Group-sparse SVD Models and Their Applications in Biological Data

Wenwen Min, Juan Liu and Shihua Zhang

Abstract—Sparse Singular Value Decomposition (SVD) models have been proposed for biclustering high dimensional gene expression data to identify block patterns with similar expressions. However, these models do not take into account prior group effects upon variable selection. To this end, we first propose group-sparse SVD models with group Lasso ($GL_1$-SVD) and group $L_0$-norm penalty ($GL_0$-SVD) for non-overlapping group structure of variables. However, such group-sparse SVD models limit their applicability in some problems with overlapping structure. Thus, we also propose two group-sparse SVD models with overlapping group Lasso ($OGL_1$-SVD) and overlapping group $L_0$-norm penalty ($OGL_0$-SVD). We first adopt an alternating iterative strategy to solve $GL_1$-SVD based on a block coordinate descent method, and $GL_0$-SVD based on a projection method. The key of solving $OGL_1$-SVD is a proximal operator with overlapping group Lasso penalty. We employ an alternating direction method of multipliers (ADMM) to solve the proximal operator. Similarly, we develop an approximate method to solve $OGL_0$-SVD. Applications of these methods and comparison with competing ones using simulated data demonstrate their effectiveness. Extensive applications of them onto several real gene expression data with gene prior group knowledge identify some biologically interpretable gene modules.

Index Terms—sparse SVD, low-rank matrix decomposition, group-sparse penalty, overlapping group-sparse penalty, coordinate descent method, alternating direction method of multipliers (ADMM), data mining

1 INTRODUCTION

Singular Value Decomposition (SVD) is one of the classical matrix decomposition models [1]. It is a useful tool for data analysis and low-dimensional data representation in many different fields such as signal processing, matrix approximation and bioinformatics [2], [3], [4]. However, the non-sparse singular vectors with all variables are difficult to be explained intuitively. In the recent years, sparse models have been widely applied in computational biology to improve biological interpretation [5], [6], [7]. In addition, many researchers applied diverse sparse penalties onto singular vectors in SVD and developed multiple sparse SVD models to improve their interpretation and capture inherent structures and patterns from the input data [8], [9]. For example, sparse SVD provides a new way for exploring bicluster patterns of gene expression data. Suppose $X \in \mathbb{R}^{p \times n}$ denotes a gene expression matrix with $p$ genes and $n$ samples. Biologically, a subset of patients and genes can be clustered together as a coherent bicluster or block pattern with similar expressions. Previous studies have reported that such a bicluster among gene expression data can be identified by low-rank sparse SVD models [10], [11], [12]. However, these sparse models ignore prior information of gene variables, and usually assume that each gene is selected in a bicluster with equal probability. Actually, one gene may belong to multiple pathways in biology [13]. As far as we know, there is not yet a model for biclustering gene expression data by integrating gene pathway information. Group sparse penalties [14], [15] should be used to induce the structured sparsity of variables for variable selection. Several studies have explored the (overlapping) group Lasso in regression tasks [16], [17]. However, little work focus on developing structured sparse SVD for biclustering high-dimensional data (e.g., biclustering gene expression data via integrating prior gene group knowledge).

In this paper, motivated by the development of sparse coding and structured sparse penalties, we propose several group-sparse SVD models for pattern discovery in biological data. We first introduce the group-sparse SVD model with group Lasso ($L_1$) penalty ($GL_1$-SVD) to integrate non-overlapping structure of variables. Compared to $L_1$-norm, $L_0$-norm is a more natural sparsity-inducing penalty. Thus, we also propose an effective group-sparse SVD via replacing $L_1$-norm with $L_0$-norm, called $GL_0$-SVD, which uses a mix-norm by combining the group Lasso and $L_0$-norm penalty. However, the non-overlapping group structure limits their applicabilities in diverse fields. We consider a more general situation, where we assume that either groups of variables are potentially overlapping (e.g., a gene may belong to multiple pathways (groups)). We also propose two group-sparse SVD models with overlapping group Lasso ($OGL_1$-SVD) and overlapping group $L_0$-norm penalty ($OGL_0$-SVD).

To solve these models, we design an alternating iterative algorithm to solve $GL_1$-SVD based on a block coordinate descent method and $GL_0$-SVD based on a projection method. Furthermore, we develop a more general approach based
on Alternating Direction Method of Multipliers (ADMM) to solve \( OGL_1\text{-SVD} \). In addition, we extend \( OGL_1\text{-SVD} \) to \( OGL_0\text{-SVD} \), which is a regularized SVD with overlapping grouped \( L_0 \)-norm penalty. The key of solving \( OGL_1\text{-SVD} \) is also a proximal operator with overlapping group \( L_0 \)-norm penalty. We propose a greedy method to solve it and obtain its approximated solution. Finally, applications of these methods and comparison with the state-of-the-art ones using a set of simulated data demonstrate their effectiveness and computational efficiency. Extensive applications of them onto the high-dimensional gene expression data show that our methods could identify more biologically relevant gene modules, and improve their biological interpretations.

**Related Work** We briefly review the regularized low rank-\( r \) SVD model as follows:

\[
\begin{align*}
\text{minimize} & \quad \|X - UDVT\|_F^2 \\
\text{subject to} & \quad \|U_i\|^2 \leq 1, \Omega_1(U_i) \leq c_1, \forall i \\
& \quad \|V_i\|^2 \leq 1, \Omega_2(V_i) \leq c_2, \forall i
\end{align*}
\]

where \( X \in \mathbb{R}^{p \times n} \) with \( p \) features and \( n \) samples, \( U \in \mathbb{R}^{p \times r} \), \( V \in \mathbb{R}^{r \times n} \) and \( D \) is diagonal matrix. \( U_i \) (\( V_i \)) corresponds to the \( i \)-th column of \( U \) (\( V \)), which is a column orthogonal matrix. To solve the above optimization problem, we introduce a general regularized rank-one SVD model:

\[
\begin{align*}
\text{minimize} & \quad \|X - d uv^T\|_F^2 \\
\text{subject to} & \quad \|u\|^2 \leq 1, \Omega_1(u) \leq c_1, \\
& \quad \|v\|^2 \leq 1, \Omega_2(v) \leq c_2,
\end{align*}
\]

where \( d \) is a positive singular value, \( u \) is a \( p \)-dimensional column vector, and \( v \) is a \( n \)-dimensional column vector. \( \Omega_1(u) \) and \( \Omega_2(v) \) are two penalty functions, \( c_1 \) and \( c_2 \) are two hyperparameters. In a Bayesian view, different prior distribution functions of \( u \) and \( v \) correspond to different regularized functions. For example, \( L_1 \)-norm is a very popular sparsity-inducing norm [18] and has been used to obtain sparse solutions in a large number of statistical models including the regression model [18], [19], SVD [20], PCA [21], K-means [22], etc.

Recently, some sparse SVD models have been proposed for coherent sub-matrix detection [20], [10], [11]. For example, Witten et al. [20] developed a penalized matrix decomposition (PMD) method, which regularizes the singular vectors with Lasso and fused Lasso to induce sparsity. Lee et al. [10] proposed a rank-one sparse SVD model with adaptive Lasso \((L_1)\) \((L_1\text{-SVD})\) of the singular vectors for biclustering of gene expression data. Some generalized sparsity penalty functions (e.g., group Lasso [14] and sparse group lasso [24]) have been widely used in many regression models for feature selection by integrating group information of variables. However, it is a challenging issue to use these generalized penalty functions such as group Lasso and overlapping group Lasso [15], [25] in the SVD framework with effective algorithms. To this end, we develop several group-sparse SVD models with different group-sparse penalties including \( \Omega_{GL}(u) \), \( \Omega_{GL_0}(u) \), \( \Omega_{OGL}(u) \) and \( \Omega_{OGL_0}(u) \) to integrate diverse group structures of variables for pattern discovery in biological data (see TABLE 1).

**2 Group-sparse SVD Models**

In this section, we propose four group sparse SVD models with respect to different structured penalties (TABLE 1). For a given data (e.g., gene expression data), we can make proper adjustments to get one-sided group-sparse SVD models via using (overlapping) group-sparse penalties for the right (or left) singular vector. For example, \( SVD(OGL_0, L_0) \) is a group-sparse SVD model, which uses the overlapping group \( L_0 \)-penalty for \( u \) and \( L_0 \)-penalty for \( v \) respectively.

| Model | Penalty function |
|-------|------------------|
| \( GL_1\text{-SVD} \) | Group/Lasso \((GL_1)\) |
| \( GL_0\text{-SVD} \) | Group-\( L_0 \) \((GL_0)\) |
| \( OGL_1\text{-SVD} \) | Overlapping-Group-Lasso \((OGL_1)\) |
| \( OGL_0\text{-SVD} \) | Overlapping-Group-\( L_0 \) \((OGL_0)\) |

Below we will introduce these models and their algorithms in detail.

**2.1 \( GL_1\text{-SVD} \)**

Suppose the left singular vector \( u \) and right singular vector \( v \) can be respectively divided into \( L \) and \( M \) non-overlapping groups: \( u^{(l)} \in \mathbb{R}^{p \times 1}, l = 1, \ldots, L \) and \( v^{(m)} \in \mathbb{R}^{1 \times n}, m = 1, \ldots, M \).
\( \mathbb{R}^{pm \times 1}, m = 1, \ldots, M \). Here, we consider the (adaptive) group Lasso (GLL) penalty [26] for \( u \) and \( v \) as follows:

\[
\Omega_{GLL}(u) = \sum_{l=1}^{L} w_l \| u^{(l)} \|_2 \quad \text{and} \quad \Omega_{GLL}(v) = \sum_{m=1}^{M} \tau_m \| v^{(m)} \|_2,
\]

where both \( w_l \) and \( \tau_m \) are adaptive weight parameters. Suppose \( w_l = \sqrt{P_l} \) and \( \tau_m = \sqrt{q_m} \) for group sizes, the penalty reduces to a traditional group Lasso.

Based on the definition of GLL penalty, we propose the first group-sparse SVD with group Lasso penalty (GLL-SVD), also namely SVD(GLL, GLL):

\[
\min_{u, v, \eta} \| X - duv^T \|_F^2,
\]

subject to \( \| u \|_2 \leq 1, \Omega_{GLL}(u) \leq c_1, \| v \|_2 \leq 1, \Omega_{GLL}(v) \leq c_2.\)

Since \( \| X - duv^T \|_F^2 = \| X \|_F^2 + d^2 - 2duv^TX \). Minimizing \( \| X - duv^T \|_F^2 \) is equivalent to minimizing \( -uv^TX \), and once the \( u \) and \( v \) are determined, the \( d \) value is determined by \( u^TXv \). We obtain the Lagrangian form of GLL-SVD model as follows:

\[
\mathcal{L}(u, v) = -u^TXv + \lambda_1 \Omega_{GLL}(u) + \lambda_2 \Omega_{GLL}(v) + \eta_1 \| u \|_2^2 + \eta_2 \| v \|_2^2,
\]

where \( \lambda_1 \geq 0, \lambda_2 \geq 0, \eta_1 \geq 0 \) and \( \eta_2 \geq 0 \) are Lagrange multipliers. To solve the problem (5), we apply an alternating iterative algorithm to optimize \( u \) for a fixed \( v \) and \textit{vice versa}.

### 2.1.1 Learning \( u \)

Fix \( v \) and let \( z = Xv \), minimizing Eq. (5) is equivalent to minimizing the following criterion:

\[
\mathcal{L}(u, \lambda, \eta) = -u^TXv + \lambda \sum_{l=1}^{L} w_l \| u^{(l)} \|_2 + \eta \sum_{l=1}^{L} \| u^{(l)} \|_2,
\]

where \( u = [u^{(1)}; u^{(2)}; \ldots; u^{(L)}] \) and \( \lambda = \lambda_1, \eta = \eta_1 \) for simplicity. It is obvious that \( \mathcal{L}(u, \lambda, \eta) \) is convex with respect \( u \), and we develop a block coordinate descent algorithm [27, 28, 29, 30] to minimize Eq. (6), i.e. one group of \( u \) is updated at a time. For a single group \( u^{(j)} \) with fixed \( u^{(j)} \) for all \( 1 \leq j \leq L \) and \( j \neq l \), the subgradient equations (see [31]) of Eq. (6) with respect to \( u^{(l)} \) is written as:

\[
\nabla_{u^{(l)}} \mathcal{L} = -z^{(l)} + \lambda w_l s^{(l)} + 2\eta u^{(l)} = 0,
\]

where \( s^{(l)} \) is the subvector of \( u^{(l)} \) and it meets

\[
s^{(l)} = \begin{cases} \frac{u^{(l)}}{\| u^{(l)} \|_2}, & \text{if } u^{(l)} \neq 0, \\ \epsilon \in \{s^{(l)}: \| s^{(l)} \|_2 \leq 1\}, & \text{otherwise}. \end{cases}
\]

Based on Eq. (7), we have \( 2\eta u^{(l)} = z^{(l)} - \lambda w_l s^{(l)} \).

If \( \| z^{(l)} \|_2 > \lambda w_l \), then we have \( u^{(l)} \neq 0 \). Since \( \eta > 0, \lambda > 0, w_l > 0 \) and \( 2\eta u^{(l)} = z^{(l)} - \lambda w_l s^{(l)} \). Thus, we have \( u^{(l)} = \frac{z^{(l)}}{\| z^{(l)} \|_2} \) and \( \eta = \frac{1}{2\eta} (1 - \lambda w_l \| z^{(l)} \|_2). \)

If \( \| z^{(l)} \|_2 \leq \lambda \), then \( u^{(l)} = 0 \). In short, we obtain the following update rule for \( u^{(l)} \) (\( l = 1, \ldots, L \)),

\[
u^{(l)} = \begin{cases} \frac{1}{2\eta} (1 - \lambda w_l \| z^{(l)} \|_2) z^{(l)}, & \text{if } \| z^{(l)} \|_2 > \lambda w_l, \\ 0, & \text{otherwise}. \end{cases}
\]

Since Eq. (6) is strictly convex and \textit{separable}, the block coordinate descent algorithm must converge to its optimal solution [27]. Finally, we can choose an \( \eta \) to guarantee \( u = \frac{u}{\| u \|_2} \) (normalizing condition).

### 2.1.2 Learning \( v \)

In the same manner, we fix \( u \) in Eq. (5) and let \( z = X^Tu \). Similarly, we can also obtain the coordinate update rule for \( v^{(m)}, m = 1, 2, \ldots, M \).

\[
v^{(m)} = \begin{cases} \frac{1}{2\eta} (1 - \frac{\lambda \tau_m}{\| z^{(m)} \|_2}) z^{(m)}, & \text{if } \| z^{(m)} \|_2 > \lambda \tau_m, \\ 0, & \text{otherwise}. \end{cases}
\]

Furthermore, to meet the normalizing condition, we chose an \( \eta \) to guarantee \( v = \frac{v}{\| v \|_2} \). Besides, if here each group only contains one element, then the group Lasso penalty reduces to the Lasso penalty. Accordingly, we get another update formula:

\[
v_t = \begin{cases} \frac{1}{2\eta} (1 - \frac{\lambda}{\| z \|_2}) z_t, & \text{if } \| z \|_2 > \lambda, \\ 0, & \text{otherwise}. \end{cases}
\]

### 2.1.3 GLL-SVD Algorithm

Based on Eqs. (9) and (10), we propose an alternating iterative algorithm (Algorithm 1) to solve the GLL-SVD model and its time complexity is \( \mathcal{O}(TnP_T + TP_T^2 + Tn^2) \), where \( T \) is the number of iterations. We can control the iteration by monitoring the change of \( d \).

In order to display the penalty function for left and right singular vectors, GLL-SVD can also be written in another form SVD(GLL, GLL), denoting that the left singular vector \( u \) is regularized by GLL penalty and the right singular vector \( v \) is regularized by GLL penalty, respectively. Similarly, we can simply modify Algorithm 1 to solve SVD(GLL, L1) model, which applies Lasso as the penalty for \( v \).

### 2.2 GL0-SVD

Unlike GLL penalty, below we consider a group \( L_0 \)-norm penalty (GL0) of \( u \) and \( v \) as follows:

\[
\Omega_{GL0}(u) = \| \phi(u) \|_0 \quad \text{and} \quad \Omega_{GL0}(v) = \| \phi(v) \|_0,
\]

where \( \phi(u) = [\| u^{(1)} \|_1; \| u^{(2)} \|_1; \ldots; \| u^{(L)} \|_1]^T \) and \( \phi(v) = [\| v^{(1)} \|_1; \| v^{(2)} \|_1; \ldots; \| v^{(M)} \|_1]^T \).

Based on the above definition of GL0 penalty, we propose the second group-sparse SVD model with GL0 penalty, namely GL0-SVD or SVD(GL0, GL0):

\[
\min_{u, v, \eta} \| X - duv^T \|_F^2,
\]

subject to \( \| u \|_2 \leq 1, \Omega_{GL0}(u) \leq k_u, \| v \|_2 \leq 1, \Omega_{GL0}(v) \leq k_v. \)

Here, we employ an alternating iterative strategy to solve problem (13). Fix \( u \) (or \( v \)), the problem (13) reduces to a projection problem with group \( L_0 \)-norm penalty.
The optimum solution of Eq. (14) is Theorem 1.

We present Theorem 1 to solve problem (14).

2.2.1 Learning u

Since $\|X - dwv^T\|_F^2 = \|X\|_F^2 + d^2 - 2dw^T Xv$. Fix v and let $z_u = X v$, Eq. (13) reduces to a group-sparse projection operator with respect to $u$:

$$\min_u - z_u^T u, \quad \text{s.t. } \Omega_{GL_0}(u) \leq k_u. \quad (14)$$

We present Theorem 1 to solve problem (14).

**Theorem 1.** The optimum solution of Eq. (14) is $P_{GL_0}(z_u)\|P_{GL_0}(z_u)\|_1$, where $P_{GL_0}(z_u)$ is a column-vector and meets

$$[P_{GL_0}(z_u)](g) = \begin{cases} z_u(g), & \text{if } g \in \text{supp}(\phi(z_u), k_u), \\ 0, & \text{otherwise}, \end{cases} \quad (15)$$

where $[P_{GL_0}(z_u)](g)$ is a sub-vector from the g-th group, $g = 1, 2, \ldots, L$ and $\text{supp}(\phi(z_u), k_u)$ denotes the set of indexes of the largest $k_u$ elements of $\phi(z_u)$.

The objective function of (14) can be simplified as $-z_u^T u = \sum_{i=1}^L -z_u^{(i)} T u^{(i)}$. Theorem 1 shows that solving problem (14) is equivalent to forcing the elements in $L - k_u$ groups of $z_u$ with the smallest group-norm values to be zeros. We can easily prove that Theorem 1 is true. Here we omit the prove process.

2.2.2 Learning v

In the same manner, fix $u$ and let $v = X^T u$, thus problem (13) can be written as a similar subproblem with respect to $v$:

$$\min_v - z_v^T v, \quad \text{s.t. } \Omega_{GL_0}(v) \leq k_v. \quad (16)$$

Similarly, based on Theorem 1, we can obtain the estimator of $v$ as $P_{GL_0}(z_v)\|P_{GL_0}(z_v)\|_1$.

2.2.3 GL_0-SVD Algorithm

Finally, we propose an alternating iterative method (Algorithm 2) to solve the optimization problem (13). The time complexity of Algorithm 2 is $O(T m^2 + T n^2 + T p^2)$, where $T$ is the number of iterations.

**Algorithm 2 GL_0-SVD or SVD(GL_0, GL_1)**

Require: Matrix $X \in \mathbb{R}^{p \times n}$, $k_u$ and $k_v$; Group information

Ensure: $u, v$ and $d$

1: Initialize $v$ with $\|v\|_1 = 1$

2: repeat

3: Let $z_v = X v$

4: for $l = 1$ to $L$

5: if $\|u^{(l)}\|_2 \leq \lambda_u w_l$ then

6: $u^{(l)} = 0$

7: else

8: $u^{(l)} = z^{(l)} (1 - \lambda_u w_l / \|z^{(l)}\|_2)$

9: end if

10: end for

11: $u = u / \|u\|_2$

12: for $m = 1$ to $M$

13: Let $z = X^T u$

14: if $\|v^{(m)}\|_2 \leq \lambda_v w_m$ then

15: $v^{(m)} = 0$

16: else

17: $v^{(m)} = z^{(m)} (1 - \lambda_v w_m / \|z^{(m)}\|_2)$

18: end if

19: end for

20: $v = v / \|v\|_2$

21: $d = z^T v$

22: until $d$ convergence

23: return $u, v$ and $d$

**Note** that once the number of elements of every group equals 1 (i.e., $q_i = 1$ for $i = 1, 2, \ldots, M$), the group $L_0$-norm penalty reduces to $L_0$-norm penalty. Moreover, Algorithm 2 with a small modification can be used to solve SVD(GL_0, GL_0), which applies $L_0$-norm as the penalty for the right singular vector $v$. In addition, compared to adaptive group lasso [26], we may consider a weighted (adaptive) group $L_0$-penalty. We rewrite $\phi(z_u) = [w_1 \|z^{(1)}\|_1, \ldots, w_L \|z^{(L)}\|_1]^T$ in Eq. (15), where $w_i$ is a weight coefficient to balance different group-size and it is defined by $w_i = 1/\sqrt{q_i}$, and $q_i$ is the number of elements in group $i$.

2.3 OGL_1-SVD

In some situations, the non-overlapping group structure in group Lasso limits its applicability in practice. For example, a gene can participate in multiple pathways. Several studies have explored the overlapping group Lasso in regression tasks [16, 17]. However, structured sparse SVD with overlapping group structure remains to be solved.

Here we consider the overlapping group situation, where a variable may belong to more than one group. Suppose $u$ corresponds to the row-variables of $X$ with overlapping groups $\mathcal{G}^u = \{G_1, G_2, \ldots, G_L\}$ and $v$ corresponds to the column-variables of $X$ with overlapping groups $\mathcal{G}^v = \{G_1, G_2, \ldots, G_M\}$. In other words, $u$ and $v$ can be respectively divided into $L$ and $M$ groups, which can be represented by $u_{G_i} \in \mathbb{R}^{p \times 1}, i = 1, \ldots, L$ and $v_{G_j} \in \mathbb{R}^{n \times 1}, j = 1, \ldots, M$. We define an overlapping group Lasso (OGL_1) penalty of $u$ as follows [15, 16, 32]:

$$\Omega_{OGL_1}(u) = \min_{\mathcal{J} \subseteq \mathcal{G}^u, \text{supp}(\phi(u)) \subseteq \mathcal{J}} \sum_{i=1}^L w_i \|u_{G_i}||, \quad (17)$$

where $\text{supp}(\cdot)$ denotes the index set of non-zero elements for a given vector.

OGL_1 is a specific penalty function for structured sparsity. It can lead to the sparse solution, whose supports are unions of predefined overlapping groups of variables. Based
on the definition of \( OGL_1 \), we propose the third group-sparse SVD model as follows:

\[
\begin{align*}
\text{minimize} & \quad \|X - duv^T\|_F^2 \\
\text{subject to} & \quad \|u\|_2 \leq 1, \Omega_{OGL_1}(u) \leq c_u, \quad \|v\|_2 \leq 1, \Omega_{OGL_1}(v) \leq c_v,
\end{align*}
\]  

where \( c_u \) and \( c_v \) are two hyperparameters. We first introduce two latent vectors \( \tilde{u} \) and \( \tilde{v} \). Let \( \tilde{u}^{(l)} = u_{G_l}, l = 1, \cdots, L \) and set \( \tilde{u} = (\tilde{u}^{(1)}, \cdots, \tilde{u}^{(L)}) \), which is a column vector with size of \( \sum_{l=1}^L |G_l| \). Similarly, we can get \( \tilde{v} \) based on \( v \). In addition, we can extend the rows and columns of \( X \) of \( p \times n \) to obtain a new matrix \( \tilde{X} \) with size of \( \sum_{l=1}^L |G_l| \times \sum_{m=1}^M |G_m| \), whose row and column variables are non-overlapping. Thus, solving the problem (18) is approximately equivalent to solving a SVD(\( GL_1, GL_1 \)) for non-overlapping \( X \). We can obtain an approximate solution of (18) by using Algorithm 1. However, if a variable belongs to many different groups, it leads to a large computational burden. For example, given a protein-protein interaction (PPI) network, which contains about 13,000 genes and 250,000 edges. If we consider each edge of the PPI network as a group, then we would construct a high-dimensional matrix \( \tilde{X} \), which contains 500,000 rows.

To address this issue, we develop a method based on alternating direction method of multipliers (ADMM) [33], [34] to directly solve problem (18). Similar with Eq. (5), we first redefine problem (18) with its Lagrangian form:

\[
L(u, v) = -u^TXv + \lambda_1 \Omega_{OGL_1}(u) + \lambda_2 \Omega_{OGL_1}(v) + \eta_1 u^Tu + \eta_2 v^Tv,
\]  

where parameters \( \lambda_1 \geq 0, \lambda_2 \geq 0, \eta_1 \geq 0 \) and \( \eta_1 \geq 0 \) are Lagrange multipliers. Inspired by [35], we develop an alternating iterative algorithm to minimize it. That is, we optimize the above problem with respect to \( u \) by fixing \( v \) and vice versa. Since \( u \) and \( v \) are symmetrical in problem (19), we only need to consider a subproblem with respect to \( u \) as follows:

\[
\text{minimize} \quad -u^Tz + \lambda \Omega_{OGL_1}(u) + \eta \|u\|^2,
\]  

where \( z = Xv \). Since the overlapping Lasso penalty is a convex function [36], we can apply ADMM [33],[34] to solve the above problem (20). To obtain the learning algorithm of (20), we first introduce an auxiliary \( y \) and redefine the above problem as follows:

\[
\text{minimize} \quad -u^Ty + \lambda \sum_{l=1}^L w_l \|y^{(l)}\|_2 + \eta \|u\|^2
\]  

subject to \( y^{(l)} = u_{G_l}, l = 1, \cdots, L \).

So the augmented Lagrangian of (21) can be written as follows:

\[
L_u(u, y, \theta) = -u^Ty + \eta \|u\|^2 + \sum_{l=1}^L \theta^{(l)}(y^{(l)} - u_{G_l}) + \lambda \sum_{l=1}^L w_l \|y^{(l)}\|_2 + \frac{\rho}{2} \sum_{l=1}^L \|y^{(l)} - u_{G_l}\|^2,
\]  

where Lagrange multipliers \( \theta = [\theta^{(1)}; \cdots; \theta^{(L)}] \) and \( y = [y^{(1)}; \cdots; y^{(L)}] \) are two column vectors with \( L \) non-overlapping groups. For convenience, we first define some column-vectors \( \tilde{\theta}^{(l)}, \tilde{y}^{(l)} \) and \( \tilde{e}^{(l)} \) \((l = 1, \cdots, L)\), and they have the same size and group structures as \( u_l \) where \( \tilde{\theta}^{(l)} \) meets that \( \tilde{\theta}^{(l)}_{G_k} = \theta^{(l)} \) if \( k = l \) and \( \tilde{\theta}^{(l)}_{G_k} = 0 \) otherwise; \( \tilde{y}^{(l)} \) meets that \( \tilde{y}^{(l)}_{G_k} = y^{(l)} \) if \( k = l \) and \( \tilde{y}^{(l)}_{G_k} = 0 \) otherwise; \( \tilde{e}^{(l)} \) meets that \( \tilde{e}^{(l)}_{G_k} = 1 \) if \( k = l \) and \( \tilde{e}^{(l)}_{G_k} = 0 \) otherwise. Note that \( \tilde{\theta}^{(l)}_{G_l}, \tilde{y}^{(l)}_{G_l} \) and \( \tilde{e}^{(l)}_{G_l} \) \( l = 1, \cdots, L \) respectively represent the elements of \( k \)-th group of \( \tilde{\theta}^{(l)}, \tilde{y}^{(l)} \) and \( \tilde{e}^{(l)} \). Thus, we have \( \tilde{\theta}^{(l)^T} u_{G_l} = u_{G_l} \tilde{\theta}^{(l)} \) and \( \tilde{y}^{(l)^T} u_{G_l} = u_{G_l} \tilde{y}^{(l)} \). So we can obtain the gradient equations with respect to \( u \) in Eq. (22) as follows:

\[
\nabla_u L_\rho = 2\eta u - \sum_{l=1}^L \tilde{\theta}^{(l)} + \rho \sum_{l=1}^L L \tilde{y}^{(l)} = 0,
\]  

where “\( \bullet \)” performs element-by-element multiplication. Thus, we can obtain the update rule for \( u \) and ensure it is a unit vector:

\[
u \leftarrow \frac{\hat{u}}{\|\hat{u}\|}, \text{ where } \hat{u} = z + \sum_{l=1}^L \tilde{\theta}^{(l)} + \rho \sum_{l=1}^L \tilde{y}^{(l)}
\]  

We also obtain the subgradient equations (see [31]) with respect to \( y^{(l)} \) in Eq. (22) as follows:

\[
\nabla_{y^{(l)}} L_\rho = \lambda w_l \cdot s^{(l)} + \theta^{(l)} + \rho (y^{(l)} - u_{G_l}) = 0,
\]  

where \( l = 1, \cdots, L \), if \( y^{(l)} \neq 0 \), then \( s^{(l)} = \frac{y^{(l)}}{\|y^{(l)}\|_2} \), otherwise \( s^{(l)} \) is a vector with \( \|s^{(l)}\|_2 \leq 1 \). For convenience, let \( t^{(l)} = \rho u_{G_l} - \theta^{(l)} \), we thus develop a block coordinate descent method to learn Lagrange multipliers \( y \). Since \( y^{(l)} \) \( l = 1, \cdots, L \) are independent. Thus, \( y^{(l)} \) \( l = 1, \cdots, L \) can be updated in parallel according to the following formula:

\[
y^{(l)} \leftarrow \begin{cases} \frac{1}{\rho} \left(1 - \frac{\lambda w_l}{\|y^{(l)}\|_2}ight) t^{(l)}, & \text{if } \|t^{(l)}\|_2 > \lambda w_l, \\ 0, & \text{otherwise}. \end{cases}
\]  

Based on ADMM [34], we also obtain the update rule for \( \theta \) as follows:

\[
\theta^{(l)} \leftarrow \theta^{(l)} + \rho (y^{(l)} - u_{G_l}), \quad l = 1, \cdots, L.
\]  

Combining Eqs. (24), (26) and (27), we thus get an ADMM based method to solve problem (21) (Algorithm 3). Note that the output of Algorithm 3 is a set of selected group indexes, defined as \( T \). For example, if \( y = [y^{(1)}; y^{(2)}; y^{(3)}], \) \( y^{(1)} = 0, \) \( y^{(2)} \neq 0, \) and \( y^{(3)} \neq 0, \) then \( T = \{2, 3\}. \)

Algorithm 3 ADMM method for problem (21)

Require: \( z \in \mathbb{R}^p, \mathcal{G}, \lambda, \rho > 0 \)

1: Initialize \( \theta \) and \( y \)
2: repeat
3: Updating \( u \) with fixed \( y \) and \( \theta \) using Eq. (24)
4: Updating \( y \) with fixed \( u \) and \( \theta \) using Eq. (26)
5: Updating \( \theta \) with fixed \( u \) and \( y \) using Eq. (27)
6: until convergence
7: \( T = \{g : \|y^{(g)}\|_2 > 0, g \in \{1, \cdots, |\mathcal{G}|\}\} \)
8: return \( T \)
Algorithm 4 OGL$_1$-SVD or SVD(OGL$_1$, OGL$_1$)

Require: Matrix $X \in \mathbb{R}^{p \times n}$, $\lambda_1$, and $\lambda_o$; $G^u$ and $G^v$
Ensure: $u$, $v$ and $d$

1: Initialize $v$ with $\|v\| = 1$
2: repeat
3: Let $z_u = Xv$
4: Get the active groups $T_u$ by Algorithm 3 with $z_u$ and $G^u$ as the input
5: $\hat{u} = z_u \circ 1_{T_u}$
6: $u = \frac{\hat{u}}{\|\hat{u}\|}$
7: Let $z_v = X^Tu$
8: Get the active groups $T_v$ by Algorithm 3 with $z_v$ and $G^v$ as the input
9: $\hat{v} = z_v \circ 1_{T_v}$
10: $v = \frac{\hat{v}}{\|\hat{v}\|}$
11: $d = z_v^Tv$
12: until $d$ convergence
13: return $u$, $v$ and $d$

In summary, based on the ADMM algorithm (Algorithm 3), we adopt an alternating iterative strategy (Algorithm 4) to solve SVD(OGL$_1$, OGL$_1$). In Algorithm 4, the operation $x = z \circ 1_T$ denotes if group $l \in T$, then $x_{G_l} = z_{G_l}$, and the remaining elements of $x$ are zero.

2.4 OGL$_0$-SVD

Here we define an overlapping group $L_0$-norm penalty (OGL$_0$) of $u$ as follows:

$$\Omega_{OGL_0}(u) = \min_{\mathcal{J} \subseteq G^u, \supp(\phi(u)) \subseteq \mathcal{J}} \sum_{l=1}^{L} \mathbb{I}(\|u_G_l\| \neq 0),$$

where $\supp(\cdot)$ denotes the index set of non-zero elements for a given vector.

Based on the definition of OGL$_0$, we propose the fourth group-sparse SVD model with overlapping group $L_0$-norm penalty (OGL$_0$-SVD) as follows:

$$\min_{u,v,d} \|X - duv^T\|_F^2$$
subject to $\|u\|_2 \leq 1, \Omega_{OGL_0}(u) \leq k_u, \|v\|_2 \leq 1, \Omega_{OGL_0}(v) \leq k_v.$

(29)

Similarly, we solve the above problem by using an alternating iterative method. Fix $u$ (or $v$), we transform the original optimization problem into a projection problem with overlapping group $L_0$-norm penalty.

Fix $v$ in problem (29) and let $z = X^Tu$, thus the problem can be written into a projection problem with overlapping group $L_0$-norm penalty:

$$\min_{u,v,d} -z^Tu, \text{ s.t. } \Omega_{OGL_0}(u) \leq k_u.$$ 

(30)

To solve the above problem, we introduce $y$ and obtain the above problem in a new way:

$$\min_{u,v,d} -z^Tu, \text{ s.t. } \Omega_{OGL_0}(y) \leq k_u, y^{(l)} = u_{G_l},$$ 

(31)

where $l = 1, \cdots, L$ and $y = [y^{(1)}, \cdots, y^{(L)}]$.

The above problem contains overlapping group-sparse induced penalty with $L_0$-norm. Thus, it is difficult to solve

the exact solution of problem (31). To this end, we use an approximate method, which replaces $z^Tu$ by using $\sum_l z_G^l u_G^l$. Since $y^{(l)} = u_{G_l}$ in problem (31), we have $\sum_l z_G^l u_G^l = \sum_l z_G^l y^{(l)}$. Thus, problem (31) approximately reduces to the below problem,

$$\min_{u,l} -\|z_g^l y^{(l)}\| \text{ s.t. } \Omega_{OGL_0}(y) \leq k_u, y^{(l)} = u_{G_l}.$$ 

(32)

Since $y$ contains a non-overlapping structure, we can easily get the optimal solution of the above problem on $u$ and $y$. To sum up, we obtain an approximate solution of (30) as Theorem 2 suggests.

**Theorem 2.** The approximate solution of (30) is 
\[ \bar{p}_{OGL_0}(z) \|\bar{p}_{OGL_0}(z)\|_2 \]
\[ \|\bar{p}_{OGL_0}(z)\|_2 \]
and
\[ \|z_G^l, \text{ if } i \in \supp(\phi(z), k_u), 0, \text{ otherwise,} \]

(33)

where $l = 1, 2, \cdots, L$, $\phi(z) = [\|z_G^1\|, \cdots, \|z_G^L\|]$ and $\supp(\phi(z), k_u)$ denotes the set of indexes of the largest $k_u$ elements of $\phi(z)$.

Briefly, Theorem 2 shows that approximately solving the problem (30) is equivalent to keeping elements of $k_u$ groups with the largest $k_u$ group-norm values and the other elements are zeros.

Fix $u$ in problem (29) and let $z = X^Tu$, thus the problem (29) reduces to the following one:

$$\min_{\|u\|_2 \leq 1} -z^Tu, \text{ s.t. } \Omega_{OGL_0}(u) \leq k_u.$$ 

(34)

Similarly, based on Theorem 2, we can obtain the approximate solution of (34) as \[ \bar{p}_{OGL_0}(z) \|\bar{p}_{OGL_0}(z)\|_2 \]. Finally, we propose an alternating iterative method based on an approximate method to solve problem (29) (Algorithm 5).

Algorithm 5 OGL$_0$-SVD or SVD(OGL$_0$, OGL$_0$)

Require: Matrix $X \in \mathbb{R}^{p \times n}$, $k_u$ and $k_o$; $G^u$ and $G^v$
Ensure: $u$, $v$ and $d$

1: Initialize $v$ with $\|v\| = 1$
2: repeat
3: Let $z_u = Xv$
4: $\hat{u} = \bar{p}_{OGL_0}(z_u)$ by using Eq. (33)
5: $u = \frac{\hat{u}}{\|\hat{u}\|}$
6: Let $z_v = X^Tu$
7: $\hat{v} = \bar{p}_{OGL_0}(z_v)$ by using Eq. (33)
8: $v = \frac{\hat{v}}{\|\hat{v}\|}$
9: $d = z_v^Tv$
10: until $d$ convergence
11: return $u$, $v$ and $d$

2.5 Convergence analysis

Inspired by [27], [37], for a two-block coordinate problem, if its objective function of each subproblem is strictly convex, then there exists a unique global optimal solution for this problem. The Gauss-Seidel method can effectively solve such a two-block coordinate problem and it converges to a critical point for any given initial (see [38] and the references
vectors and get a rank this step for \( r \) to identify the next pair of sparse singular vectors. Repeat \( X \) for gene selection.

Next we discuss the convergence of \( GL_0 \)-SVD (Algorithm 2). In [35], the authors developed a class of methods based on a proximal gradient strategy to solve a broad class of nonconvex and nonsmooth problems:

\[
\begin{align*}
\text{minimize } F(u, v) &= f(u) + g(v) + H(u, v), \\
&= u^T X v,
\end{align*}
\]

where \( f(u) \) and \( g(v) \) are nonconvex and nonsmooth functions and \( H(u, v) \) is a smooth function (also see [39], [40], [41]). \( GL_0 \)-SVD model can be seen as such a problem:

\[
\begin{align*}
H(u, v) &= -u^T X v, \\
f(u) &= \begin{cases} 
0, & \text{if } \|u\| = 1, \Omega_{GL_0}(u) \leq k_u, \\
+\infty, & \text{otherwise},
\end{cases} \\
g(v) &= \begin{cases} 
0, & \text{if } \|v\| = 1, \Omega_{GL_0}(v) \leq k_v, \\
+\infty, & \text{otherwise}.
\end{cases}
\end{align*}
\]

Note \( F(u, v) = f(u) + g(v) + H(u, v) \) of \( GL_0 \)-SVD is semialgebraic and meets the KL property. Regarding the semialgebraic and KL property, please see [35], [39], [41].

Based on the Theorem 1 in [35] (also see Theorem 2 in [41]), we can obtain that \( GL_0 \)-SVD algorithm converges to a critical point.

In a word, \( GL_1 \)-SVD (Algorithm 1), \( GL_0 \)-SVD (Algorithm 2) and \( OGL_1 \)-SVD (Algorithm 4) converge to their corresponding critical points. Although \( OGL_0 \)-SVD (Algorithm 5) applies an approximate strategy, it has a good convergence in practice.

### 2.6 Group-sparse SVD for edge-guided gene selection

Given a high-dimensional data (e.g., gene expression data) and a prior network (e.g., a gene interaction network), we can consider a special edge group structure, in which each edge (e.g., a gene interaction) is considered as a group. In our study, the gene interaction network is regarded as the graph \( (V, E) \) where \( V = \{1, \cdots, p \} \) is the set of nodes (genes) and \( E \) is the set of edges (gene interactions). \( SVD(OGL_0, L_0) \) can be applied to analyze such high-dimensional gene expression data via integrating group information \( G^\gamma = E \). The estimated sparse solution is used for gene selection.

### 2.7 Learning multiple factors

To identify the next gene module, we subtract the signal of current pair of singular vectors from the input data (i.e., \( X := X - du^T v \)), and then apply SVD\((OGL_0, L_0)\) again to identify the next pair of sparse singular vectors. Repeat this step for \( r \) times, we obtained \( r \) pairs of sparse singular vectors and get a rank \( r \) approximation of matrix \( X \).

### 3 Simulation Study

In the section, we applied these group sparse SVD methods (\( GL_1 \)-SVD, \( GL_0 \)-SVD, \( OGL_1 \)-SVD and \( OGL_0 \)-SVD) to a set of simulated data and compared their performance with several sparse SVD method without using prior group information including \( L_0 \)-SVD [42], [43], \( L_1 \)-SVD [10]. We generated two types of simulation data with respect to non-overlapping group structure (GR) and overlapping group structure (OGR) respectively (Fig. 1A).

Without loss of generality, we first generated \( u \) and \( v \) for GR and OGR cases, and then generated a rank-one data matrix by using formula

\[
X = duv^T + \gamma e,
\]

where \( d = 1 \), \( e_{ij} \overset{i.i.d.}{\sim} \mathcal{N}(0,1) \) and \( \gamma \) is a nonnegative parameter to control the signal-to-noise ratio (SNR). The logarithm of SNR (logSNR) is defined by:

\[
\text{logSNR} = \log_{10} \left( \frac{\|duv\|^2_F}{\mathbb{E}(\|\gamma e\|^2_F)} \right) = \log_{10} \left( \frac{\|duv\|^2_F}{\gamma^2 np} \right),
\]

where \( \mathbb{E}(\|\gamma e\|^2_F) \) denotes the expected sum of squares of noise.

We evaluated the performance of all methods by the following measures including true positive rate (TPR), true negative rate (TNR), false positive rate (FPR), false discovery rate (FDR) and accuracy (ACC). They are defined as follows:

\[
\begin{align*}
\text{TPR} &= \frac{TP}{P}, \quad \text{TNR} = \frac{TN}{N}, \quad \text{FPR} = \frac{FP}{N}, \\
\text{FDR} &= \frac{FP}{TP+FP}, \quad \text{ACC} = \frac{TP+TN}{TP+FP+FN+TN},
\end{align*}
\]

where \( P \) denotes the number of positive samples, \( N \) denotes the number of negative samples, \( TP \) denotes the number of true positive, \( TN \) denotes the number of true negative, \( FP \) denotes the number of false positive, and \( FN \) denotes the number of false negative, respectively.

### 3.1 Non-overlapping group structure (GR)

We generated the simulated data matrix \( X \in \mathbb{R}^{p \times n} \) with \( n = 100 \) samples without groups, and \( p \) row variables from \( 50 \) groups. We first generated \( v = \text{rnorm}(n, \text{mean} = 0, \text{sd} = 1) \), which samples \( n \) elements from the standard normal distribution. Then we generated \( u = [\{u_{G1}, \ldots, u_{G50}\}] \) with \( 50 \) groups where if \( i \in \{3, 4, 13, 14, 15, 33, 34, 43, 44, 45\} \), and \( u_{G_i} = \text{sample}([-1, 1], q) \), which samples \( q \) elements from \([-1, 1]\), and \( q \) denotes the number of members of a group, otherwise \( u_{G_i} = 0 \). Finally, we obtained the simulated matrix \( X \) by using Eq. (37).

Here we first considered \( q \in \{20, 100\} \) and logSNR \( \in \{-1, -2, -2.2, -2.4, -2.6, -2.8\} \) to generate simulated data with GR. Given a logSNR, suppose \( u, v \) and \( d \) are known, then we got a \( \gamma \) by using Eq. (38). For each pair \( (q, \text{logSNR}) \), we generated 50 simulated matrices \( X_s \).

Here, we evaluated the performance of \( GL_1 \)-SVD and \( GL_0 \)-SVD with \( k_u = 10 \) (groups) in this simulated data. For comparison, we forced the identified \( u \) to contain 200 non-zero elements if \( q = 20 \) (i.e., the number of rows of \( X \) is \( p = 1000 \)) and 1000 non-zero elements if \( q = 100 \) (i.e., \( p = 5000 \)) for \( L_1 \)-SVD and \( L_0 \)-SVD by tuning their parameters. For visualization, we first tested \( GL_1 \)-SVD and \( GL_0 \)-SVD with \( k_v = 10 \) to a GR simulated \( X \) with \( p = 1000 \).
and logSNR = −2, to explain their performance and compared them with other methods (Fig. 2A). Obviously GL1-SVD and GL0-SVD can improve performance of variable selection by integrating non-overlapping group information of variables. Further, we tested our methods on more GR simulation data (Fig. 2B and TABLE 2). We also found that the performance of GL0-SVD and GL1-SVD is significantly superior to that of L0-SVD, L1-SVD in terms of different logSNRs with p = 1000 or p = 5000 (Fig. 2B). In particular, the greater the noise of simulated data, the better the performance of our methods. Furthermore, compared with GL1-SVD, GL0-SVD obtains higher singular values for different logSNRs (Fig. 2B). We also compare the different algorithms on GPU time of an ordinary personal computer, all the algorithm takes less than one second. Computational results illustrate that our models can enhance the power of variable selection by integrating group information of variables.

3.2 Overlapping group structure (OGR)

To generate OGR simulated data, we first generated v with n = 100 (samples) and its elements are from a standard normal distribution, i.e., v = rnorm(n, mean = 0, sd = 1). Then we generated u with overlapping groups. We considered an overlapping group structure for u as follows: 

$$G_1 = \{1, 2, \ldots, 2t\}, G_2 = \{t + 1, t + 2, \ldots, 3t\}, \ldots, G_{4t} = \{47t + 1, 47t + 2, \ldots, 49t\}, G_{4t+1} = \{48t + 1, 48t + 2, \ldots, 50t\}$$

where every group and its adjacent groups overlap half of the elements. Note that the dimension of u is p = 50t. If $$i \in \{3, 13, 14, 33, 43, 44\}$$ (active groups), then $$u_{G_i} = \text{sample}\{-1, 1\}$$ (2t denotes the number of members of a group), otherwise $$u_{G_i} = 0$$. We considered $$t \in \{20, 100\}$$ and logSNR ∈ {−1, −2, −2.2, −2.4, −2.6, −2.8} for generating OGR simulated data. Note that once u, v and logSNR is given, we could generate the simulated matrix $$X \in \mathbb{R}^{p \times n}$$ using Eq. (37) where d = 1. For each pair ($$q, \logSNR$$), we generated 50 simulated matrices Xs.

For visualization, we first applied OGL1-SVD and OGL0-SVD with $$k_v = 5$$ onto a simulated X with OGR, p = 1000 and logSNR = −2, to explain their performance and compared them with other methods (Fig. 2A). Obviously OGL1-SVD and OGL0-SVD can improve performance of variable selection by integrating overlapping group information of variables. Further, we tested our methods on more OGR simulation data (Fig. 2B and TABLE 2). For comparison, we forced the identified u to contain 200 non-zero elements if $$t = 20$$ (i.e., the number of rows of X is p = 1000) and 1000 non-zero elements $$t = 100$$ for L1-SVD and L0-SVD by tuning their parameters. The performance of OGL0-SVD and OGL1-SVD are significantly superior to that of L0-SVD, L1-SVD in terms of different logSNRs with p = 1000 or p = 5000 (Fig. 2B). OGL1-SVD and OGL0-SVD get similar singular values, whereas OGL1-SVD needs more time in terms of different logSNRs with p = 1000 or p = 5000 (Fig. 2B).

Finally, we also investigated the effect of our methods with different sizes of data Xs. In the simulated data, there are 10 active groups (each group contains q members) for GR cases, and there are 5 active groups for OGR cases and each group contains 2t members. We set $$q = t$$, thus we can set a common original signal of u for GR and OGR cases. Based on the defini-
tion of $u$ and $v$, we set $q \in \{20, 40, 100, 160, 200\}$ (i.e., $p \in \{1000, 2000, 5000, 8000, 10000\}$) and logSNR=-2.8 to generate $X$ using Eq. (37). For each pair $(q, \text{logSNR})$, we generated 50 simulated matrices $X$s. We applied $GL_0$-SVD, $GL_1$-SVD, $L_0$-SVD and $L_1$-SVD onto the simulated data and compared their performance in different ways (TABLE 2). In summary, the group-sparse SVD methods obtain higher TPR, TNR and ACC (lower FPR and FDR) than $L_1$-SVD and $L_0$-SVD do (TABLE 2). Naturally, the group-sparse methods spent a bit more time, and obtain lower singular value $d$.

TABLE 2: Evaluation results of $GL_0$-SVD, $GL_1$-SVD, $L_0$-SVD and $L_1$-SVD in terms of TPR, TNR, FPR, FDR, ACC, singular value $d$ and time (second). The input $X \in \mathbb{R}^{p \times 100}$ vary with different $p=1000, 2000, 5000, 8000$ and 10000. All these simulated $X$s are generated at logSNR = -2.8. Each entry is an average over 50 replications.

|           | $p = 1000$ | $p = 2000$ | $p = 5000$ | $p = 10000$ |
|-----------|------------|------------|------------|-------------|
| TPR       | 0.21       | 0.21       | 0.25       | 0.25        |
| TNR       | 0.80       | 0.80       | 0.81       | 0.81        |
| FPR       | 0.20       | 0.20       | 0.19       | 0.18        |
| FDR       | 0.79       | 0.79       | 0.78       | 0.76        |
| ACC       | 0.68       | 0.68       | 0.69       | 0.70        |
| $d$       | 337.91     | 367.61     | 242.38     | 267.73      |
| time      | 0.10       | 0.02       | 0.27       | 0.07        |

4 BIOLOGICAL APPLICATIONS

We applied our models to two gene expression data of two well-known large-scale projects.

4.1 Biological data

CGP expression data. We first downloaded a gene expression dataset from the Cancer Genome Project (CGP) [44] with 13321 genes across 641 cell lines (samples). The 641 cell lines are derived from different tissues and cancer types.

TCGA expression data. We also obtained twelve cancer gene expression datasets for twelve cancer types across about 4000 cancer samples (http://www.cs.utoronto.ca/~yueli/PanMiRa.html), which is downloaded from TCGA database (http://cancergenome.nih.gov/). The twelve cancer types consist of Bladder urothelial carcinoma (BLCA, 134 samples), Breast invasive carcinoma (BRCA, 847 samples), Colon and rectum carcinoma (CRC, 550 samples), Head and neck squamous-cell carcinoma (HNSC, 303 samples), Kidney renal clear-cell carcinoma (KIRK, 474 samples), Brain lower grade glioma (LGG, 179 samples), Lung adenocarcinoma (LUAD, 350 samples), Lung squamous-cell carcinoma (LUSC, 315 samples), Prostate adenocarcinoma (PRAD, 170 samples), Skin cutaneous melanoma (SKCM, 234 samples), Thyroid carcinoma (THCA, 224 samples), Uterine corpus endometrioid carcinoma (UCEC, 478 samples). We normalized each gene expression dataset of a given cancer type using R function scale. Furthermore, we also downloaded the corresponding clinical data of the above 12 cancer types from Firehose (http://firebrowse.org/).

KEGG pathway data. To integrate the pathway group information with SVD($GL_0$, $L_0$), we also downloaded the KEGG [13] gene pathways from the Molecular Signatures Database (MSigDB) [45]. We considered a KEGG pathway as a group and removed all the KEGG pathways with $> 100$ genes. Finally, we obtained 151 KEGG pathways across 2778 genes by only considering the intersection genes between the CGP gene expression and KEGG pathways data. On average, a gene pathway contains about 40 genes.

PPI network. We also downloaded a protein-protein interaction (PPI) network from Pathway Commons (http://www.pathwaycommons.org/). In our application, we also considered the edge set of the PPI network as overlapping groups $G^u$, i.e., a gene (protein) interaction edge represents a group.

Data collection. Finally, we generated three biological datasets to assess our models:

- **Dataset 1**: “CGP + PPI”. This dataset was obtained by combining CGP gene expression and PPI network data with 13321 genes, 641 samples and 262462 interactions.
- **Dataset 2**: “TCGA + PPI”. This dataset was obtained by combining TCGA gene expression and PPI network data. For each TCGA cancer type, we obtain the expression of 10399 genes and a PPI network of 10399 genes and 257039 interactions.
- **Dataset 3**: “CGP + KEGG”. This dataset was obtained by combining CGP gene expression and KEGG pathway data including gene expression of 641 samples and 151 KEGG pathways across 2778 genes.

Biological analysis of gene modules: To assess whether the identified modules have significant biological functions, we employed the bioinformatics tool DAVID (https://david.ncifcrf.gov/) [46] to perform gene enrichment analysis with GO biological processes (BP) and KEGG pathways. The terms with Benjamin corrected $p$-value $< 0.05$ are considered as significant ones.

For each cancer type, we first used SVD($OGL_0$, $L_0$) to find a gene module with similar expressions (Fig. 1). To analyze the clinical relevance of each module in a given cancer type, we ran a multivariate multivariate Cox proportional hazard model to obtain the prognostic scores for the patients of this cancer type. It was implemented by using predict function in the R package ‘survival’ with type = “lp”.

Then we divided the patients of this cancer type into low-risk and high-risk groups based on the prognostic scores.
Finally, we assessed the overall survival difference between the two groups using log-rank test and drew a Kaplan-Meier (KM) curve for visualization.

### 4.2 Application to CGP data with a PPI network

We applied SVD($OGL_0$, $L_0$) to the “CGP+ PPI” dataset consisting of the CGP gene expression and PPI network data. We set $k_a = 100$ to extract gene interactions, $k_v = 50$ to select 50 samples. Here we only focused on the first identified gene module, which contains 56 genes and 272 interactions. We first found that the subnetwork of identified module in the prior PPI network is dense (Fig. 3A). As we expected, the genes from this module contains a large number of linked genes in the prior PPI network (the degree sum of these genes is 6321). The identified 50 samples of this module are significantly related with some blood related cancers including AML (6 of 16), B cell leukemia (7 of 7), B cell lymphoma (7 of 10), Burkitt lymphoma (11 of 11), lymphoblastic leukemia (6 of 11), lymphoid neoplasm (5 of 11). On the other hand, these samples are specifically related with blood tissue (50 of 100). Moreover, this module is enriched in 77 different GO biological processes and 11 KEGG pathways, most of which are immune and blood related pathways including immune response-activating cell surface receptor signaling pathway, immune response-regulating cell surface receptor signaling pathway and immune response-activating signal transduction (Fig. 3B).

Finally, we also applied $L_0$-SVD to extract a gene module with the same number of genes and samples for comparison. However, this module only contains 35 edges and most of the identified genes are isolated. We repeatedly sample 80 percent of genes and samples from the original matrix (i.e., the CGP data) for 10 times. For each sampled data matrices, we applied $OGL_0$-SVD and $L_0$-SVD to identify a gene module with the same settings above. We obtained an average of 382.8 (edges) for $OGL_0$-SVD, but an average of 35.6 (sum of edges) for $L_0$-SVD. These results indicate that $OGL_0$-SVD can identify gene modules with more connected edges in the prior PPI network by integrating edge-group structure.

### Table 3: Top gene modules identified by SVD($OGL_0$, $L_0$) and $L_0$-SVD from 12 different cancer-type gene expression datasets. #edge indicates the number of edges of all module genes, and #degree indicates the sum of degree of all module genes in the prior PPI network.

| Data       | #gene | #degree$^a$ | #edge$^a$ | #degree$^b$ | #edge$^b$ |
|------------|-------|-------------|-----------|-------------|-----------|
| BLCA-module| 66    | 22          | 135       | 1956        | 1800      |
| SKCM-module| 65    | 82          | 202       | 8348        | 2800      |
| LUAD-module| 75    | 25          | 173       | 1434        | 2948      |
| LUSC-module| 67    | 19          | 170       | 793         | 2813      |
| BRCA-module| 79    | 83          | 162       | 8493        | 5211      |
| HNSC-module| 84    | 23          | 159       | 1291        | 3596      |
| THCA-module| 50    | 32          | 382       | 6871        | 4099      |
| KIRC-module| 66    | 29          | 194       | 4848        | 10497     |
| LGG-module | 59    | 198         | 139       | 6048        | 7549      |
| PRAD-module| 68    | 69          | 178       | 7463        | 9210      |
| UCEC-module| 55    | 22          | 251       | 4512        | 14144     |
| CRC-module | 49    | 1           | 384       | 1177        | 16620     |

$^a$ corresponds to $L_0$-SVD.

$^b$ corresponds to SVD($OGL_0$, $L_0$).

### 4.3 Application to TCGA data with a PPI network

We next applied SVD($OGL_0$, $L_0$) to analyze the gene expression data of 12 different cancers including BLCA, BLCA, LUAD, LUSC, BRCA, HNSC, THCA, KIRC, LGG, UCEC, and CRC. Note that SVD($OGL_0$, $L_0$) was tested independently for each cancer to identify a important gene module across a set of tumor patients. SVD($OGL_0$, $L_0$) uses overlapping group $L_0$-norm penalty for gene interaction selection with $k_a = 100$ (i.e. identify 100 edge-groups) and $L_0$-norm penalty for sample-variable selection with $k_v = 50$ (i.e. 50 samples). Thus, we obtained 12 gene modules for 12 cancer types with about 65 genes on average (TABLE 3). For convenience, we defined the identified module of ‘X’ cancer-type as X-module (TABLE 3). For comparison, we also enforced $L_0$-SVD to identify a gene module with the...
As we expected, all modules identified by SVD(OGL₀, L₀) contain more edges than the ones identified by L₀-SVD. Interestingly, we also find that most of the module genes identified by SVD(OGL₀, L₀) are with higher degree in the prior PPI network than those by L₀-SVD without using prior information (TABLE 3), indicating that these genes with higher degree tend to be related with cancer. In addition, we observed a overlapping pattern, containing four cancer-specific modules (BLCA, SKCM, LUAD and LUSC cancers which are related with skin/epidermis tissue) sharing many common genes. Subsequent functional analysis shows these common genes are enriched in some important immune-related pathways (Fig. 4A). Taken together of 12 cancer-type gene modules, we find 412 enriched GO biological processes (GO BPs, or GOBPs) and 48 KEGG pathways with Benjamini-Hochberg adjusted p-value < 0.05. Interestingly, some important cancer-related KEGG pathways are discovered in both BRCA-module and HNSC-module including ECM-receptor interaction (with p-value = 4.3e-18 in BRCA-module and p-value = 2.8e-22 in HNSC-module), Focal adhesion (with p-value = 3.2e-16 in BRCA-module and p-value = 3.5e-22 in HNSC-module), and Pathways in cancer (with p-value = 5.0e-03 in BRCA-module and p-value = 2.3e-04 in HNSC-module) (Fig. 4A). Some important immune related pathways in both BLCA, SKCM, LUAD and LUSC-module
TABLE 4: Summary of the top ten gene modules identified by SVD($OGL_0$, $OGL_0$) in the “CGP+KEGG” dataset. #Gene and #Sample denote the number of genes and samples in each module; KEGG Term denotes the enriched KEGG pathways; Tumor and Tissue Type denote the enriched tumor and tissue types respectively.

| ID | #Gene | #Sample | KEGG Term                                                                 | Tumor Type                                                                 | Tissue Type |
|----|-------|---------|---------------------------------------------------------------------------|---------------------------------------------------------------------------|-------------|
| 1  | 221   | 100     | Ribosome; Primary immunodeficiency; B cell receptor signaling pathway; Intestinal immune network for iga production; Asthma | AML, B cell lymphoma; B cell lymphoma; Burkitt lymphoma; lymphoblastic leukemia; Lymphoid neoplasm other | Blood       |
| 2  | 99    | 100     | DNA replication; Homologous recombination; Base excision repair; Mismatch repair; Nucleotide excision repair | B cell leukemia; Lung; Small cell carcinoma; Lymphoblastic leukemia; Lymphoblastic T cell leukemia | Blood; Lung |
| 3  | 200   | 100     | Ecm receptor interaction; Glycosaminoglycan degradation; Primary immunodeficiency; Glycosaminoglycan biosynthesis heparan sulfate; Arrhythmogenic right ventricular cardiomyopathy arvc | Glioma; Lymphoblastic leukemia; Lymphoblastic T cell leukemia; Osteosarcoma | Blood; Bone; Soft-tissue |
| 4  | 216   | 100     | Pathogenic escherichia coli infection; Lysine degradation; Notch signaling pathway; Adherens junction; Rna degradation | AML, B cell leukemia; B cell lymphoma; Burkitt lymphoma; Lymphoblastic leukemia; Lymphoblastic T cell leukemia; Lymphoid neoplasm other | Blood       |
| 5  | 83    | 100     | Allograft rejection; Type 1 diabetes mellitus; Asthma; Graft versus host disease; Intestinal immune network for iga production | AML, B cell leukemia; B cell lymphoma; Burkitt lymphoma; Hodgkin lymphoma; Lymphoblastic T cell leukemia; Lymphoid neoplasm other; Myeloma | Blood       |
| 6  | 166   | 100     | Complement and coagulation cascades; Phenylalanine metabolism; Primary bile acid biosynthesis; Ppar signaling pathway; Steroid biosynthesis | Large intestine; Liver | GI tract |
| 7  | 253   | 100     | Hematopoietic cell lineage; Acute myeloid leukemia; Fc epsilon ri signaling pathway; Fc gamma r mediated phagocytosis; Primary immunodeficiency | AML, B cell lymphoma; Lymphoblastic leukemia; lymphoblastic T cell leukemia; lymphoid neoplasm other; oesophageus; upper aerodigestive tract | Blood; Upper aerodigestive |
| 8  | 141   | 100     | Glutathione metabolism; Metabolism of xenobiotics by cytochrome p450; Steroid hormone biosynthesis; Ascorbate and aldarate metabolism; Arachidonic acid metabolism | Liver, lung: NSCLC: adenocarcinoma; Lung: NSCLC: squamous cell carcinoma; Oesophagus | GI tract; Lung; Upper aerodigestive |
| 9  | 132   | 100     | Glycosaminoglycan biosynthesis chondroitin sulfate; Mismatch repair; N glycan biosynthesis; Glycosylphosphatidylinositol GPI anchor biosynthesis; Vibriob cholelrae infection | Glioma; Lung; small cell carcinoma; Lymphoblastic leukemia; Neuroblastoma | CNS |
| 10 | 193   | 100     | B cell receptor signaling pathway; Circadian rhythm mammal; Primary immunodeficiency; Fc gamma r mediated phagocytosis; Fc epsilon ri signaling pathway | AML, B cell lymphoma; Glioma; Lung; small cell carcinoma; Lymphoblastic leukemia; Lymphoid neoplasm other | Blood       |

including Cell adhesion molecules, Hematopoietic cell lineage, Natural killer cell mediated cytotoxicity, Primary immunodeficiency, and T cell receptor signaling pathway. This is consistent with previous studies that lung cancer is related to immune response [47]. In addition, several previous KEGG pathways are found in THCA-module including Oxidative phosphorylation ($p = 1.1e-59$), Parkinson’s disease ($p = 7.9e-52$), Alzheimer’s disease ($p = 5.8e-48$) and Huntington’s disease ($p = 1.8e-46$). For clarity, the top five enriched GO BPs of each cancer module are shown (Fig. 4A). Several GO BPs are common in many cancers, while most of them are specific to certain cancers (Fig. 4A).

Moreover, we find that those patients of 11 cancers (BLCA, BLCA, LUAD, LUSC, BRCA, HNSC, THCA, KIRC, LGG, UCEC, CRC) can be effectively separated into two risk-groups, which are significantly different with survival outcome ($p$-value < 0.05, log-rank test) (Fig. 4B). All the results showed that our method by integrating interaction structure could identify more biologically relevant gene modules, and improve their biological interpretations.

4.4 Application to CGP data with KEGG Pathways

We applied SVD($OGL_0$, $OGL_0$) to the “CGP + KEGG”. We set $k_u = 5$ (i.e., five pathways), $k_v = 100$ (i.e. 100 samples) in SVD($OGL_0$, $OGL_0$) for convenience. Since the identified top ten pair of singular vectors explained more than 60% of the variance, we focused on the top ten pair singular vectors to extract ten gene functional modules (TABLE 4).

For each gene module, we computed the overlap significance between its sample set and each tissue or cancer class using a hypergeometric test, implemented via R function phyper. Those cancer types with a few samples ($\leq 5$) are ignored. We find that each module is significantly related to at least one cancer or tissue type, indicating that they indeed be biologically relevant.

It is worth noting that some cancer/tissue subtype-specific KEGG pathways are discovered. We summarized all key messages of the identified modules in TABLE 4. For example, we find that all the samples of module 1 belong to blood tissue, and most of samples of module 1 are significantly enriched in some lymphoid-related cancers. Interestingly, the corresponding five blood-specific KEGG pathways (including ribosome, primary immunodeficiency, b cell receptor signaling pathway, intestinal immune network for iga production, and asthma) are related to lymphoma. Another example module 6 is specifically related to large intestine, liver cancer and GI tract tissue with important KEGG pathways including complement and coagulation cascades, pheny-
5 Discussion and conclusion

Inferring blocking patterns from high-dimensional biological data is still a central challenge in computational biology. On one hand, we aim to identify some gene subsets, which are co-expressed across some samples in the corresponding gene expression data. On the other hand, we expect that these identified gene sets belong to some biologically meaningful groups such as functional pathways. To this end, we propose group-sparse SVD models to identify gene blocking patterns (modules) via integrating gene expression data and prior gene knowledge.

We note that the concept of group sparse has also been introduced into some related models such as the non-negative matrix factorization (NMF) model [48]. To our knowledge, non-overlapping group Lasso was applied as the penalty onto NMF. Obviously, our models are very different from the group-sparse NMF model. First, we used a more direct non-overlapping sparse penalty with L₀-norm penalty into the SVD model and propose a group sparse SVD with group L₀-norm penalty (GL₀-SVD). More importantly, we prove the convergence of the GL₀-SVD algorithm. Second, since the non-overlapping group structure in group Lasso or group L₀-norm penalty limits their applicability in practice. Several works have studied the (overlapping) group Lasso in regression tasks. However, little work focuses on developing structured sparse matrix factorization models with overlapping group structure. We propose two overlapping group SVD models (OGL₁-SVD and OGL₀-SVD), and discuss their convergence issue. Computational results of high-dimensional gene expression data show that our methods could identify more biologically relevant gene modules and improve their biological interpretations than the state-of-the-art sparse SVD methods. Moreover, we expect that our methods could be applied to more high-dimensional biological data such as single cell RNA-seq data, and high-dimensional data in other fields. In the future, it will be valuable to extend current concept into structured sparse tensor factorization model for multi-way analysis of multi-source data.

Acknowledgement

Shihua Zhang and Juan Liu are the corresponding authors of this paper. Wenwen Min would like to thank the support of the National Key Research and Development Program of China [2017YFC0908405]; CAS Frontier Science Research Key Project for Top Young Scientist [QYZDB-SSW-SYS008].

References

[1] A. P. Singh and G. J. Gordon, “A unified view of matrix factorization models,” in Machine Learning and Knowledge Discovery in Databases (ECML/PKDD), 2008, pp. 358–373.

[2] O. Alter, P. O. Brown, and D. Botstein, “Singular value decomposition for genome-wide expression data processing and modeling,” Proceedings of the National Academy of Sciences, vol. 97, no. 18, pp. 10101–10106, 2000.

[3] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” IEEE Transactions on Signal Processing, vol. 54, no. 11, pp. 4311–4322, 2006.

[4] X. Zhou, J. He, G. Huang, and Y. Zhang, “SVD-based incremental approaches for recommender systems,” Journal of Computer and System Sciences, vol. 81, no. 4, pp. 717–733, 2015.

[5] I. Sohn, J. Kim, S.-H. Jung, and C. Park, “Gradient lasso for cox proportional hazards model,” Bioinformatics, vol. 25, no. 14, pp. 1775–1781, 2009.

[6] H. Fang, C. Huang, H. Zhao, and M. Deng, “Cclasso: correlation inference for compositional data through lasso,” Bioinformatics, vol. 31, no. 19, pp. 3172–3180, 2015.

[7] K. Greenlaw, E. Szefer, J. Graham, M. Lesperance, F. S. Nathoo, and A. D. N. Initiative, “A bayesian group sparse multi-task regression model for imaging genetics,” Bioinformatics, vol. 33, no. 16, pp. 2513–2522, 2017.

[8] J. Shi, N. Wang, Y. Xia, D.-Y. Yeung, I. King, and J. Jia, “SCMF: Sparse covariance matrix factorization for collaborative filtering,” in IJCAI, 2013, pp. 2705–2711.

[9] L. Jing, P. Wang, and L. Yang, “Sparse probabilistic matrix factorization by laplace distribution for collaborative filtering,” in IJCAI, 2015, pp. 1771–1777.

[10] M. Lee, H. Shen, J. Z. Huang, and J. Marron, “Biclustering via sparse singular value decomposition,” Biometrics, vol. 66, no. 4, pp. 1087–1095, 2010.

[11] M. Sill, S. Kaiser, A. Benner, and A. Kopp-Schneider, “Robust biclustering by sparse singular value decomposition incorporating stability selection,” Bioinformatics, vol. 27, no. 15, pp. 2089–2097, 2011.

[12] D. Yang, Z. Ma, and A. Buja, “A sparse singular value decomposition method for high-dimensional data,” Journal of Computational and Graphical Statistics, vol. 23, no. 4, pp. 923–942, 2014.

[13] M. Kanehisa and S. Goto, “KEGG: Kyoto encyclopedia of genes and genomes,” Nucleic Acids Research, vol. 28, no. 1, pp. 27–30, 2000.

[14] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 68, no. 1, pp. 49–67, 2006.

[15] L. Jacob, G. Obozinski, and J.-P. Vert, “Group lasso with overlap and graph lasso,” in ICML. ACM, 2009, pp. 433–440.

[16] G. Obozinski, L. Jacob, and J.-P. Vert, “Group lasso with overlaps: the latent group lasso approach,” arXiv preprint arXiv:1110.0413, 2011.

[17] L. Yuan, J. Liu, and J. Ye, “Efficient methods for overlapping group lasso,” in Advances in Neural Information Processing Systems, 2011, pp. 352–360.

[18] R. Tibshirani, “Regression shrinkage and selection via the lasso,” Journal of the Royal Statistical Society. Series B (Methodological), pp. 267–288, 1996.

[19] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 67, no. 2, pp. 301–320, 2005.

[20] D. M. Witten, R. Tibshirani, and T. Hastie, “A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis,” Biostatistics, vol. 10, no. 3, pp. 515–534, 2009.

[21] H. Zou, T. Hastie, and R. Tibshirani, “Sparse principal component analysis,” Journal of Computational and Graphical Statistics, vol. 15, no. 2, pp. 265–286, 2006.

[22] L. Clemmensen, T. Hastie, D. Witten, and B. Ersbøll, “Sparse discriminant analysis,” Technometrics, vol. 53, no. 4, pp. 406–413, 2011.

[23] D. M. Witten and R. Tibshirani, “A framework for feature selection in clustering,” Journal of the American Statistical Association, vol. 105, no. 490, pp. 713–726, 2010.

[24] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, “A sparse-group lasso,” Journal of Computational & Graphical Statistics, vol. 22, no. 2, pp. 231–245, 2013.
25. R. Tibshirani, “Regression shrinkage and selection via the lasso: a retrospective,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 3, pp. 273–282, 2011.

26. H. Wang and C. Leng, “A note on adaptive group lasso,” *Computational Statistics & Data Analysis*, vol. 52, no. 12, pp. 5277–5286, 2008.

27. P. Tseng, “Convergence of a block coordinate descent method for nondifferentiable minimization,” *Journal of Optimization Theory & Applications*, vol. 109, no. 3, pp. 475–494, 2001.

28. J. Friedman, T. Hastie, H. Höfling, R. Tibshirani et al., “Pathwise coordinate optimization,” *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.

29. J. Friedman, T. Hastie, and R. Tibshirani, “Regularization paths for generalized linear models via coordinate descent,” *Journal of Statistical Software*, vol. 33, no. 1, p. 1, 2010.

30. N. Simon, J. Friedman, and T. Hastie, “A blockwise descent algorithm for group-penalized multivariate and multinomial regression,” arXiv preprint arXiv:1311.6529, 2013.

31. M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.

32. J. Huang, T. Zhang, and D. Metaxas, “Learning with structured sparsity,” *Journal of Machine Learning Research*, vol. 12, no. Nov, pp. 3371–3412, 2011.

33. Z. Qin and D. Goldfarb, “Structured sparsity via alternating direction methods,” *Journal of Machine Learning Research*, vol. 13, no. May, pp. 1433–1468, 2012.

34. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

35. J. Bolte, S. Sabach, and M. Teboulle, “Proximal alternating linearized minimization for nonconvex and nonsmooth problems,” *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.

36. S. Mosci, S. Villa, A. Verri, and L. Rosasco, “A primal-dual algorithm for group sparse regularization with overlapping groups,” in *Advances in Neural Information Processing Systems*, 2010, pp. 2604–2612.

37. L. Grippo and M. Sciandrone, “On the convergence of the block nonlinear gauss–seidel method under convex constraints,” *Operations Research Letters*, vol. 26, no. 3, pp. 127–136, 2000.

38. M. Razaviyayn, M. Hong, and Z.-Q. Luo, “A unified convergence analysis of block successive minimization methods for nonsmooth optimization,” *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

39. T. Pock and S. Sabach, “Inertial proximal alternating linearized minimization (ipalm) for nonconvex and nonsmooth problems,” *SIAM Journal on Imaging Sciences*, vol. 9, no. 4, pp. 1756–1787, 2016.

40. F. Yang, Y. Shen, and Z. S. Liu, “The proximal alternating iterative hard thresholding method for $L_0$ minimization, with complexity $O(1/\sqrt{k})$,” *Journal of Computational and Applied Mathematics*, vol. 311, pp. 115–129, 2017.

41. Y. Xu and W. Yin, “A globally convergent algorithm for nonconvex optimization based on block coordinate update,” *Journal of Scientific Computing*, pp. 1–35, 2017.

42. M. Asteris, A. Kyrillidis, O. Koyejo, and R. Poldrack, “A simple and provable algorithm for sparse diagonal cca,” in *ICML*. ACM, 2016, pp. 1148–1157.

43. W. Min, J. Liu, and S. Zhang, “A novel two-stage method for identifying miRNA-gene regulatory modules in breast cancer,” in *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2015, pp. 151–156.

44. M. J. Garnett, E. J. Edelman, C. D. Greenman, A. Dastur, K. W. Lau, P. Greninger, I. R. Thompson, X. Luo, J. Soares et al., “Systematic identification of genomic markers of drug sensitivity in cancer cells,” *Nature*, vol. 483, no. 7391, pp. 570–575, 2012.

45. A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander et al., “Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles,” *Proceedings of the National Academy of Sciences*, vol. 102, no. 43, pp. 15 545–15 550, 2005.

46. D. W. Huang, B. T. Sherman, and R. A. Lempicki, “Systematic and integrative analysis of large gene lists using david bioinformatics resources,” *Nature Protocol*, vol. 4, no. 1, pp. 44–57, 2009.

47. P. Roepman, J. Jassem, E. F. Smit, T. Muley, J. Niklinski, T. van de Velde, A. T. Witteveen, W. Rzyman, A. Floore, S. Burgers et al., “An immune response enriched 72-gene prognostic profile for early-stage non–small-cell lung cancer,” *Clinical Cancer Research*, vol. 15, no. 1, pp. 284–290, 2009.

48. J. Kim, R. D. C. Monteiro, and H. Park, “Group sparsity in nonnegative matrix factorization,” in *SIAM International Conference on Data Mining*, 2012, pp. 851–862.