \) for sample \( i \) as \( \widehat{\Pi}_{u_1}^{(i)} \). Then for an arbitrary threshold, \( \pi_{\text{thr}} \), the stable path for \( u_1, \widehat{S}_{u_1}^{\text{stable}} \), (the set of stable samples) is the set of non-zero coefficients with selection probabilities at least \( \pi_{\text{thr}} \). Essentially, samples with high selection probability are kept, and those with low selection probabilities are disregarded. Then, given the union of the selected samples from all \( \lambda_{u_1} \in \Lambda_{u_1} \), we can estimate the average number of selected samples (i.e., non-zero coefficients) for the regularization region \( \Lambda_{u_1} \), denoted as \( q_{\Lambda_{u_1}} \). From Theorem 1 in Meinshausen and Bühlmann,\(^{27} \) the expected number of falsely selected samples, \( E(u_1) \) with stability selection is bounded by:

\[
E(u_1) \leq \frac{1}{2\pi_{\text{thr}} - 1} \frac{q_{\Lambda_{u_1}}^2}{n}.
\]  

Thus, by reducing the average number of selected samples (i.e. \( q_{\Lambda_{u_1}} \)) or by increasing the threshold \( \pi_{\text{thr}} \), we reduce the expected number of falsely selected samples or the per-family error rate, or if we divide by \( n \), the per-comparison error rate (PCER).\(^{33} \) It is noted by the authors that the threshold value range \( \pi_{\text{thr}} = [0.6, 0.9] \) tend to yield similar results. For \( \pi_{\text{thr}} \) fixed, if we choose the average number of selected samples \( q_{\Lambda_{u_1}} \) to be at most \( e_{\Lambda_{u_1}} = \sqrt{E(u_1)(2\pi_{\text{thr}} - 1)n} \) (and hence the regularization region \( \Lambda_{u_1} \)), we control the family wise error rate for some \( E(u_1) \). Following ideas in Sill et al.,\(^{16} \) we estimate \( \widehat{u}_1 \) with the smallest regularization parameter value in the regularization region that ensures that \( q_{\Lambda_{u_1}} \leq e_{\Lambda_{u_1}} \).

Thus, the component-wise estimate for \( \widehat{u}_1 \) is given by:

\[
\widehat{u}_{1i} = \text{sign}(\langle X v_1 \rangle_1)\langle |(X v_1)_i| - \lambda_{\min_{u_1}}/2 \rangle_+.
\]  

We estimate the right singular vectors \( v_1^{(d)} \) and infer the non-zero coefficients or variables that form a variable cluster in a similar way. Specifically, for each possible \( \lambda_{v_1}^{(d)} \), we draw \( J \) subsamples without replacement and we estimate the selection probabilities for each variable \( j = 1, \ldots, p \) as the number of times variable \( j \) is selected from \( J \) applications. Denote the selection probability corresponding to \( \lambda_{v_1}^{(d)} \) for variable \( j \) as \( \widehat{\Pi}_{v_1}^{(d)} \). Given a threshold \( \pi_{\text{thr}} \) and the desired Type I error value \( E(v_1^{(d)}) \), we obtain the regularization region \( \Lambda_{v_1}^{(d)} \) such that \( q_{\Lambda_{v_1}^{(d)}} \leq e_{\Lambda_{v_1}^{(d)}} \), where \( e_{\Lambda_{v_1}^{(d)}} = \sqrt{E(v_1^{(d)})(2\pi_{\text{thr}} - 1)p^{(d)}} \).

Then the stable set for \( v_1^{(d)}, \left[ \widehat{S}_{v_1}^{\text{stable}} \right] \), is the non-zero coefficients or variables with selection probabilities at least \( \pi_{\text{thr}} \). Given the smallest regularization parameter value in the regularization region, the component-wise estimate for \( \widehat{v}_{1i}^{(d)} \) is given by:

\[
\widehat{v}_{1i}^{(d)} = \text{sign}(\langle X^T u_1 \rangle_1)\langle |(X^T u_1)_i| - \lambda_{\min_{u_1}}/2 \rangle_+.
\]
At convergence, the components of $\hat{V}_1^{(d)}$ become $\hat{V}_1^{(d)} = 1(j \in \hat{S}_{stable}^{(d)}) \hat{v}_j^{(d)}$, where $1(\cdot)$ is an indicator function. Similarly, the components of $\hat{U}_1$ become $\hat{U}_1 = 1(i \in \hat{S}_{stable}^{(u)}) \hat{u}_i^{(u)}$.

The algorithm iterates between $U_1$ and $V_1^{(d)}$, $d = 1, \ldots, D$ until there is convergence. Refer to Algorithm 1 for more details. For convergence, we estimate (1) the relative difference between the objectives ($\sum_{d=1}^{D} \| X^{(d)} - \big( \hat{U}_1 \hat{V}_1^{(d)\top} \big) \|_F^2$) at previous and current iterations, and (2) max ($\| U_1 - \hat{U}_1 \|^2$, min ($\| V_1^{(1)} - \hat{V}_1^{(1)} \|^2$, ..., $\| V_1^{(D)} - \hat{V}_1^{(D)} \|^2$)). The algorithm converges if either (1) or (2) is less than a pre-specified threshold (e.g. 0.0001). Based on our simulation studies and real data analysis, we found an empirical value of 0.0001 provided satisfactory results while keeping the algorithm efficient. If this needs to be tuned, we recommend to set this threshold to balance computational speed and biclustering results. In simulations, the biclustering results from varying thresholds could be compared to the truth and the threshold with better results chosen. In real data analysis where there is no underlying truth, the threshold could be based on clinically or biologically meaningful biclustering results.

**Algorithm 1:** Integrative sparse singular value decomposition algorithm with stability selection (iSSVD).

1. **Input:** $X^{(d)}$, $K$ (optional), Type I errors $E(\hat{u})$ and $E(\hat{v}^{(d)})$, $\pi_{thr}$. $d = 1, \ldots, D$
2. **Output:** $\hat{U} = [\hat{u}_1, \ldots, \hat{u}_K]$, $\hat{V}_1^{(d)} = [\hat{v}_1^{(d)}, \ldots, \hat{v}_K^{(d)}]$, $\hat{v}_j^{(d)} = \{ j \in K \}^{p(d)}$, $\omega_{v_j^{(d)}} = \{ j \in K \}^{p(d)}$
3. **Initialize:** Apply standard SVD to the concatenated data $X$. Let $\{ s_1, u_1, v_1 \}$ be the first SVD triplet. (Note: $v_1 \in \mathbb{R}^{d(1)+\ldots+d(D)}$).
4. **repeat**
5. **solve for $\hat{u}_1$, and hence $\hat{v}_1^{(d)}$**
   (a) Draw subsamples $J$ and estimate $\hat{\Pi}_{\lambda,1}$. Define $\Lambda_{\lambda,1} = \epsilon \Lambda_{\lambda,1}$.
   (b) Estimate the set of non-zero sample coefficients, $\hat{S}_{stable}^{(u)}$.
   (c) Solve for $\hat{u}_1$ using equation (10). Let $s_{u_1} = \| \hat{u}_1 \|_2$, and $\hat{u}_1 = \hat{u}_1 / s_{u_1}$.
   (d) Set $U_1 = \hat{u}_1 / \| \hat{u}_1 \|_2$
6. **end**
7. **for $d = 1, \ldots, D$ do**
8. **solve for $\hat{v}_1^{(d)}$, and hence $\hat{v}_1^{(d)}$**
   (a) Draw subsamples $J$ and estimate $\hat{\Pi}_{\lambda,1}^{(d)}$. Define $\Lambda_{\lambda,1}^{(d)} = \epsilon \Lambda_{\lambda,1}^{(d)}$.
   (b) Estimate the set of non-zero variable coefficients, $\hat{S}_{stable}^{(v)}$.
   (c) Solve for $\hat{v}_1^{(d)}$ using equation (11). Let $s_{v_1} = \| \hat{v}_1^{(d)} \|_2$, $\hat{v}_1^{(d)} = \hat{v}_1^{(d)} / s_{v_1}$.
   (d) Set $V_1^{(d)} = \hat{v}_1^{(d)} / \| \hat{v}_1^{(d)} \|_2$
9. **end**
10. **until convergence**
11. Set $\hat{S}_{stable}^{(v)} = s_{v_1} X^{(d)} \hat{v}_1^{(d)}$; $\hat{s}_{\hat{v}_j^{(d)}} = 1(i \in \hat{S}_{stable}^{(v)}) \hat{v}_{i \hat{v}_j^{(d)}} = 1(i \in \hat{S}_{stable}^{(u)}) \hat{u}_{i \hat{v}_j^{(d)}}$
12. **for subsequent biclusters** repeat steps 3 to 22 using deflated data.
13. **stop** if either $\hat{S}_{stable}^{(v)} = \emptyset$ or $\hat{S}_{stable}^{(u)} = \emptyset$

We note that our method, referred to as integrative sparse singular value decomposition (iSSVD), is different from the one proposed in Liu et al.29 because we are concerned with the problem of simultaneously detecting row (sample) and column (variable) clusters. As such, we regularize both $U_1$ and $V_1^{(d)}$ while Liu et al.29 regularize $V_1^{(d)}$ and apply $k$-means clustering on $u$ after convergence. Our approach allows us to define subgroups in rows and columns of our data simultaneously and makes it appealing to detect sample subgroups characterized by specific groups of variables. Further, compared with several existing methods,29,34,23,24 we use stability selection to detect stable and robust sample and variable clusters, while controlling for Type I error of falsely selecting samples and variables in a bicluster. In addition, to ensure that each sample belongs to only one subject cluster, the authors in Sun et al.23,24 and Sill et al.16 proposed to use only unclustered samples when estimating subsequent biclusters. This is concerning to us since smaller sample sizes are used to estimate subsequent biclusters. Instead, we track samples that are clustered and we assign weights to ensure that those samples have zero coefficients in subsequent biclusters. We do the same for variable clusters if it is desired to have
non-overlapping variable clusters. Of note, concatenating the \( D \) views and applying the biclustering method with stability selection proposed by Sill et al.\(^{16} \) assumes that the regularization parameters are the same for each view, that is, \( \lambda_{vi} = \lambda_{v1}^{(1)} = \cdots = \lambda_{v1}^{(D)} \). This assumption may result in choosing tuning parameters that are either too small or too large for a particular view; this can lead to a solution that is trivial or not sparse, and can inflate Type I error (refer to Supplemental Material for more details).

**Remark 1 Point-wise control:** Searching for a plausible range of tuning parameters is computationally demanding. For computational efficiency, we implemented the point-wise control methods.\(^{27,16} \) Specifically, we considered a smaller searching path that shortens the time to find the optimal \( \lambda_{vi} \) or \( \lambda_{d}^{(i)} \). We adopted the point-wise error control approach implemented in s4vd\(^{16} \) and expanded it to be feasible for multi-view data. For example, if solving for \( \tilde{\lambda}_{1i} \), we can look for a regularization path with a single tuning parameter \( \Lambda_{ni} = \{ \lambda_{ni} \} \) and draw subsamples \( I \) to calculate the average number of selected coefficients \( q_{n_{ni}} \), then we can estimate the selection threshold by

\[
\pi_{thr} = \frac{1}{2} \left( \frac{q_{n_{ni}}^2}{E(u_{ni})m} + 1 \right).
\]

(12)

We define a region for the threshold to be \([ \pi_{min}, \pi_{max} ]\) and we search for a \( \pi_{thr} \) that falls into this range. We start from the median value of the lambda range, and this range will be updated based on the reconstructed threshold. Next calculation uses the median value of the new lambda range and this continues until the reconstructed threshold satisfies the aforementioned range. Thus, instead of calculating the entire stability paths in each iteration, the algorithm finds appropriate parameters with fewer calculations. We incorporated this algorithm into iSSVD and compared the run time with s4vd.

**Remark 2 Choosing the number of biclusters:** To choose the number of biclusters (i.e. \( K \)), we implemented the following approaches: (a) For each view, we calculate the proportions of variation explained by its singular values and select the number of singular values associated with the variation proportion that is larger than a threshold (e.g. larger than 70%); then we set the number of biclusters to be the maximum number of singular values plus one. (b) The user can specify the number of biclusters to be detected beforehand. Then the algorithm will set the maximum number of biclusters to be the smaller number from either (a) or (b). We demonstrate the first criteria using simulation data from Scenario 2, and we report the proportion of the true biclusters detected in Table S2.3 of the Supplemental Material.

### 3 Simulations

We consider two main scenarios to assess the proposed method in detecting biclusters from multi-view data. In both scenarios, we simulate two views \( X^{(1)} \) and \( X^{(2)} \). In Scenario One, we allow for some samples to not belong to any sample cluster. In Scenario Two, each sample belongs to a bicluster. This scenario is especially relevant in disease subtyping where it is important that each sample belongs to only one sample cluster. In each scenario, we generate 50 Monte Carlo data sets for each view. The parameter settings used can be found in Supplemental Material Table S1.5.

#### 3.1 Scenario one

In this scenario, data matrices \( X^{(1)} \in \mathbb{R}^{n \times p^{(1)}} \) and \( X^{(2)} \in \mathbb{R}^{n \times p^{(2)}} \) are generated, respectively, as

\[
X^{(1)} = U S V^{(1)T} + E^{(1)}, \quad X^{(2)} = U S V^{(2)T} + E^{(2)}.
\]

(13)

Hence, the concatenated data is \( X = [X^{(1)}, X^{(2)}] \in \mathbb{R}^{n \times (p^{(1)} + p^{(2)})} \). We set the number of biclusters \( K = 4 \) and the dimensions of data to be \( n = 100 \) and \( p^{(1)} = p^{(2)} = 1, 000 \). Each bicluster has 10 rows and 100 columns. As such, there are only 40 samples that belong to the sample clusters with the remaining 60 samples not belonging to any cluster. Similarly, there are 400 signal variables characterizing the sample clusters, and the remaining 600 variables are noise. The left singular matrix \( U \in \mathbb{R}^{100 \times 100} \) is the common left singular matrix for the two views. Since we design four integrative biclusters, we randomly select 10 rows in each column of the first four columns in matrix \( U \) and assign data values generated from a uniform distribution \( U(0.5, 1) \), while ensuring there is no overlapping samples. The remaining \( n - 40 \) rows in the first four columns are assigned zero values. For the entries of the remaining \( n - K \) columns, we use data generated from the normal distribution with mean 0 and variance 1. We obtain the right singular matrix for each of the two views, \( V^{(d)} \in \mathbb{R}^{1000 \times 100} \), as follows. We randomly select 100 rows in each column of the first 4 columns in matrix \( V^{(d)} \) and fill the corresponding elements from the uniform distribution \( U(0.5, 1) \), while also ensuring there are no overlapping rows (correspondingly no overlapping variables characterizing the sample clusters). As before, we fill out the entries of the remaining columns in \( V^{(d)} \) with data generated from the standard normal distribution. For the singular values \( S \), we set the
first four entries to 27, 20, 18, and 10, respectively, and the rest with small values $c$. Thus, $S \in \mathbb{R}^{100 \times 100} = \text{diag}(27, 20, 18, 10, c, \ldots, e)$ where $e = 0.3$; these singular values were chosen to make the biclusters detectable. Therefore, each view is reconstructed as $X^{(d)} = USV^{(d)T}$, $d = 1, 2$. We then add random noise generated from a normal distribution with mean 0 and variance $\sigma^2$ to each view. We will assess the performance of the method for small to large variances. Since the two views often tend to have different scales in real data analysis, we consider different scalings for $X^{(1)}$, $X^{(2)}$ as done in Liu et al.29

Case 1: In this case, the two views have different scales. Specifically, the concatenated data is of the form $X = \{X^{(1)}, sX^{(2)}\}$ where $s \in \{1, 2, 5, 10\}$; this allows us to investigate the performance of the proposed and existing methods in situations of unbalanced scales. This is commonly the case in multi-view data and it can be challenging for single-view methods.

Case 2: Case 2 is similar to Case 1 but we fix the scalar $s$ and study the performance of a method under different levels of noise. We vary the standard deviation $\sigma \in \{0.1, 0.2, \ldots, 1\}$.

Case 3: In Case 3, we expand the dimensions of data to $n = 500$ and $p^{(1)} = p^{(2)} = 10, 000$. Here, each bicluster has 50 rows and 200 columns. The remaining $n - 200$ rows do not belong to any cluster, and the remaining $p^{(d)} - 800$, $d = 1, 2$, variables are noise. We use this case to study the computational efficiency when $p^{(d)} \gg n$.

3.2 Scenario two

In the first scenario, we allowed for sample overlaps. In this scenario, each sample belongs to a bicluster. As before, we have two views and they have dimensions $n = 200$, $p^{(1)} = p^{(2)} = 1, 000$. There are four integrative biclusters for each view; each sample cluster has 50 samples, each variable cluster has 100 variables and the remaining $p^{(d)} - 400$ variables are noise. The singular matrices are $U \in \mathbb{R}^{n \times K}$, $S \in \mathbb{R}^{K \times K}$ and $V^{(d)} \in \mathbb{R}^{p^{(d)} \times K}$, $d = 1, 2$ and $K = 4$. The 50 entries for each column in $U$ are generated from the uniform distribution $U(0, 1)$, while ensuring that there are no overlapping entries. The remaining 150 entries in $U$ for each $k$ column is set to 0. For the right singular matrix, $V^{(d)}$, $d = 1, 2$, we generate 100 entries for each column from $U(0, 1)$. The remaining $p^{(d)} - 100$ entries in each column are set to zero. For the singular values, we set it to 27, 20, 18, and 10; thus $S = \text{diag}(27, 20, 18, 10)$. The entries in $E^{(1)}$ and $E^{(2)}$ are all generated as i.i.d random samples from $N(0, \sigma^2)$, where $\sigma \in \{0.1, 0.2, \ldots, 1\}$. We consider another simulation setting where the errors are not simulated from the Gaussian distribution (refer to Section 3.2 in the Supplemental Material).

3.3 Competing methods and software

We compared the performance of our method with four biclustering methods for multi-view data and one biclustering method with stability selection developed for data from one view. For the multi-view-based methods, we consider the proximal co-clustering24 [mvProx], the multi-view svd [mvSVD],23 and the generalized biclustering (GBC)13 methods. We perform mvProx and mvSVD using the R-package mvcluster (Version 1.0). This R package includes the mvSVD method as well as two proximal co-clustering methods using $l_1$-norm regularization [mvProxL1] and $l_0$-norm regularization [mvProxL0], respectively. The core functions for mvProxL0, mvProxL1, and mvSVD are written in C++. The algorithms from the R package mvcluster (mvSVD, mvProxL0, and mvProxL1) detect one integrative bicluster at a time. To detect subsequent biclusters for the mvcluster (mvSVD, mvProxL0, and mvProxL1) methods, we follow suggestions by Sun et al.24,23 and we manually subset the data after each run and we delete the samples (rows) that are detected previously. We set the maximum number of biclusters to be detected to 4. We stack the views when applying the biclustering with stability selection [s4vd] method, and we use the R-package s4vd (Version 1.1.1). GBC is a Bayesian biclustering method for detecting biclusters from multiple views that allow each view to have a different probability distribution. Please refer to Section 1 in the Supplemental Material for description of these methods. The simulations have been carried out using the Minnesota Supercomputing Institute Mangi compute cluster. Simulations of mvSVD, mvProxL0, mvProxL1, and s4vd have been implemented with R 4.0.4, and simulations of iSSVD have been implemented with Python 3.7.

3.4 Evaluation criteria

We evaluate the proposed and existing methods based on bicluster similarity measures, F-score, and variable and sample selection. These are widely used measures in the statistical and machine learning literature for assessing biclustering methods.16,19 For similarity measures (i.e. similarity between the algorithm-generated biclusters and true biclusters from the same data), we consider bicluster relevance and recovery, which are defined as follows.
Suppose $M$ is the set of estimated biclusters and $M^*$ is the set of true biclusters, each containing a set of columns $C$ and a set of rows $R$. Let $M = \{M_1, M_2, \ldots, M_m\}$ and $M^* = \{M_1^*, M_2^*, \ldots, M_q^*\}$, where $M_i = R_i \times C_i$ and $M_i^* = R_i^* \times C_i^*$, and $\times$ denotes the Cartesian product of the sets of rows and columns. Then the Jaccard index for two biclusters, each obtained from the Cartesian product is

$$\text{Jac}(M_i, M_i^*) = \frac{M_i \cap M_i^*}{M_i \cup M_i^*}.$$ 

Similarly as in Sill et al., the average relevance and recovery scores are defined as:

$$\text{Relevance} = \frac{1}{m} \sum_{a=1}^{m} \max_{b \in \{1,2,\ldots,q\}} \text{Jac}(M_a, M_b^*)$$

$$\text{Recovery} = \frac{1}{l} \sum_{b=1}^{l} \max_{a \in \{1,2,\ldots,m\}} \text{Jac}(M_b^*, M_a)$$

Essentially, the relevance score shows how well the detected biclusters represent the true ones while the recovery score evaluates to what degree the true biclusters are recovered by the algorithm.

For a combined effect of relevance and recovery measures, we consider the F-score, the harmonic mean of average relevance and recovery:

$$F\text{-score} = \frac{2 \times \text{Relevance} \times \text{Recovery}}{\text{Relevance} + \text{Recovery}}.$$ 

For samples and variables selected, we consider false positives (FP) and false negatives (FN). The FP is defined as the ratio of number of falsely selected non-zero elements outside of true bicluster against number of elements in the true bicluster. Conversely, the FN is defined as the ratio of number of non-zeros computed by the algorithm in the true bicluster against the number of elements in the true bicluster.

### 3.5 Simulation results

#### 3.5.1 Scenario one

**Unbalanced scales.** In the first case of scenario one, we vary the scalar $s$ to be 1, 2, 5, and 10 to evaluate the methods in situations of unbalanced scales. The average recovery and relevance scores are shown in Figure 2 for fixed noise level $\sigma = 0.2$. Please refer to Supplemental Material Table S1.3 for more results. For scalar $s = 1$, that is, both views on the same scale, iSSVD and s4vd had higher average recovery scores, whereas mvSVD and GBC had lower scores and mvProxL0 and mvProxL1 performed worse. The performance of iSSVD is slightly better and more stable than s4vd based on the scores of average recovery, since the median is higher and the variability is lower. From the relevance scores (Figure 2, right panel), s4vd has a higher performance, suggesting that iSSVD assigned more noise to
biclusters. As we increase the scales from 2 to 10, the two views become more and more unbalanced. We observe that the average recovery scores of iSSVD are still higher, suggesting that our method can perform well when data are unbalanced. However, for s4vd and mvSVD, their abilities to detect biclusters decrease dramatically; the performance of GBC increased for $s=2$ and 5 but decreased for $s=10$. Furthermore, mvProxL0 and mvProxL1 perform poorly in all the situations. The average relevance scores of these methods show a similar trend but mvProxL0 tend to be better than mvProxL1. The relevance scores of s4vd are higher in less unbalanced scales ($s=1, 2$) settings. However, when the two views are more unbalanced, the biclusters detected by iSSVD are more representative of the true biclusters. In the unbalanced scales, s4vd mostly is able to detect the biclusters from the more dominant view, which in this case is the second view.

Supplemental Material Figures S4 to S11 give results for Scenario 1, for Cases 1 and 2, when data are standardized, centered, or scaled. Compared to Figures 2 and 3, Supplemental Material Figures S4 to S11 suggest that centering or standardizing data so that each variable has mean zero and variance one prior to implementing the biclustering algorithms result in poor bicluster detection performance. On the other hand, scaling each variable to have variance 1 or each view to have Frobenius norm 1 and implementing iSSVD yield results that are comparable to the original data. The results for s4vd when variables are scaled to have variance 1 are better than unstandardized data for $s > 1$. This is anticipated as this form of scaling results in comparable variables in each view; the two views tend to behave similarly, especially when the simulation

Figure 3. Simulation results for case 2. The boxplots show the distributions of average relevance and recovery over 50 randomly simulated multi-view datasets. Sigmas indicate the value of noise level $\sigma$. Here we fix the scalar to be $s=5$. The values of these indices and F-scores, FP, and FN rates are described in Supplemental Material Table S1.4.

Figure 4. Simulation results for Scenario Two. The boxplots show the distributions of average relevance and recovery over 50 randomly simulated multi-view data. Sigmas indicate the value of noise level $\sigma$. Each randomly simulated data is of size $n=200$ and $p^{(1)} = p^{(2)} = 1,000$. Each bicluster in one view has 50 rows and 100 columns. We ensure that every sample (row) belongs to one of the four biclusters and that there are no overlaps. The average number of unclustered samples, F-scores, FP, and FN rates are described in the Supplemental Material.
setup is essentially the same. However, this is rarely the case when dealing with real-world biomedical data. Further, as discussed in Section 2 of the Supplemental Material, stacking the data and applying s4vd is likely to inflate Type I errors. iSSVD is a biclustering algorithm developed for multi-view data and it can be used in situations where different views have different fundamental scales. As such, we believe that iSSVD is more suitable for multi-view biclustering problems. If normalization is desired, we recommend to standardize each view to have Frobenius norm 1.

3.5.1.2 Varying noise level. The results for case 2 are shown in Figure 3 and Table S1.4 in the Supplemental Material. For noise level 0.1, the average recovery score for iSSVD is almost 1, whereas that of other methods, except s4vd, is considerably lower. This indicates that iSSVD is able to detect all four integrative biclusters in the data. As the noise level increases so that the data become more corrupted, the performance of all methods deteriorate. When σ is larger than 0.5, the median of average relevance of iSSVD is below 0.5. In noisy settings, the methods tend to detect more samples and variables outside the true biclusters as signals and propose them as biclusters, while detecting the true biclusters less frequently. When σ is larger than 0.8, the average relevance scores are almost close to 0, indicating that the ability of these methods to detect biclusters that are representative of the true biclusters are largely impaired by noise. In situations of noisy data, the algorithms tend to assign noise as biclusters and are less capable to detect true hidden structures.

3.5.1.3 Run times. Next we show the runtimes measured in seconds for the biclustering algorithms in Supplemental Material Table S1.1. The time is captured from running every data set in Case 1, that is, with dimensions n = 100 and p = 1,000, and averaged out to get the runtimes of each data set. We use pointwise control for iSSVD and s4vd. The speed of iSSVD is comparable with mvProxL0, mvProxL1, and mvSVD which are written in C++, but s4vd took approximately 10 times longer to finish running the same job. GBC took approximately 70 times longer to execute the same job. In case 3, we use ultra-high-dimensional setting to evaluate the computational efficiency of the algorithms. The dimensions of the data are n = 500 and p = 10,000. On average, iSSVD takes about 2 to 3 min for one Monte Carlo simulation while s4vd takes more than 10 min (Supplemental Material Table S1.2). Furthermore, mvSVD, mvProxL0, and mvProxL1 all have difficulty to converge in this situation. GBC is computationally expensive and thus not practical to run for this case.

3.5.2 Scenario two

The simulation results of Scenario Two are shown in Figure 4 and Supplementary Material Table S2.2. When σ = 0.1, almost all samples are put into their right clusters by iSSVD and only approximately 8 samples are unclustered when σ = 0.2 (Supplemental Material Table S2.1). For s4vd, the recovery scores are high but the relevance scores are significantly lower. This is because s4vd can only detect one integrative bicluster, remaining about three-fourths of the samples unclustered. The performance of mvSVD is also worse than iSSVD. It leaves about one-fourth of the samples unclustered and its scores for relevance and recovery are around 0.5. Other multi-view methods could not detect meaningful integrative biclusters in this scenario, similarly to scenario one. As the noise level increases, s4vd has an unstable performance, as seen by the large variation in the boxplots. Meanwhile, the medians of iSSVD remain the highest among the five methods compared. We can still observe downward trend in the scores as σ increases because the data become messier. We also note that the average relevance scores for s4vd increases, albeit lower than iSSVD, but it has at least half of the samples unclustered over the range of σ (Supplemental Material Table S2.1). For iSSVD, about half of the samples are unclustered when σ > 0.6. Note that mvSVD can assign many samples to biclusters (as observed from the number of unclustered samples in Supplemental Material Table S2.1) but it still achieves lower average relevance and recovery scores, since the samples are not correctly assigned to the true sample clusters. When the errors were generated from the standard t-distribution with degree of freedom 3, we again found the performance of iSSVD to be better than existing methods (refer to Section 3.2 in Supplemental Material).

In Section 3.3 of the Supplemental Material, we explore the stability selection paths for iSSVD for estimated coefficients in one U and V matrices. For a selection probability threshold of 0.6 and above, true coefficients (or samples) that belong to each bicluster are selected (Supplemental Material Figures S15 and S16), with some few noise coefficients also selected in the variable clusters (Supplemental Material Figure S16).

Simulation results for Scenarios One and Two suggest that the proposed method, iSSVD, is better at detecting true biclusters, in both balanced and unbalanced scales settings, when compared to existing multi-view biclustering methods. Further, the proposed method is better for unbalanced scale settings when compared to the biclustering method with stability selection that is applicable to data from one view.
4 Real data analysis

Chronic obstructive pulmonary disease (COPD) is a chronic progressive disease that affects at least 16 million adults in the US and presents a substantial economic and social burden. Like many complex diseases, COPD is characterized by a high degree of heterogeneity with many patients differing in their prognosis and response to therapy. Although the most common cause of COPD is cigarettes smoking, not all smokers go on to develop COPD. Detecting subgroups of people who may be at risk for developing COPD, and who are characterized by specific molecular features is important to understand the heterogeneity in COPD and to improve outcomes for COPD patients. For instance, in Chang et al., gene expression data and a network-based clustering method was used to detect molecular subtypes for COPD.

In this article, we carried out a study that integrated RNA sequencing and proteomics datasets from the COPDGene Study to detect COPD subgroups using the proposed biclustering method. The original data set has 1317 proteins and 21,669 genes. There were 463 samples with proteomics, RNA sequencing and clinical/demographic variables. Prior to applying the proposed and existing methods, we used univariate filtering to reduce the dimensionality of the data. In particular, we regressed each gene and protein with FEV1 (percent predicted force expiratory volume in one second) while controlling for age, gender, and race. We retained genes and proteins with p-value < 0.05. After filtering, we had 229 proteins and 4415 genes for our analyzes. The final datasets for the proteomics and RNA sequencing data were $X^{(1)} \in \mathbb{R}^{463 \times 229}$ and $X^{(2)} \in \mathbb{R}^{463 \times 4415}$, respectively.

| Bicluster (N) | 1 (109) | 2 (111) | 3 (127) | 4 (116) | P-value |
|---------------|---------|---------|---------|---------|---------|
| **Demographics and clinical** | | | | | |
| Age (years) | 71.73 (7.00) | 63.28 (8.21) | 66.80 (8.47) | 67.85 (8.04) | <.00001 |
| Body mass index | 29.46 (5.80) | 29.17 (6.01) | 28.34 (6.56) | 29.77 (6.55) | 0.2333 |
| Pack years | 50.77 (26.53) | 39.57 (19.85) | 43.94 (24.85) | 46.46 (26.69) | 0.01433 |
| Duration of smoking (years) | 37.82 (12.01) | 36.22 (12.11) | 36.60 (11.58) | 36.76 (12.36) | 0.8997 |
| Gender | | | | | |
| Males | 65 (60%) | 52 (47%) | 58 (46%) | 60 (52%) | 0.1402 |
| Females | 44 (40%) | 59 (53%) | 69 (54%) | 56 (48%) | |
| Smoking status | | | | | |
| Former | 94 (86%) | 66 (59%) | 98 (77%) | 94 (81%) | <0.0001 |
| Current | 15 (14%) | 45 (41%) | 29 (23%) | 22 (19%) | |
| Dyspnea score (MMRC) | 1.34 (1.45) | 0.77 (1.17) | 1.18 (1.28) | 1.02 (1.27) | 0.0048 |
| BODE index | 2.20 (2.44) | 0.90 (1.51) | 1.48 (2.07) | 1.18 (1.64) | 0.0005 |
| **Outcomes** | | | | | |
| FEV1/FVC | 0.60 (0.17) | 0.72 (0.11) | 0.67 (0.16) | 0.66 (0.15) | <0.00001 |
| FEV1 (% predicted) | 70.43 (27.88) | 85.23 (21.04) | 78.63 (25.75) | 77.07 (24.51) | 0.0026 |
| COPD status | No | 35 (34%) | 56 (57%) | 57 (51%) | 48 (47%) | 0.0096 |
| | Yes | 67 (66%) | 42 (43%) | 54 (49%) | 54 (53%) | |
| **Symptoms** | | | | | |
| Exacerbation frequency | 0.29 (0.86) | 0.19 (0.56) | 0.20 (0.58) | 0.16 (0.49) | 0.6571 |
| Percent emphysema (thirona) | 9.66 (11.19) | 3.45 (5.25) | 8.52 (12.76) | 7.26 (10.52) | <0.0001 |
| Percent 15 | -931.71 (25.00) | -919.20 (19.48) | -926.60 (28.42) | -925.15 (25.57) | 0.0021 |
| Ever had asthma | No | 104 (95.4%) | 109 (98.2%) | 125 (98.4%) | 112 (96.6%) | 0.4798 |
| | Yes | 5 (4.6%) | 2 (1.8%) | 2 (1.6%) | 4 (3.4%) | |
| Gastroesophageal reflux | No | 69 (63%) | 86 (77%) | 79 (62%) | 74 (64%) | 0.0473 |
| | Yes | 40 (37%) | 25 (23%) | 48 (38%) | 42 (36%) | |
| Been to ER or hospitalized for lung problems | No | 98 (90%) | 106 (95.5%) | 116 (91.3%) | 108 (93.1%) | 0.4274 |
| | Yes | 11 (10%) | 5 (4.5%) | 11 (8.7%) | 8 (6.9%) | |
We applied our algorithm and the other algorithms considered in the simulation section to the filtered data. $s4vd$ was applied to the concatenated filtered RNA sequencing and proteomics data. For $iSSVD$ and $s4vd$, we allowed the algorithms to detect the number of biclusters. We also did not allow for overlaps in rows and columns; that is, each sample and variable is assigned to only one bicluster. This will allow us to detect distinct patient clusters that are characterized by distinct biomarkers. Four biclusters were detected by $iSSVD$ and 10 samples were unclustered. $s4vd$ did not detect any stable cluster. For $mvProxL0$, $mvProxL1$, and $mvSVD$, we manually subset samples that are clustered when obtaining subsequent biclusters. In the end, five biclusters were obtained for $mvProxL0$, with 13 samples that were not clustered. $mvSVD$ and $mvProxL1$ detected four and two biclusters, respectively. One sample was not clustered for $mvProxL1$. GBC identified four biclusters but many samples identified by GBC belonged to multiple clusters. To avoid overlaps and to facilitate comparisons, we made random bicluster assignment for samples that overlapped. Please refer to the Supplemental Material Table S1.6 for our parameter settings.

### 4.1 Clinical characteristics of sample clusters identified

For all methods, we assigned the unclustered samples to the detected biclusters as follows. For each method, we obtained the first principal component (PC) from principal component analysis (PCA) of data for each sample cluster detected. That is, for each sample cluster, we used only the variables characterizing that sample to obtain the principal components. The first PC for a sample cluster explains the maximum variation and can be used to summarize the information for that cluster. We correlated the first PC for each sample cluster with data for each sample that was not clustered. We then assigned that sample to the bicluster with the highest correlation. Thus, in the end, all samples belonged to one bicluster. The first $iSSVD$ bicluster consisted of 109 samples, 213 genes, and 27 proteins. The second bicluster was made of 111 samples, 195 genes, and 25 proteins. The third bicluster had 127 samples, 165 genes, and 25 proteins. The fourth bicluster comprised of 116 samples, 169 genes, and 19 proteins.

Compared to $iSSVD$, all other competing methods with the exception of $mvProxL0$, did not find meaningful sample subgroups from the data. The sample clusters detected by $mvProxL1$, $mvSVD$, and GBC were not differentiated on lung function (as measured by FEV1/FVC, FEV1 [% predicted]), demographics/clinical variables, and symptoms (Tables S4.1 to and S4.4 in the Supplemental Material). The sample clusters detected by $mvProxL0$ showed differences in some variables, but were not different across key lung function variables such as FEV1/FVC and FEV1 [% predicted]). Compared to $mvProxL0$, the sample clusters detected by our method showed differences in more variables.

The four sample clusters identified by $iSSVD$ where well-differentiated on some key demographic, clinical, and outcome variables such as age, FEV1/FVC ratio, and BODE index (Table 1). The FEV1/FVC ratio is widely used to diagnose COPD. FEV1 (forced expiratory volume in one second) is the volume of breath exhaled in one second, and it is used to gauge severity of COPD. The BODE (body mass index, obstruction, dyspnea, and exercise) index is a multidimensional assessment of an individual’s risk of death; higher values suggest increased risk. From Table 1, individuals in Biclusters 1, 3, and 4 have lower lung function as depicted by FEV1 compared to individuals in Bicluster 2. Bicluster 1 has a lower mean FEV1/FVC ratio and lower mean FEV1 value, followed by Biclusters 3 and 4. Individuals in Bicluster 2 have better lung function as can be observed by the higher FEV1/FVC ratio, FEV1 percent predicted and lower mean dyspnea and BODE index values. Using the terminology in Chang et al., we refer to Bicluster 1 as the “severely affected” group, Bicluster 3 as the “moderately affected” group, and “Bicluster 4” as the less affected group. Also, we refer to Bicluster 2 as the “preserved lung function” group since the mean FEV1/FVC ratio and FEV1 values were on average within normal limits. Bicluster 1, the severely affected group, also had a higher BODE index and a higher dyspnea score. They were more likely to be old, males and they tended to be previous smokers. Bicluster 3, the moderately affected group, had the next highest dyspnea score, BODE index, and frequency of exacerbation, followed by Bicluster 4, the less affected group. The preserved lung function group, Bicluster 2, was predominantly females, who were less likely to have COPD as defined by FEV1/FVC ratio, reported less symptoms (e.g. lower emphysema as measured by Thirona and 15%), had lower mean age, and tended to be current smokers. We discuss some differences between the biclusters detected by our method and the clusters derived by Chang et al. Five-year follow-up data were available for 162 individuals. We observe that the subgroups identified by $iSSVD$ are again well-differentiated on some key outcomes, clinical variables and symptoms (Supplemental Material Table S4.5). Biclusters 1, 3, and 4 again had lower FEV1/FVC, lower FEV1 percent predicted, higher BODE index and higher dyspnea score. The group with preserved lung function (Bicluster 2) had lower dyspnea score, lower BODE index, better lung function, and were less likely to have visited the emergency room (ER) or to be hospitalized for lung problems.
4.2 Subtype-specific biologic pathway and disease enrichment analysis for variable clusters characterizing sample clusters identified by iSSVD

We used the Ingenuity Pathway Analysis (IPA) software to investigate the molecular and cellular functions, pathways, and diseases enriched in the proteins and genes identified for each bicluster. IPA searches the ingenuity pathway knowledge base, which is manually curated from scientific literature and over 30 databases, for gene interaction. We focused on the variable clusters identified by iSSVD since the sample subgroups identified by this method were well-differentiated on many clinical, demographic, and outcome variables. We observed strong enrichment of functional pathways (Supplemental Material Tables S4.6 and S4.7). Some of the significantly enriched canonical pathways that mapped to the gene and protein lists for the severely affected group (Bicluster 1) included IL-8 signaling, mTOR pathway, and Intrinsic Prothrombin activation pathway. The mTOR signaling pathway is involved in many cellular processes such as cell growth, metabolism, and survival. Research suggests that the activation of the mTOR signaling pathway can induce cell senescence in the lung, which in turn can result in COPD. Some pathways enriched for the moderately affected group (Bicluster 3) included airway pathology in COPD and oxidative phosphorylation. Also, the inflammasome pathway, airway pathology in COPD, and IL-17 signaling pathways were over-represented in our gene and protein lists for the less affected group (Bicluster 4). Some pathways enriched for the preserved lung function group (Bicluster 2) included taurine biosynthesis and enhanced cardiac hypertrophy signaling pathway. While there were overlaps in some of the enriched pathways for the clusters, there were also unique pathways identified.

In addition to IPA canonical pathways, proteins and genes were also categorized to related diseases and functions. Again, there were some overlaps in the top 5 enriched diseases and functions for the clusters (Supplemental Material Tables S4.11 and S4.12). The severely affected group (Bicluster 1) was characterized by diseases that included cardiovascular (such as atherosclerosis, ventricular dysfunction and peripheral vascular disease) and inflammatory (such as chronic inflammatory disorder) diseases. Some of the diseases characterizing the moderately affected group (Bicluster 3) included cardiovascular (such as infarction and ischemia of brain) and cancer. The less affected group (Bicluster 4) was characterized by neurological (including abnormal regeneration by peripheral nervous system and cerebrovascular dysfunction) and hereditary disorder. Also, the preserved lung function group (Bicluster 2) was characterized by an inflammatory response (such as inflammation of lung) and cancer. We also used IPA for network analysis to connect key genes, proteins, and enriched categories of diseases and functions. Our results showed that the severely affected group was characterized by the cardiovascular disease, organismal injury and abnormalities, hematological system development and function (based on our protein list). The preserved lung function group was characterized by the dermatological diseases and conditions, organismal injury and abnormalities, organismal development network (from our protein list).

5 Discussion and conclusion

In this article, we extended existing biclustering method based on sparse singular value decomposition for data from one view to data from multiple views. Our method followed ideas in Sill et al. and incorporated stability selection, a subsampling based variable selection that allows to control Type I error rates. The proposed algorithm estimates the probability of samples and variables to belong to a bicluster, finds stable biclusters, and results in interpretable row-column associations. The proposed algorithm, developed in Python 3, is computationally efficient and user-friendly and will be useful in many disease subtyping applications. Through simulation studies, we showed that the proposed method outperforms several other single- and multi-view biclustering methods in detecting artificial biclusters.

When our method was applied to RNA sequencing and proteomics data from the COPDGene study, we detected four biclusters that were well-differentiated by some demographics and clinical variables as well as key COPD outcomes. Three biclusters which we call severely, moderately, and less affected groups, seemed to have poor lung function and clinical outcomes, while one bicluster, which we call preserved lung function group seemed to have better (preserved) lung function and clinical outcomes. We also performed an enrichment analysis of the genes and proteins characterizing the sample clusters. While certain biological processes were most enriched in specific biclusters, there was also notable overlap in processes across biclusters. Particularly enriched molecular and cellular functions included cellular movement, cellular growth and proliferation and cell death and survival.

The limitations of our work include the following. First, when applied to real data, a smaller preset Type I error rate tends to yield small biclusters, which results in more unclustered samples. However, increasing the error rate might compromise the strength against noise. Second, we mention ways to choose the number of biclusters $K$ but these could be improved, especially in situations of noisy data. Third, we noticed that under regular settings, the run times of iSSVD and s4vd tend to be considerably longer than point-wise control settings. However, since we are not searching for the
entire range of regularization parameters in a point-wise control, it is likely that we overlook the optimal regularization parameters. Thus, future work can seek to improve the computational time to accommodate the computational demands for stability selection that searches the entire range of regularization parameters.

In conclusion, we have developed a biclustering method for multi-view data capable of detecting stable row and column clusters. The encouraging simulation and real data findings motivate further applications to detect disease subtypes and subtype-specific molecular features.

Acknowledgements
The project described was supported by grant 5KL2TR002492-03 from the National Institutes of Health, 1R35GM142695-01 from the National Institute Of General Medical Sciences of the National Institutes of Health, and by Award Number U01 HL089897 and Award Number U01 HL089856 from the National Heart, Lung, and Blood Institute. COPDGene is also supported by the COPD Foundation through contributions made to an Industry Advisory Board that has included AstraZeneca, Bayer Pharmaceuticals, Boehringer-Ingelheim, Genentech, GlaxoSmithKline, Novartis, Pfizer, and Sunovion.

Disclaimer
The views expressed in this article are those of the authors and do not reflect the views of the United States Government, the Department of Veterans Affairs, the funders, the sponsors, or any of the authors’ affiliated academic institutions.

Declaration of conflicting interests
The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The project described was supported by grant 5KL2TR002492-03 from the National Institutes of Health, 1R35GM142695-01 from the National Institute Of General Medical Sciences of the National Institutes of Health, and by Award Number U01 HL089897 and Award Number U01 HL089856 from the National Heart, Lung, and Blood Institute. COPDGene is also supported by the COPD Foundation through contributions made to an Industry Advisory Board that has included AstraZeneca, Bayer Pharmaceuticals, Boehringer-Ingelheim, Genentech, GlaxoSmithKline, Novartis, Pfizer, and Sunovion.

Data availability
The data used were provided by the COPDGene Study group. COPDGene clinical and RNA Sequencing data are available on dbGap. We provide a Python package, \texttt{iSSVD}, to facilitate the use of our method. Its source codes, along with a README file are available via https://github.com/weijie25/iSSVD this link.

ORCID iD
Sandra E Safo https://orcid.org/0000-0001-9593-4778

Supplemental material
Supplemental material for this article is available online.

References
1. Henriques R and Madeira SC. Triclustering algorithms for three-dimensional data analysis: a comprehensive survey. ACM Comput Surv 2018; 51: 1–43.
2. Getz G, Levine E and Domany E. Coupled two-way clustering analysis of gene microarray data. Proc Natl Acad Sci USA 2000; 97: 12079–12084.
3. Ben-Dor A, Chor B, Karp R, et al. Discovering local structure in gene expression data: the order-preserving submatrix problem. In: Proceedings of the sixth annual international conference on Computational biology. 2002; pp. 49–57.
4. Prelic A, Bleuler S, Zimmermann P, et al. A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 2006; 22: 1122–1129.
5. Pandey G, Atluri G, Steinbach M, et al. An association analysis approach to biclustering. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. 2009; pp. 677–686.
6. Huttenhower C, Mutungu KT, Indik N, et al. Detailing regulatory networks through large scale data integration. Bioinformatics 2009; 25: 3267–3274.
7. Li G, Ma Q, Tang H, et al. Qubic: a qualitative biclustering algorithm for analyses of gene expression data. Nucleic Acids Res 2009; 37: e101–e101.
8. Xie J, Ma A, Zhang Y, et al. Qubic2: a novel and robust biclustering algorithm for analyses and interpretation of large-scale rna-seq data. *Bioinformatics* 2020; 36: 1143–1149.

9. Tanay A, Sharan R and Shamir R. Discovering statistically significant biclusters in gene expression data. *Bioinformatics* 2002; 18: S136–S144.

10. Hochreiter S, Bodenhofer U, Heusel M, et al. Fabia: factor analysis for bicluster acquisition. *Bioinformatics* 2010; 26: 1520–1527.

11. Gao C, McDowell IC, Zhao S, et al. Context specific and differential gene co-expression networks via Bayesian biclustering. *PLoS Comput Biol* 2016; 12: e1004791.

12. Chi EC, Allen GI and Baraniuk RG. Convex biclustering. *Biometrics* 2017; 73: 10–19.

13. Li Z, Chang C, Kundu S, et al. Bayesian generalized biclustering analysis via adaptive structured shrinkage. *Biostatistics* 2018; 21: 610–624.

14. Lazzeroni L and Owen A. Plaid models for gene expression data. *Stat Sin* 2002; 12: 61–86.

15. Lee M, Shen H, Huang JZ, et al. Biclustering via sparse singular value decomposition. *Biometrics* 2010; 66: 1087–1095.

16. Sill M, Kaiser S, Benner A, et al. Robust biclustering by sparse singular value decomposition incorporating stability selection. *Biometrics (Oxford, England)* 2011; 27: 2089–2097.

17. Tepper M and Sapiro G. Compressed nonnegative matrix factorization is fast and accurate. *IEEE Trans Signal Process* 2016; 64: 2269–2283.

18. Xiang L, Zhu K and Lu S. Bern: mining coregulation patterns in transcriptomics via Boolean matrix factorization. *Bioinformatics* 2020; 36: 4030–4037.

19. Sun H, Miao G and Yan X. Noise-resistant bicluster recognition. In: 2013 IEEE 13th International Conference on Data Mining, IEEE, 2013, pp. 707–716.

20. Li J, Reisner J, Pham H, et al. Biclustering with missing data. *Inf Sci (Ny)* 2020; 510: 304–316.

21. Vandromme M, Jacques J, Taillard J, et al. A biclustering method for heterogeneous and temporal medical data. *IEEE Trans Knowl Data Eng* 2020; 34: 1–1.

22. Michalak M, Jakšik R and Šležak D. Heuristic search of exact biclusters in binary data. *Int J Appl Math Comput Sci* 01 Mar. 2020; 30: 161–171.

23. Sun J, Bi J and Kranzler HR. Multi-view singular value decomposition for disease subtyping and genetic associations. *BMC Genet* 2014; 15: 73.

24. Sun J, Lu J, Xu T, et al. Multi-view sparse co-clustering via proximal alternating linearized minimization. In: *International Conference on Machine Learning*, 2015, pp. 757–766.

25. Bunte K, Leppäaho E, Saarinen I, et al. Sparse group factor analysis for biclustering of multiple data sources. *Bioinformatics* 2016; 32: 2457–2463.

26. Zou H. The adaptive lasso and its oracle properties. *J Am Stat Assoc* 2006; 101: 1418–1429.

27. Meinshausen N and Bühlmann P. Stability selection. *J R Stat Soc: Ser B (Statistical Methodology)* 2010; 72: 417–473.

28. Regan EA, Hokanson JE, Murphy JR, et al. Genetic epidemiology of copd (copdgene) study design. *COPD* 2011; 7: 32–43.

29. Liu B, Shen X and Pan W. Integrative and regularized principal component analysis of multiple sources of data. *Stat Med* 2016; 35: 2235–2250.

30. Tibshirani R. Regression shrinkage and selection via the lasso. *J R Stat Soc: Ser B (Methodological)* 1996; 58: 267–288.

31. Schwarz G, et al. Estimating the dimension of a model. *Ann Stat* 1978; 6: 461–464.

32. Zou H. Biclustering and its oracle properties. *J Am Stat Assoc* 2006; 101: 1418–1429.

33. Meinshausen N and Bühlmann P. Stability selection. *J R Stat Soc: Ser B (Statistical Methodology)* 2010; 72: 417–473.

34. Regan EA, Hokanson JE, Murphy JR, et al. Genetic epidemiology of copd (copdgene) study design. *COPD* 2011; 7: 32–43.

35. Liu B, Shen X and Pan W. Integrative and regularized principal component analysis of multiple sources of data. *Stat Med* 2016; 35: 2235–2250.

36. Tibshirani R. Regression shrinkage and selection via the lasso. *J R Stat Soc: Ser B (Methodological)* 1996; 58: 267–288.

37. Schwarz G, et al. Estimating the dimension of a model. *Ann Stat* 1978; 6: 461–464.

38. Zou H. Biclustering and its oracle properties. *J Am Stat Assoc* 2006; 101: 1418–1429.

39. Meinshausen N and Bühlmann P. Stability selection. *J R Stat Soc: Ser B (Statistical Methodology)* 2010; 72: 417–473.

40. Regan EA, Hokanson JE, Murphy JR, et al. Genetic epidemiology of copd (copdgene) study design. *COPD* 2011; 7: 32–43.

41. Liu B, Shen X and Pan W. Integrative and regularized principal component analysis of multiple sources of data. *Stat Med* 2016; 35: 2235–2250.