TYPE Original Research
PUBLISHED 09 June 2023
DOI 10.3389/fgene.2023.1179439

Clustering single-cell multimodal omics data with jrSiCKLSNMF

Dorothy Ellis, Arkaprava Roy and Susmita Datta*

Department of Biostatistics, University of Florida, Gainesville, FL, United States

OPEN ACCESS

EDITED BY
Himel Mallick,
Cornell University, United States

REVIEWED BY
Suvo Chatterjee,
Indiana University, United States
Jihwan Oh,
Merck, United States
Piyali Basak,
Merck, United States

*CORRESPONDENCE
Susmita Datta,
✉ susmita.datta@ufl.edu

RECEIVED 04 March 2023
ACCEPTED 23 May 2023
PUBLISHED 09 June 2023

CITATION
Ellis D, Roy A and Datta S (2023), Clustering single-cell multimodal omics data with jrSiCKLSNMF.
Front. Genet. 14:1179439.
doi: 10.3389/fgene.2023.1179439

Introduction: The development of multimodal single-cell omics methods has enabled the collection of data across different omics modalities from the same set of single cells. Each omics modality provides unique information about cell type and function, so the ability to integrate data from different modalities can provide deeper insights into cellular functions. Often, single-cell omics data can prove challenging to model because of high dimensionality, sparsity, and technical noise.

Methods: We propose a novel multimodal data analysis method called **j**oint graph-**r**egularized **Si**ngle-**C**ell **K**ullback-**L**eibler **S**parse **N**on-negative **M**atrix **F**actorization (jrSiCKLSNMF, pronounced "junior sickles NMF") that extracts latent factors shared across omics modalities within the same set of single cells.

Results: We compare our clustering algorithm to several existing methods on four sets of data simulated from third party software. We also apply our algorithm to a real set of cell line data.

Discussion: We show overwhelmingly better clustering performance than several existing methods on the simulated data. On a real multimodal omics dataset, we also find our method to produce scientifically accurate clustering results.

KEYWORDS

graph regularization, KL divergence, multimodal omics, multiplicative updates, scATAC-seq, scRNA-seq, sparsity

1 Introduction

Next-generation sequencing (NGS) technologies have enabled the extraction of large amounts of cellular information from biological tissues. These data are collectively known as omics and include metabolomics, transcriptomics, epigenomics, proteomics, and metagenomics. Within the last decade, the integration of multiple omics profiles has led to advances in precision medicine and the identification of underlying disease mechanisms (Reel et al., 2021). Furthermore, advances in single-cell sequencing technologies have enabled the extraction of omic profiles at the resolution of a single-cell (Tang et al., 2009; Buenrostro et al., 2015). Within the last half-decade, the extraction of multiple omics profiles from the same set of single cells has become possible (Stoeckius et al., 2017; Chen et al., 2019; Ma et al., 2020; Swanson et al., 2021). Lee et al. (2020) and Ogbeide et al. (2022) detail a wide variety of technologies currently available to collect data from multiple omics modalities from the same set of cells. The genome, transcriptome, and proteome are connected through the central dogma of molecular biology: DNA is transcribed to RNA, which is in turn translated to proteins (Li and Biggin, 2015). Costa Dos Santos et al. (2021) discuss an extension to the central dogma; in this updated version, the metabolome drives the flow of omics information through the cell. This updated version also includes the epigenome, which are biochemical modifications to DNA that affect structure and regulation of the genome (Park et al., 2016). These include histone modifications, chromatin accessibility, and DNA methylation.

While omics data collected from the same cell are all inter-related, each modality still provides some unique information about that cell. Thus, the integration of these data across omics modalities can enable deeper insights into cellular functions than the analysis of each modality in isolation. Among these deeper insights is improved cell-type clustering. Expression of omics data varies among cell types, and this cellular heterogeneity is not captured in bulk data (Ellis et al., 2021). Accurately clustering cells can, for example, enable insights into and analysis of cell-type-specific responses to disease. Additionally, some omics modalities are more informative for differentiating between certain cell types than others; for example, in Hao et al. (2021), CD4^{+} cells and CD8^{+} cells had similar RNA expression profiles but had different protein expression profiles. Currently, there are only a few methods available to integrate count data across multiple single-cell omics modalities. Many of these methods require $\log(x + 1)$ normalization methods that can introduce bias into the transformed data by exaggerating the differences between 0 and low count observations (Townes et al., 2019; Elyanow et al., 2020). Most other methods also choose a fixed number of highly variable features on which to perform clustering; however, these highly variable features may not necessarily be the most informative for cell clustering and can leave out important information (Townes et al., 2019). Hence, we develop **j**oint graph-**r**egularized **Si**ngle-**C**ell **K**ullback-**L**eibler **S**parse **N**on-negative **M**atrix **F**actorization (jrSiCKLSNMF, pronounced "junior sickles NMF") for the count-valued omics data within each modality while integrating across omics information in order to offer more accurate cell-type clustering. Through our method, we aim first to extract latent factors that are relevant to cell-type clustering and consequently enable convenient clustering on these latent factors. Secondly, we allow the visualization of cell type clusters by leveraging the data compression abilities of NMF. Non-negative matrix factorization has been used for various modern applications, including latent factor extraction, data compression, and clustering. Additionally, many NMF methods have already been applied to the analysis of omics data. These include Multi-NMF (Liu et al., 2013; Wang et al., 2015; Rappoport and Shamir, 2018), integrative NMF (Chalise and Fridley, 2017; Liu et al., 2020), and jNMF (Greene and Cunningham, 2009; Akata et al., 2011; Wang et al., 2015; Dai et al., 2020) for multi-omics data; NMF (Lee and Seung, 1999) and graph-regularized NMF (Cai et al., 2008; 2011; Elyanow et al., 2020) for single-modality omics data; SC-JNMF (Shiga et al., 2021) for different quantifications of scRNA-seq data measured on the same set of cells; and scAI (Jin et al., 2020), which, like our method, is for multimodal single-cell omics data. Some of these methods, including jNMF, Multi-NMF, and graph-regularized NMF, arose first from the fields of image processing and document classification.

Although our method can theoretically integrate any number of modalities of single-cell count-valued data collected from the same set of cells or any number of bulk assays collected from the same individual, we primarily focus on integrating single-cell RNA-sequencing (scRNA-seq) and single-cell assay for transposase-accessible chromatin using sequencing (scATAC-seq) data from the same set of single cells. Methods for collecting these data include sci-CAR (Cao et al., 2018), SNARE-seq (Chen et al., 2019), SHARE-seq (Ma et al., 2020). scRNA-seq allows for the detection and analysis of messenger RNAs (mRNAs) at a single-cell resolution. These data consist of count matrices where each column corresponds to a cell and each row to a gene (Haque et al., 2017). scATAC-seq identifies accessible regions (peaks) within the chromatin of a single-cell; the data consist of matrices of counts of nucleosome free region (NFR) fragments, where each column corresponds to a cell and each row corresponds to a given range of base pairs (Yan et al., 2020). Due to several challenges such as batch effects, technical noise, and sparsity, these data require extensive quality control, normalization, and batch effect correction before downstream analyses, including cell clustering and annotation, network analysis, and differential expression analysis, can proceed (Yan et al., 2020; Ellis et al., 2021).

In Section 2, we discuss the motivation for our model in detail, provide the loss function, and discuss the implementation. We discuss both the initialization of our matrix product approximation as well as the optimization of this product. "Joint" NMF methods share one of either feature matrix $\mathbf{W}$ or observation matrix $\mathbf{H}$ across different modalities of data or different individuals. For jrSiCKLSNMF, we share $\mathbf{H}$ across all omics modalities and treat it as a latent cell-specific factor matrix. To adjust for differences in quality and quantity of information across modalities, we use graph regularization on each modality v's $\mathbf{W}^{v}$ matrix. Elyanow et al. (2020), whose research also served as a primary motivation for this work, detail this approach of using graph regularization for the feature matrix $\mathbf{W}$ for single-modality scRNA-seq data. Because both modalities of these data are inherently sparse, we also include a sparsity constraint on $\mathbf{H}$ or, alternatively, a unit norm constraint on the L2 norm of the rows of $\mathbf{H}$ as detailed for single-modality data in Le Roux et al. (2015). Because we are integrating different types of count data, we use the Poisson Kullback-Leibler (KL) divergence across all modalities.

In Section 3, we compare our method with competing methods on simulated data. We also provide a real data example. While there are a multitude of methods currently available for integrating bulk omics data across modalities and also methods to integrate data from different single-cell populations measured on the same individual (Krassowski et al., 2020; Subramanian et al., 2020; Miao et al., 2021), there are only a few approaches for the integration of measurements from the same set of single cells. Some of these methods include Seurat v. 4 (Hao et al., 2021), BREM-SC (Wang et al., 2020), CiteFuse (Kim et al., 2020), scAI, and MOFA+ (Argelaguet et al., 2020). We briefly discuss these existing methods in Section 3.3 before comparing them to jrSiCKLSNMF in Section 3.4. Of these, only our method and BREM-SC take into account the count nature of both the scATAC-seq and scRNA-seq modalities; all other methods require some form of log normalization on the data. Coincidentally, BREM-SC, which assumes the data follow a Dirichlet-Multinomial distribution, was, after the four variations of jrSiCKLSNMF, the fifth highest performing method on simulated data with no introduced noise. Finally, in Section 4, we discuss potential extensions of jrSiCKLSNMF as well as its limitations.

2 Materials and methods

In general, all non-negative matrix factorization (NMF) models attempt to find a reduced rank latent representation, where the number of latent factors often is pre-specified (Lee and Seung, 1999). Among various uses of NMF, our method is, primarily, designed for clustering cell types by first extracting latent factors shared across omics modalities and then clustering these latent factors using any clustering method. We perform all analyses and coding in R (R Core

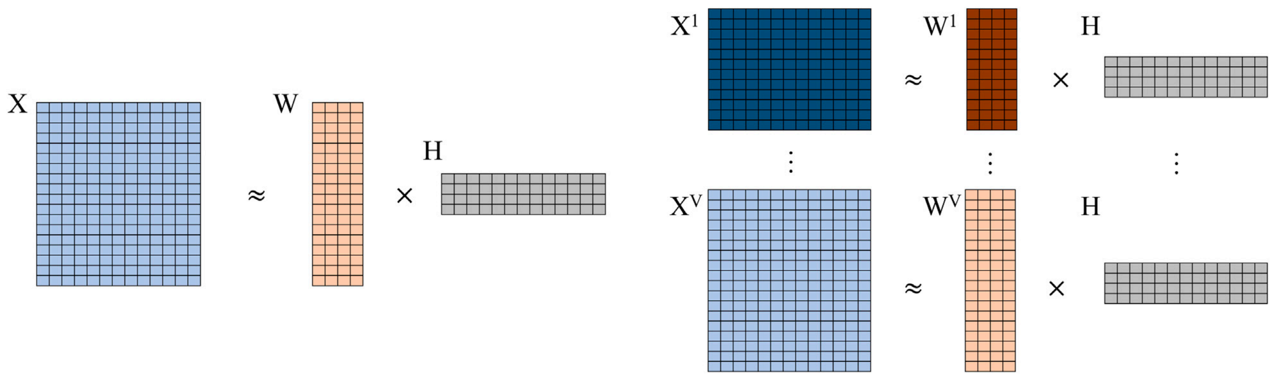


FIGURE 1
Comparison of vanilla NMF (left) to jrSiCKLSNMF without graph or sparsity constraints (right). Note that for jrSiCKLSNMF, **H** is shared among all modalities $v \in 1 \ldots V$.

Team, 2022). We also make extensive use of the RCPP and RCPPARMADILLO packages from Eddelbuettel and François (2011) and Eddelbuettel and Sanderson (2014), respectively. In the next subsection, we introduce and develop our proposed joint NMF model based on the KL divergence with regularization and sparsity constraints.

2.1 Non-negative matrix factorization (NMF)

As detailed above, NMF algorithms approximate an observed, M features by N observations, non-negative data matrix $\mathbf{X}$ as the product of an $M \times D$ non-negative reduced-dimension feature matrix $\mathbf{W}$ and a $D \times N$ non-negative reduced-dimension observation matrix $\mathbf{H}$ such that

$$\mathbf{X} \approx \mathbf{WH}, \tag{1}$$

where $D < \min\{M, N\}$ is the rank of this approximation. Hence, NMF aims to produce a reduced rank approximation of the original non-negative data matrix $\mathbf{X}$. For any $D \times D$ non-negative invertible matrix $\mathbf{Q}$, we have $\mathbf{WQQ}^{-1}\mathbf{H} = \mathbf{WH}$. This implies that $(\mathbf{W}, \mathbf{H})$ and $(\mathbf{WQ}, \mathbf{Q}^{-1}\mathbf{H})$ lead to equivalent approximations. Because of this, $\mathbf{W}$ and $\mathbf{H}$ are not identifiable. The required conditions for identifiability complicate the computational steps, and there has been much work to determine sufficient identifiability criteria (Fu et al., 2018; 2019; Gillis and Rajkó, 2023). However, we can restrict the parameter space by applying different constraints on $\mathbf{W}$ and $\mathbf{H}$. Specifically, we use a graph regularization constraint on $\mathbf{W}$ and propose two possible constraints on $\mathbf{H}$. The first one is a sparsity constraint with a Frobenius norm penalty. The second constraint sets the L2 norm of the rows of $\mathbf{H}$ to 1. These two constraints are compared in simulations. These constraints, along with the non-negative constraints on $\mathbf{W}$ and $\mathbf{H}$, though they do not by any means solve the identifiability issue, can help to mitigate it by reducing the possible solution space for $\mathbf{Q}$ Fu et al. (2019). Additionally, graph regularization constraints on the $\mathbf{W}$ matrix ensure the preservation of geometrical structures within the data. Both the sparsity constraint on $\mathbf{H}$ and the graph regularization constraint on $\mathbf{W}$ enforce sparsity, which is desirable due to sparsity in single-cell omics data (Cai et al., 2008; 2011; Kimura and Yoshida, 2011; Gillis, 2012; Peng et al., 2019; Zhou et al., 2021). Moreover, the unit L2 norm constraint on the rows of $\mathbf{H}$ enables us to avoid tuning the regularization parameter λ_H without sacrificing any accuracy in the clustering results for lower noise levels in our simulation study. The use of the L2 norm constraint also appears, for our real data example, to extract more meaningful factors in the Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018) plots. In order to approximate $\mathbf{X}$ as $\mathbf{WH}$, the most common techniques are to minimize the square of the Frobenius norm of the difference between $\mathbf{X}$ and $\mathbf{WH}$ or to minimize the KL or Itakura-Saito (IS) divergence between the two matrices (Lee and Seung, 1999; Févotte et al., 2009). These methods are all special cases of the β-divergence, with $\beta = 0, 1, 2$ for the Frobenius norm, KL divergence, and IS divergence, respectively Févotte and Idier (2011). For our method, we minimize the loss based on the KL divergence between Poisson($\mathbf{X}$) and Poisson($\mathbf{WH}$) as in Elyanow et al. (2020). Even though $\mathbf{WH}$, the approximation of $\mathbf{X}$ is of the same dimension, the data contained in $\mathbf{WH}$ are of lower resolution compared to the original matrix $\mathbf{X}$. This data compression property of NMF can be helpful for data visualization on top of using the reduced dimensional matrix $\mathbf{H}$ generated by jrSiCKLSNMF algorithm for clustering.

2.2 Motivation for jrSiCKLSNMF

To our best knowledge, jrSiCKLSNMF is the first joint NMF method that simultaneously utilizes the KL divergence across multiple modalities of single-cell count data, a graph regularization constraint for the omics features, and a sparsity constraint for the cells. Many current methods, including Seurat, MOFA+, scAI, and CiteFuse, require using the $\log(x + 1)$ transformation due to the normality assumptions of these models. Similarly, using the Frobenius norm to measure the distance between two count matrices also requires the $\log(x+1)$ transformation. As we mention earlier, transformation of data via the $\log(x+1)$ normalization can introduce bias, especially for UMI data, because it exaggerates the difference between zero and non-zero counts and can thereby negatively impact downstream analyses (Townes et al., 2019). Since we use the Poisson KL divergence, our method does not require data to undergo this $\log(x+1)$ transformation. This method extends the work done in Elyanow et al. (2020); Dai et al. (2020); Liu et al. (2020) to single-cell multimodal omics count data collected from the same set of cells. In Figure 1, we show a comparison of basic (vanilla) NMF and our developed method without sparsity constraints or graph regularization. From this parallel comparison, we can see that the $\mathbf{H}$ matrices are shared among all modalities (v) while the $\mathbf{W}^v$ matrices and median library size normalized count matrices $\mathbf{X}^v$ are different within each modality.

2.3 Loss functions for jrSiCKLSNMF

For our method, we concentrate on two types of loss functions: the first loss function adds a sparsity constraint on $\mathbf{H}$ and the second one sets the square root of the sum of the squared elements of the rows of $\mathbf{H}$ to sum to one. For both constraint methods, we seek to minimize the loss by using multiplicative updates (MU) (Lee and Seung, 1999). Since the constraints on $\mathbf{W}^v$ are the same regardless of the constraints on $\mathbf{H}$, we will describe the graph constraints and their components here. For each $\mathbf{W}^v$, we have a graph Laplacian $\mathbf{L}^v$ that is associated with the feature-feature similarity graph for the raw count data in modality v. Setting $\mathbf{L}^v$ to the $M^v \times M^v$ identity matrix $\mathbf{I}^v$, we have $\mathrm{tr}(\mathbf{W}^v)^T\mathbf{I}^v\mathbf{W}^v = tr((\mathbf{W}^v)^T(\mathbf{W}^v)) = \|\mathbf{W}^v\|_F^2$, which simplifies to a sparsity constraint on the square of the Frobenius norm $\|\cdot\|_F$ of $\mathbf{W}^v$. Penalty parameter $\lambda_{\mathbf{W}^v}$ is a pre-specified constant for the graph regularization parameter on $\mathbf{W}^v$ in each modality. For both loss equations, we use MU. MU is a gradient descent algorithm with an adaptive step size that ensures that all entries of every matrix at each iteration are positive. Eq. 2 defines the KL divergence between the v^{th} median library size normalized omics data matrix $\mathbf{X}^v$ and the matrix product of each reduced dimension omics feature matrix $\mathbf{W}^v$ with shared, reduced dimension cell matrix $\mathbf{H}$, subject to a sparsity constraint on the shared $\mathbf{H}$ and graph regularization on each $\mathbf{W}^v$. For each $\mathbf{X}^v$, x_{ij}^v corresponds to the value in the i^{th} row and j^{th} column.

$$L(\mathbf{X}^v,\mathbf{W}^v,\mathbf{H}) = \min_{\mathbf{W}^v,\mathbf{H}} \sum_{v=1}^{V}\sum_{i=1}^{M^v}\sum_{j=1}^{N} x_{ij}^v \log\left(\frac{x_{ij}^v}{(\mathbf{W}^v\mathbf{H})_{ij}}\right) - x_{ij}^v + (\mathbf{W}^v\mathbf{H})_{ij}$$
$$+ \frac{1}{2}\lambda_{\mathbf{W}^v}\mathrm{tr}\left([\mathbf{W}^v]^T\mathbf{L}^v\mathbf{W}^v\right) + \frac{1}{2}\lambda_{\mathbf{H}}\|\mathbf{H}\|_F^2. \tag{2}$$

Equation 3 is a similar loss functions but instead ensures that the L2 norm $\|\cdot\|_2$ of each column of $\mathbf{H}$ equals 1.

$$L(\mathbf{X}^v,\mathbf{W}^v,\mathbf{H}) = \min_{\mathbf{W}^v,\mathbf{H}} \sum_{v=1}^{V}\sum_{i=1}^{M^v}\sum_{j=1}^{N} x_{ij}^v \log\left(\frac{x_{ij}^v}{(\mathbf{W}^v\mathbf{H})_{ij}}\right) - x_{ij}^v + (\mathbf{W}^v\mathbf{H})_{ij}$$
$$+ \frac{1}{2}\lambda_{\mathbf{W}^v}\mathrm{tr}\left([\mathbf{W}^v]^T\mathbf{L}^v\mathbf{W}^v\right), \text{such that for each column}$$
$$\times\, \mathbf{h} \in \mathbf{H}, \|\mathbf{h}\|_2 = 1. \tag{3}$$

One can also choose to use the Frobenius norm $\|\cdot\|_F$ instead of the KL divergence while dealing with V modalities of continuous data rather than V modalities of count data. We thus outline the objective function with the Frobenius norm and a sparsity constraint on $\mathbf{H}$ in Eq. 4 and the objective function with column L2 norm constraints in Eq. 5:

$$L(\mathbf{X}^v,\mathbf{W}^v,\mathbf{H}) = \min_{\mathbf{W}^v,\mathbf{H}} \sum_{v=1}^{V}\sum_{i=1}^{M^v}\sum_{j=1}^{N} \|\mathbf{X}^v - \mathbf{W}^v\mathbf{H}\|^2$$
$$+ \ \frac{1}{2}\lambda_{\mathbf{W}^v}\mathrm{tr}\left([\mathbf{W}^v]^T\mathbf{L}^v\mathbf{W}^v\right) + \frac{1}{2}\lambda_{\mathbf{H}}\|\mathbf{H}\|_F^2 \tag{4}$$

$$L(\mathbf{X}^v,\mathbf{W}^v,\mathbf{H}) = \min_{\mathbf{W}^v,\mathbf{H}} \sum_{v=1}^{V}\sum_{i=1}^{M^v}\sum_{j=1}^{N} \|\mathbf{X}^v - \mathbf{W}^v\mathbf{H}\|^2 + \frac{1}{2}\lambda_{\mathbf{W}^v}\mathrm{tr}\left([\mathbf{W}^v]^T\mathbf{L}^v\mathbf{W}^v\right),$$
$$\text{such that for each column } \mathbf{h} \in \mathbf{H}, \|\mathbf{h}\|_2 = 1. \tag{5}$$

Equation 4 resembles the joint method SG-jNMF2 outlined in Dai et al. (2020); however, our method places the sparsity constraint on the shared $\mathbf{H}$ matrix and enforces graph regularization on the $\mathbf{W}^v$ parameters in each modality while the method outlined in Dai et al. (2020) places both the graph regularization and the sparsity constraint on either the shared $\mathbf{H}$ when integrating multi-omics data or places both the graph regularization and the sparsity constraint on a shared $\mathbf{W}$ when integrating different datasets with shared features. Although we have not tested using different objective functions in different modalities (i.e., using the KL divergence in one modality and using the Frobenius norm in another), Luo et al. (2019) outline a method called Hybrid NMF (H-NMF), which identifies patient modules via a shared $\mathbf{H}$ but uses the KL divergence in the count genotypic modality and the Frobenius norm in the continuous phenotypic modality.

2.3.1 Gradients of loss function

As the loss functions defined in Eqs 2, 4 do not have closed form minimizers, we apply the gradient descent optimization routine with MU proposed by Lee and Seung (1999). In contrast to traditional gradient descent, here, we compute the updates by using Hadamard (element-wise) products. Specifically, each update is equal to the element-wise product between the current value and a matrix that is the element-wise division of the negative part of the gradient by the positive part of the gradient. It is however important to note that MU updates are derived from the traditional gradient descent step, with a pre-specified rule for the step-size parameter. We compute the gradient of the loss with respect to each $\mathbf{W}^v$ and $\mathbf{H}$ as,

$$\nabla_{\mathbf{W}^v} L(\mathbf{X}^v,\mathbf{W}^v,\mathbf{H}) = (\mathbf{1}_{M\times 1})\left(\mathbf{1}_{1\times N}(\mathbf{H}^v)^T\right) - (\mathbf{X}^v \oslash (\mathbf{W}^v\mathbf{H}))(\mathbf{H}^v)^T$$
$$+ \frac{1}{2}\lambda_{\mathbf{W}^v}\left(\mathbf{L}^v\mathbf{W}^v + (\mathbf{L}^v)^T\mathbf{W}^v\right). \tag{6}$$

In the case of the sparsity constraint on $\mathbf{H}$, we provide the gradient for the loss in Eq. 7a. For the case when we enforce a unit norm constraint on the L2 norms of the rows of $\mathbf{H}$, we also need to modify the gradient as in Eq. 7b. The procedure for calculating the gradient for this constraint is detailed for $\mathbf{W}$ in the single-modality case in Le Roux et al. (2015) and builds on work from Douglas et al. (2000) on gradient descent with unit norm constraints. This modification to the gradient avoids rescaling of $\mathbf{W}^v$ at each iteration to ensure the unit L2 norm constraint holds for the rows of $\mathbf{H}$ and avoids saving a version of $\mathbf{H}$ that has not undergone L2 normalization.

$$\nabla_{\mathbf{H}} L(\mathbf{X}^v, \mathbf{W}^v, \mathbf{H}) = \sum_{v=1}^{V} (\mathbf{W}^v)^T \mathbf{1}_{M^v \times 1} \mathbf{1}_{1 \times N} - (\mathbf{W}^v)^T (\mathbf{X}^v \oslash (\mathbf{W}^v \mathbf{H})) + \lambda_{\mathbf{H}} \mathbf{H}, \tag{7a}$$

$$\begin{aligned}\nabla_{\mathbf{H}} L(\mathbf{X}^v, \mathbf{W}^v, \mathbf{H}) = {} & \sum_{v=1}^{V} (\mathbf{W}^v)^T \mathbf{1}_{M^v \times 1} \mathbf{1}_{1 \times N} - (\mathbf{W}^v)^T (\mathbf{X}^v \oslash (\mathbf{W}^v \mathbf{H})) \\ & - \mathbf{H} \otimes \left(\mathbf{1}_{D \times D}\left((\mathbf{W}^v)^T \mathbf{1}_{M^v \times 1} \mathbf{1}_{1 \times N}\right)\right) \\ & + \mathbf{H} \otimes \left(\mathbf{1}_{D \times D}\left((\mathbf{W}^v)^T (\mathbf{X}^v \oslash (\mathbf{W}^v \mathbf{H}))\right)\right).\end{aligned} \tag{7b}$$

We use these gradients to obtain the MU rules for each $\mathbf{W}^v$ and for $\mathbf{H}$.

2.4 Computation

Fitting NMF models to omics data entails many challenges, including appropriate data pre-processing, normalization, and algorithmic initialization of NMF. For clarity, we explain these steps in Sections 2.4.1–2.4.4 before providing an overview of the algorithm in Section 2.4.5.

2.4.1 Quality control and normalization

Before applying the algorithm, we must perform quality control (QC) and normalization. These are vital steps for downstream analyses (Ellis et al., 2021). For QC, it is appropriate to perform standard QC, including filtering out low-quality cells, such as those with a high percentage of mitochondrial genes, low gene expression, or very high gene expression in the scRNA-seq modality. For both of the datasets we used in our analysis, this QC step was already performed. Since we develop this method primarily for multimodal single-cell data, from now on, we refer to "observations" as "cells" and "features" as "omics features." In the case of scRNA-seq data, the entries of the data before median library size normalization would be the UMI counts; and for scATAC-seq data, these are the counts of accessible peaks/bins. To generate the median library size normalized matrix $\mathbf{X}^v$ for each modality v, we first divide the counts in each cell by the sum of counts within that cell (i.e., the library size) and then multiply all entries by the median library size (Zheng et al., 2017; Elyanow et al., 2020). This does not violate count assumptions for the Poisson distribution. We use the KL divergence to measure the discrepancy between the distributions Poisson($\mathbf{X}^v$) and Poisson($\mathbf{W}^v\mathbf{H}$).

2.4.2 Construction of the $\mathbf{L}^v$ matrices

The $\mathbf{L}^v$ matrix is the $M^v \times M^v$ graph Laplacian matrix of $\mathbf{G}^v$. $\mathbf{G}^v$ is an $M^v \times M^v$ interaction network graph within the v^{th} omics modality. We construct $\mathbf{L}^v$ from the raw data rather than from the median library size normalized data. To construct the graph Laplacian matrix $\mathbf{L}^v$, one first needs to define $\mathbf{A}^v$, the adjacency matrix of $\mathbf{G}^v$, and $\mathbf{D}^v$, the diagonal matrix of vertex degrees of $\mathbf{G}^v$. The graph Laplacian matrix is defined as $\mathbf{L}^v = \mathbf{D}^v - \mathbf{A}^v$ (Merris, 1994). Optimally, to construct $\mathbf{G}^v$, one would use data from a different single-cell experiment on the same tissue or from a bulk experiment on the same tissue to avoid overfitting. $\mathbf{G}^v$ is not restricted to a specific kind of graph; this method can accommodate the use of any graph that accurately captures the similarity between features (Cai et al., 2008; 2011). The use of co-expression networks from bulk tissue studies is also permissible (Elyanow et al., 2020). In our analyses, we use k-nearest neighbor (KNN) graphs as implemented in the SCRAN package (Lun et al., 2016) to generate the graph $\mathbf{G}^v$ for each modality. We also tested using shared nearest neighbor (SNN) graphs; however, regularization using KNN outperformed these SNN graphs. Because we are calculating the feature-feature similarities and $M^v \gg N$ for all modalities v, distances calculated in Euclidean space for the KNN graph are meaningful. In the case when $N > M^v$, we would need a different approach for constructing graphs. Since we perform this graph construction on feature-feature networks, we will, without loss of generality, refer to each point within the constructed graph as a feature.

2.4.3 Determination of D and initialization of the $\mathbf{W}^v$ matrices and the $\mathbf{H}$ matrix

An important aspect of using any NMF-based method to analyze data matrix $\mathbf{X}^v$ is the determination of the number of latent factors D and the initialization of matrices $\mathbf{W}^v$ and $\mathbf{H}$. Since our method of identifying an appropriate number of latent factors requires initializing and updating $\mathbf{W}^v$ and $\mathbf{H}$, we will discuss their initialization first. Random initialization is a common way to initialize $\mathbf{W}$ and $\mathbf{H}$ (Lee and Seung, 1999; Cai et al., 2008; Elyanow et al., 2020; Liu et al., 2020), but many other methods of initialization have been developed over the years. In particular, initialization based on singular-value decomposition (SVD) has become increasingly popular (Boutsidis and Gallopoulos, 2008; Qiao, 2015; Esposito, 2021) as a way of initializing non-negative matrix factorization problems. To initialize $\mathbf{W}^v$ for each modality, we first perform Non-negative Double Singular Value Decomposition (NNDSVD), a method developed by Boutsidis and Gallopoulos (2008) for NMF initialization, on each $\mathbf{X}^v$ and use the $\mathbf{W}^v$ matrices from each output. To initialize $\mathbf{H}$, we concatenate all $\mathbf{X}^v$ together to generate $\mathbf{X}^{all}$, perform NNDSVD on this concatenated matrix, and then use the $\mathbf{H}$ matrix from the NNDSVD output. While NNDSVD encourages a sparse initialization, because we use MU which cannot escape from 0 values, we use a dense initialization where we insert the average value instead of 0. Additionally, since NNDSVD is a non-negative version of singular value decomposition, the sum of each eigenvector decreases for each component as the number of factors increases. This is not necessarily the case for NMF. We therefore initialize $\mathbf{W}^v$ such that each column sums to the mean column sum. We perform this same operation on the rows of $\mathbf{H}$. We also tested using random initialization, which, due to ease of implementation, is a common method of initialization. It did not perform as accurately as NNDSVD and, on simulated data with no added noise, an individual regularization graph, and a sparsity constraint on $\mathbf{H}$, had an adjusted Rand index (ARI) (Hubert and Arabie, 1985) of 0.886, which was about 10% lower than the 0.988 achieved using NNDSVD. We provide side-by-side boxplots of these results on simulated data in Supplementary Figure S1.

It can be difficult to identify an appropriate D for unsupervised data problems like clustering. In our workflow, we provide a method of visual selection. We initialize the $\mathbf{W}^v$ and $\mathbf{H}$ matrices for a user-specified range of number of factors (default is 2–20) under either

NNDSVD or random initialization (we strongly recommend NNDSVD). We then run the algorithm for a specified number of iterations (100 for sparsity constraint and 1 for the L2 Norm constraint) and then plot the resulting loss function. We recommend selecting the number of latent factors that corresponds to the elbow of the plot. We provide an example of this on real data in Section 3 in Figure 6. The computational time increases with increasing D; for an example of this, see Supplementary Figure S2.

2.4.4 Selection of λ values

Selection of the λ values is a time-intensive step. As the number of modalities increases, the selection step becomes even more time demanding. We thus run extensive simulations for scRNA-seq and scATAC-seq data and, using these simulations, identify some default choices for these parameters. Based on our experiments we find that $\lambda_{W^{RNA}} = 10$, $\lambda_{W^{ATAC}} = 50$, and $\lambda_H = 500$ work well for the sparsity constraint model and that $\lambda_{W^{RNA}} = 3$, $\lambda_{W^{ATAC}} = 15$ work well for the L2 norm constraint on the rows of $\mathbf{H}$. To illustrate this, we provide a plot of the ARI values for 512 combinations of $\lambda_{W^{RNA}}, \lambda_{W^{ATAC}}$, and λ_H in Supplementary Figure S3 for a fixed $D = 10$ for the no-added-noise simulated data.

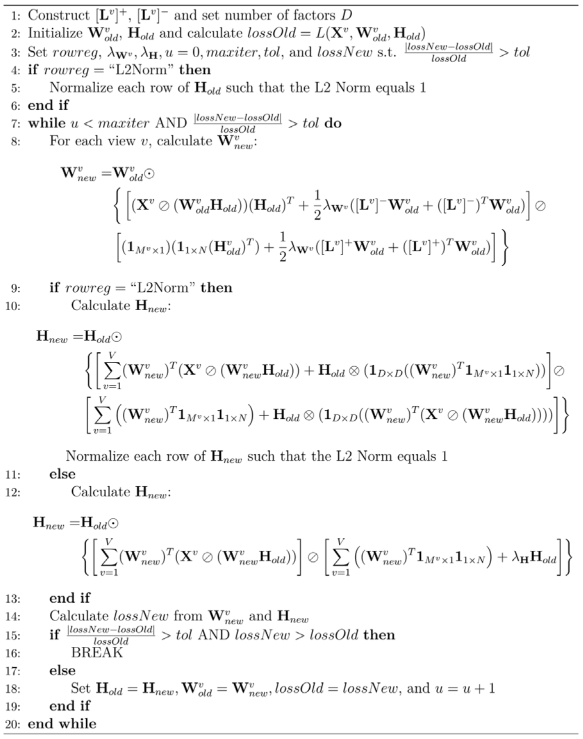
We recommend $\lambda_{W^{RNA}} = 10$, $\lambda_{W^{ATAC}} = 50$, and $\lambda_H = 500$ as the default for the sparsity constraint model and $\lambda_{W^{RNA}} = 3$, $\lambda_{W^{ATAC}} = 15$ for the model with the L2 norm constraint on the rows of $\mathbf{H}$ as the default choices for all of our simulations and our real data application. The value of 10 for the RNA modality agrees with previous literature for KL-based NMF (KL-NMF) algorithms on scRNA-seq data (Elyanow et al., 2020). Finally, the computational time does not seem highly dependent on these values, but we do see faster computational times for $\lambda_{W^{RNA}} = \lambda_{W^{ATAC}} = 1000$. We plot these in Supplementary Figure S4. However, $\lambda_{W^{RNA}} = \lambda_{W^{ATAC}} = 1000$ are not considered due to their poor performance.

2.4.5 Overview of algorithm

The pseudocode in Figure 2 summarizes all the steps for the jrSiCKLSNMF algorithm. First, we must construct the graph-Laplacian matrices from feature-feature similarity graphs and select a number of factors D that we wish to use to construct the $\mathbf{W}^v$ matrices and the $\mathbf{H}$ matrix. Note that D must be the same across all modalities. Next, we set the $\lambda_{\mathbf{W}^v}$ values, the $\lambda_{\mathbf{H}}$ value, the update tolerance, and the new loss. The λ values are tuning parameters. For our simulations, we set the maximum number of iterations to 10,000 and the tolerance to 10^{-6} for both the sparsity and the L2 norm constraint. Then, using MU we iteratively update $\mathbf{W}^v$ and $\mathbf{H}$ until convergence (i.e., the percentage difference of the update is less than the tolerance) or we reach a maximum number of iterations. In line 8 of Figure 2 we show the multiplicative updates for $\mathbf{W}^v_{u+1}$, the $(u+1)^{\text{th}}$ updates of the $\mathbf{W}^v$ matrices in sequence, using the corresponding feature matrices $\mathbf{W}^v_u$ and cell matrix $\mathbf{H}_u$. Similarly, in line 10 of Figure 2, we show the calculations:

$$\mathbf{W}^v_{u+1} = \mathbf{W}^v_u \odot \left\{ \left[\left(\mathbf{X}^v \oslash \left(\mathbf{W}^v_u \mathbf{H}_u\right)\right)\left(\mathbf{H}_u\right)^T + \frac{1}{2}\lambda_{\mathbf{W}^v}\left([\mathbf{L}^v]^- \mathbf{W}^v_u + \left([\mathbf{L}^v]^-\right)^T \mathbf{W}^v_u \right) \right] \oslash \left[(\mathbf{1}_{M\times 1})\left(\mathbf{1}_{1\times N}\left(\mathbf{H}_u\right)^T\right) + \frac{1}{2}\lambda_{\mathbf{W}^v}\left([\mathbf{L}^v]^+ \mathbf{W}^v_u + \left([\mathbf{L}^v]^+\right)^T \mathbf{W}^v_u \right) \right] \right\}. \tag{8}$$

here, $[\mathbf{L}^v]^-$ and $[\mathbf{L}^v]^+$ indicate the absolute values of the negative and the positive parts of the $\mathbf{L}^v$ in each modality, respectively, the $\odot$ symbol indicates the Hadamard product, and the $\oslash$ symbol indicates Hadamard division. After updating all $\mathbf{W}^v$ matrices, we proceed to updating $\mathbf{H}_{u+1}$ from the new $\mathbf{W}^v_{u+1}$ matrices and the old $\mathbf{H}_u$ matrix via Eq. 9a for the sparsity constraint on $\mathbf{H}$ and Eq. 9b for the L2 norm constraint.



```
1: Construct $[\mathbf{L}^v]^+$, $[\mathbf{L}^v]^-$ and set number of factors $D$
2: Initialize $\mathbf{W}^v_{old}$, $\mathbf{H}_{old}$ and calculate $lossOld = L(\mathbf{X}^v, \mathbf{W}^v_{old}, \mathbf{H}_{old})$
3: Set $rowreg$, $\lambda_{\mathbf{W}^v}, \lambda_{\mathbf{H}}, u = 0, maxiter, tol$, and $lossNew$ s.t. $\frac{|lossNew - lossOld|}{lossOld} > tol$
4: if $rowreg$ = "L2Norm" then
5:     Normalize each row of $\mathbf{H}_{old}$ such that the L2 Norm equals 1
6: end if
7: while $u < maxiter$ AND $\frac{|lossNew - lossOld|}{lossOld} > tol$ do
8:     For each view $v$, calculate $\mathbf{W}^v_{new}$:

       $\mathbf{W}^v_{new} = \mathbf{W}^v_{old} \odot \left\{ \left[ (\mathbf{X}^v \oslash (\mathbf{W}^v_{old}\mathbf{H}_{old}))(\mathbf{H}_{old})^T + \frac{1}{2}\lambda_{\mathbf{W}^v}([\mathbf{L}^v]^- \mathbf{W}^v_{old} + ([\mathbf{L}^v]^-)^T \mathbf{W}^v_{old}) \right] \oslash \left[ (\mathbf{1}_{M^v\times 1})(\mathbf{1}_{1\times N}(\mathbf{H}^v_{old})^T) + \frac{1}{2}\lambda_{\mathbf{W}^v}([\mathbf{L}^v]^+ \mathbf{W}^v_{old} + ([\mathbf{L}^v]^+)^T \mathbf{W}^v_{old}) \right] \right\}$

9:     if $rowreg$ = "L2Norm" then
10:        Calculate $\mathbf{H}_{new}$:

           $\mathbf{H}_{new} = \mathbf{H}_{old} \odot \left\{ \left[ \sum_{v=1}^{V} (\mathbf{W}^v_{new})^T (\mathbf{X}^v \oslash (\mathbf{W}^v_{new}\mathbf{H}_{old})) + \mathbf{H}_{old} \otimes (\mathbf{1}_{D\times D}((\mathbf{W}^v_{new})^T \mathbf{1}_{M^v\times 1}\mathbf{1}_{1\times N})) \right] \oslash \left[ \sum_{v=1}^{V} \left((\mathbf{W}^v_{new})^T \mathbf{1}_{M^v\times 1}\mathbf{1}_{1\times N}\right) + \mathbf{H}_{old} \otimes (\mathbf{1}_{D\times D}((\mathbf{W}^v_{new})^T (\mathbf{X}^v \oslash (\mathbf{W}^v_{new}\mathbf{H}_{old})))) \right] \right\}$

           Normalize each row of $\mathbf{H}_{new}$ such that the L2 Norm equals 1
11:    else
12:        Calculate $\mathbf{H}_{new}$:

           $\mathbf{H}_{new} = \mathbf{H}_{old} \odot \left\{ \left[ \sum_{v=1}^{V} (\mathbf{W}^v_{new})^T (\mathbf{X}^v \oslash (\mathbf{W}^v_{new}\mathbf{H}_{old})) \right] \oslash \left[ \sum_{v=1}^{V} \left((\mathbf{W}^v_{new})^T \mathbf{1}_{M^v\times 1}\mathbf{1}_{1\times N}\right) + \lambda_{\mathbf{H}}\mathbf{H}_{old} \right] \right\}$

13:    end if
14:    Calculate $lossNew$ from $\mathbf{W}^v_{new}$ and $\mathbf{H}_{new}$
15:    if $\frac{|lossNew - lossOld|}{lossOld} > tol$ AND $lossNew > lossOld$ then
16:        BREAK
17:    else
18:        Set $\mathbf{H}_{old} = \mathbf{H}_{new}$, $\mathbf{W}^v_{old} = \mathbf{W}^v_{new}$, $lossOld = lossNew$, and $u = u + 1$
19:    end if
20: end while
```

FIGURE 2
Pseudocode for jrSiCKLSNMF. Note that the sparsity parameter is not included in the row regularization. While it is possible to use both the $\lambda_{\mathbf{H}}$ sparsity parameter and the unit L2 norm constraint on the rows of $\mathbf{H}$, it is not necessary. Since we did not use both constraints simultaneously, to save space, we are excluding the $\lambda_{\mathbf{H}}$ term when *rowreg* = "L2Norm" from the calculation of $\mathbf{H}_{u+1}$.

$$\mathbf{H}_{u+1} = \mathbf{H}_u \odot \left\{ \left[\sum_{v=1}^{V} \left(\mathbf{W}^v_{u+1}\right)^T \left(\mathbf{X}^v \oslash \left(\mathbf{W}^v_{u+1}\mathbf{H}_u\right)\right) \right] \oslash \left[\sum_{v=1}^{V} \left(\left(\mathbf{W}^v_{u+1}\right)^T \mathbf{1}_{M\times 1}\left(\mathbf{1}_{1\times N}\right) \right) + \lambda_{\mathbf{H}}\mathbf{H}_u \right] \right\}, \tag{9a}$$

$$\mathbf{H}_{u+1} = \mathbf{H}_u \odot \left\{ \left[\sum_{v=1}^{V} \left(\mathbf{W}^v_{u+1}\right)^T \left(\mathbf{X}^v \oslash \left(\mathbf{W}^v_{u+1}\mathbf{H}_u\right)\right) + \mathbf{H}_u \otimes \left(\mathbf{1}_{D\times D}\left(\left(\mathbf{W}^v_{u+1}\right)^T \mathbf{1}_{M^v\times 1}\mathbf{1}_{1\times N} \right)\right) \right] \oslash \left[\sum_{v=1}^{V} \left(\left(\mathbf{W}^v_{u+1}\right)^T \mathbf{1}_{M^v\times 1}\mathbf{1}_{1\times N} \right) + \mathbf{H}_u \otimes \left(\mathbf{1}_{D\times D}\left(\left(\mathbf{W}^v_{u+1}\right)^T \left(\mathbf{X}^v \oslash \left(\mathbf{W}^v_{u+1}\mathbf{H}_u\right)\right) \right)\right) \right] \right\}. \tag{9b}$$

This process of iterative updates continues until the algorithm converges.

2.5 Clustering

In our *post hoc* analysis of the simulated data, we perform k-means clustering on the estimated $\mathbf{H}$ matrix. For fair comparison, we set the number of clusters to be equal to the true value. For Seurat, which uses a resolution parameter rather than the number of clusters, we experiment with a subset of the data to determine a suitable resolution parameter that ensures that the number of clusters is close to 3. One may use any clustering method to perform clustering on the consensus matrix **H**, including using the **H** matrix itself as a clustering algorithm. We use k-means because we can set the number of clusters to the true number of clusters easily, and it has good clustering performance. To aid in the determination of the number of clusters on real datasets, we provide wrapper functions for the R packages NBCLUST (Charrad et al., 2014) and CLVALID (Brock et al., 2008). These packages generate validation metrics and plots to help in determining the ideal number of clusters for k-means and other clustering methods.

3 Results

To compare the performance of our algorithm against other methods, we perform a simulation study. Since our algorithm is for use with exploratory data analyses and clustering, it is somewhat difficult to evaluate its performance on a real dataset where the true clusters are unknown. We use GSE130399 (Zhu et al., 2019), which is labeled, to generate parameters from which to simulate datasets, and GSE126074 (Chen et al., 2019), which has an annotation but is not labeled, to assess the performance of our algorithm on a real data example. To perform our simulation study, we use two different R packages: SPARSIM (Baruzzo et al., 2020) for scRNA-seq data simulation and SIMATAC (Navidi et al., 2021), for scATAC-seq data simulation. We generate all plots using the R package ggplot2 version 3.4.2 (Wickham, 2016).

3.1 Evaluation metrics for clustering

To determine the accuracy of clusters and compare these clusters to other methods, we use the ARI as implemented in the R package ARICODE. We use this to evaluate how the clusters identified by each method compare to the ground truth in the simulated data and to the annotations for the real data. The ARI uses the hypergeometric distribution to correct for clusters that are correct due to random chance. We also explored comparison of the adjusted mutual information (AMI) (Xuan Vinh et al., 2009), and the results were similar.

3.2 Simulation study

For the simulation study, we simulate four sets of 100 independent dual-assay scRNA-seq/scATAC-seq datasets, each with increasing amounts of added noise starting from 0. Each dataset consists of 100 cells each of 3 different cell types for a total of 300 cells. There are approximately 900 genes in the scRNA-seq modality and approximately 5,800 bins in the scATAC-seq modality for the simulated cells. These vary marginally among simulations. In the next section, we discuss the reasoning behind this number of genes and bins. We use this labeled simulated data to determine λ values as well by examining different combinations of λ values and their corresponding ARIs. We then choose the values that correspond to the highest average ARI.

3.2.1 Data simulation scheme

To simulate the data jointly, we use the R packages SPARSIM (Baruzzo et al., 2020) to simulate scRNA-seq expresion and SIMATAC (Navidi et al., 2021) to simulate scATAC-seq expression. We estimate simulation parameters from GSE130399, a real Paired-seq (Zhu et al., 2019) cell-line dataset. SPARSIM estimates parameters from real data and then uses a Gamma-Multivariate hypergeometric mixture model to simulate scRNA-seq count data. SIMATAC also estimates parameters from real data but uses a Bernoulli-Poisson hurdle model to generate data. To prepare the data for parameter estimation, we perform aggressive quality control using the R packages SEURAT (Satija et al., 2015) and SIGNAC (Stuart et al., 2021) for the scRNA-seq modality and scATAC-seq modality, respectively. First, we exclude cells which have fewer than 400 and greater than 2000 RNA counts and cells that have fewer than 300 or greater than 4000 ATAC bins. In the RNA modality, we exclude genes with fewer than 10 counts per cell and in the ATAC modality, we exclude bins with fewer than 20 counts per cell as in Zhu et al. (2019).

We then select the 1,000 most highly variable genes in the RNA-seq modality and the features that are common among 95% of the cells in the ATAC-seq modality. After performing this quality control and feature selection, we are left with 382 HEK293 cells, 366 HepG2 cells, and 1,003 mix cells from which to sample. To generate each of the 100 datasets, we randomly select 100 HepG2 cells, 100 HEK293 cells, and 100 mix cells without replacement. The mix cells are a mixture of the HepG2 and HEK293T cells; however, for the purpose of generating data for this simulation, we treat them as a third cell type. From this subset, we then use SPARSIM to estimate simulation parameters and finally generate cells for the RNA modality and use SIMATAC to estimate simulation parameters and generate simulated cells for the ATAC modality. To avoid confusion between modalities, instead of using M^v, we use M^{RNA} to denote the number of features in the scRNA-seq modality and M^{ATAC} to denote the number of features in the scATAC-seq modality. As mentioned earlier, we generate four sets of datasets; one with no noise and three with increasing amounts of noise. SPARSIM and SIMATAC simulate added noise differently; SPARSIM uses an estimated variability parameter and SIMATAC adds noise from a Gaussian distribution to the final dataset. Therefore, in our simulation study, we follow the respective protocols for adding noise to each modality. For the lowest added noise datasets, for each simulated dataset, we generate noise from a uniform distribution ($\mathcal{U}(1, 1.25)$) and multiply this noise by the corresponding variability parameter for each RNA feature. In the ATAC modality, we simulate the data in SIMATAC and then, following the protocol for generating noise, add Gaussian noise generated from normal distribution ($\mathcal{N}(-0.25, 0.25)$) for each entry in the $\mathbf{X}^{ATAC}$ matrix. We repeat this noise generation process twice more, using distributions $[\mathcal{U}(1, 1.5), \mathcal{N}(-0.5, 0.5)]$ and $[\mathcal{U}(1, 2), \mathcal{N}(-1, 1)]$.

3.3 Current single-cell multimodal omics methods

Since this is a relatively new technology, we compare our method to five other methods of integrating single-cell

TABLE 1 Methods for comparison to jrSiCKLSNMF with a brief description *Seurat has also been successfully used on dual assay scRNA-seq/scATAC-seq but was developed for CITE-seq data.

Method	Data designed for	Type of model
BREM-SC	CITE-seq	Bayesian random effects mixture
CiteFuse	CITE-seq	Similarity Network Fusion
MOFA+	Any two -omics datasets	Factor Analysis
scAI	scRNA-seq and ATAC-seq	Non-negative Matrix Factorization
Seurat v. 4	CITE-seq*	Weighted Nearest Neighbor

multimodal omics data. These methods are not necessarily designed for use with dual scRNA-seq and scATAC-seq data. These methods are BREM-SC (Wang et al., 2020), Seurat v. 4.0, MOFA+, scAI, and CiteFuse. We briefly describe them in Table 1 and describe them in more detail in sub-subSections 3.3.1–3.3.4. This is not an exhaustive list of methods, and all of these methods are implemented in R. There are other methods that are implemented in Python (Van Rossum and Drake, 2009) that we do not discuss here. Each of these methods can work with, at a minimum, two modalities of simultaneous measurements of omics data on the same set of single cells. Some, like MOFA+, can work with more than two modalities. While the focus of our comparisons is on these 5, there are other methods of integrating data across omics profiles.

3.3.1 BREM-SC

The Bayesian random effects mixture model for single-cell multi-omics data (BREM-SC) model is intended for use on data collected from cellular indexing of transcriptomes and epitopes by sequencing (CITE-seq). These are joint RNA and Antibody-Derived Tags (ADT) single-cell data. ADT data are much lower dimension than scRNA-seq since it only works with a few proteins per cell; in Stoeckius et al. (2017), which introduces CITE-seq, the number of features in the ADT modality is 13 (Stoeckius et al., 2017). BREM-SC uses a Bayesian Dirichlet-multinomial model with cell-specific random effects shared between the two modalities to perform cell clustering. In Eq. 10, we provide the complete log likelihood for BREM-SC:

$$\begin{aligned}\log P\boldsymbol{\alpha}^{RNA}, \boldsymbol{\alpha}^{ADT}, Z, b_j \mid \mathbf{X}^{RNA}, \mathbf{X}^{ADT}\big) &\propto \sum_{j=1}^{C}\sum_{k=1}^{K} I\left(z_j = k\right)\\ &\times \log\left\{\left(\prod_{i=1}^{G}\frac{\Gamma\left(x_{ij}^{RNA} + \alpha_{i(k)}^{RNA}b_j\right)}{\Gamma\left(\alpha_{i(k)}^{RNA}b_j\right)}\right)\frac{\Gamma\left(\left|\alpha_{(k)}^{RNA}b_j\right|\right)}{\Gamma\left(T_j^{RNA} + \left|\alpha_{(k)}^{RNA}b_j\right|\right)}\right.\\ &\left.\times\left(\prod_{d=1}^{D}\frac{\Gamma\left(x_{dj}^{ADT} + \alpha_{d(k)}^{ADT}b_j\right)}{\Gamma\left(\alpha_{d(k)}^{ADT}b_j\right)}\right)\times\frac{\Gamma\left(\left|\alpha_{(k)}^{ADT}b_j\right|\right)}{\Gamma\left(T_j^{ADT} + \left|\alpha_{(k)}^{ADT}b_j\right|\right)}\right\}\\ &+\sum_{j=1}^{C}\left(-\log b_j - \frac{\left(\log b_j\right)^2}{2\sigma_b^2}\right)\\ &+\sum_{j=1}^{C}\left(-\frac{1}{2}\log\sigma_b^2\right) \qquad (10)\end{aligned}$$

$\boldsymbol{\alpha}^{RNA}$, a G gene by K cluster matrix and $\boldsymbol{\alpha}^{ADT}$, a D protein marker by K matrix, contain the cell cluster-specific Dirichlet parameters for the RNA and ADT modalities, respectively. $\alpha_{i(k)}^{RNA}$ is the value for gene i in cluster k of $\boldsymbol{\alpha}^{RNA}$, and $\alpha_{d(k)}^{ADT}$ is the value for protein marker d in cluster k of $\boldsymbol{\alpha}^{ADT}$. $\alpha_{(k)}^{RNA}$ and $\alpha_{(k)}^{ADT}$ are the vectors of Dirichlet priors for the k^{th} cell cluster in the RNA and ADT modalities, respectively. If cell j belongs to the k^{th} cell type, its gene expression profile $\mathbf{p}_j^{RNA}$ follows cell-type-specific prior distribution $\mathrm{Dir}(\alpha_{(k)}^{RNA})$, and its marker expression profile $\mathbf{p}_j^{ADT}$ follows $\mathrm{Dir}(\alpha_{(k)}^{ADT})$. Z is a latent variable vector comprised of elements z_j that represent the cell type label $k \in (1, \ldots K)$ for each cell $j \in (1, \ldots, C)$. Here, C is the total number of cells, and K is the total number of cell labels. b_j is the random effect for the j^{th} cell and follows distribution $\mathrm{LogNormal}(0, \sigma_b^2)$, where σ_b^2 indicates the among-cell variability. $\mathbf{X}^{RNA}$ and $\mathbf{X}^{ADT}$ are the G gene by C RNA data matrix and D protein marker by C ADT data matrix, respectively. $I(\cdot)$ is the indicator function and $\Gamma(\cdot)$ is the gamma function. x_{ij}^{RNA} is the entry in the i^{th} row and j^{th} column of $\mathbf{X}^{RNA}$ while x_{dj}^{ADT} is the entry in the d^{th} row and j^{th} column of $\mathbf{X}^{ADT}$. Finally, T_j^{RNA} and T_j^{ADT} are the total UMI counts and the total ADT counts, respectively for the j^{th} cell. BREM-SC uses a Gibbs sampler to update cluster assignment z_j and uses a random walk Metropolis within Gibbs sampler to iteratively update $\alpha_{(k)}^{RNA}$, $\alpha_{(k)}^{ADT}$, and b_j.

3.3.2 CiteFuse

CiteFuse, like BREM-SC, is also intended for CITE-Seq (dual assay scRNA-seq and single-cell ADT) data. It uses similarity network fusion to integrate the two modalities. First, CiteFuse performs a centered log-ratio transformation to normalize the ADT counts. Next, it calculates cell-to-cell similarity by using a similarity metric called *perb* from R package PROPR (Quinn et al., 2017). For the RNA expression, it uses Pearson's correlation on highly variable genes identified by SCRAN. It then scales the matrices using an exponential similarity kernel and fuses them via a similarity network fusion algorithm (Wang et al., 2014). To compare to our method, we use *perb* for the scRNA-seq data and Pearson's correlation for scATAC-seq because, as scRNA-seq data are sparser and noisier than ADT data, so too are scATAC-seq data sparser and noisier than scRNA-seq data.

3.3.3 MOFA+

Multi-omics Factor Analysis v2 (MOFA+) captures global sources of variability in multi-omics data in a small number of latent factors via a Bayesian matrix factorization framework. MOFA+ can be used on single-cell data, grouped data, and is available for more than two modalities. Eq. 11 gives the underlying equation for the matrix factorization model:

$$\mathbf{Y}_{gm} = \mathbf{Z}_g\mathbf{W}_m^T + \boldsymbol{\epsilon}_{gm}. \qquad (11)$$

Here, $\mathbf{Y}_{gm}$ is a matrix of observations of the m^{th} modality and g^{th} group. For single-cell data, group indicates the source of the tissue. $\mathbf{W}_m$ is a weight matrix for the m^{th} modality, $\mathbf{Z}_g$ is the factor matrix for the g^{th} group and $\boldsymbol{\epsilon}_{gm}$ represents the residual for the m^{th} modality and the g^{th} group. Each $\mathbf{Z}_g$ is of dimension $N_g \times K$, where N_g is the number of observations per group and K is the number of latent factors. $\mathbf{W}_m^T$ has dimension $D_m \times K$, where D_m is the number of features in the M^{th} modality. It also uses regularization for both the factors and weights in the form of an Automatic Relevance Determination (ARD) prior to model activity of factors across modalities or sample groups and a spike-and-slab prior to encourage sparsity.

3.3.4 scAI

scAI, like our method, is based on NMF. Eq. 12 is the loss function for scAI where M^1 genes by N cells matrix $\mathbf{X}_1$ and M^2 loci by N cells matrix $\mathbf{X}_2$ correspond to RNA and ATAC modalities, respectively. $\mathbf{W}_1$ is an M^1 by D factors gene loading matrix, $\mathbf{W}_2$ is an M^2 by D loci loading matrix, $\mathbf{H}$ is the $D \times N$ cell loading matrix where $\mathbf{H}_{.j}$ is the j^{th} column of $\mathbf{H}$, the $\mathbf{Z}$ matrix is a cell-cell similarity matrix, $\circ$ represents dot multiplication, $\mathbf{R}$ is a binary matrix generated by a binomial distribution with probability s, and α, λ, and γ are regularization parameters. Like our method, it shares the $\mathbf{H}$ matrix but, unlike our method, it binarizes the ATAC-seq modality of the data.

$$\min_{\mathbf{W}_1,\mathbf{W}_2,\mathbf{H},\mathbf{Z}\geq 0} \alpha\|\mathbf{X}_1 - \mathbf{W}_1\mathbf{H}\|_F^2 + \|\mathbf{X}_2(\mathbf{Z}\circ\mathbf{R}) - \mathbf{W}_2\mathbf{H}\|_F^2 + \lambda\left\|\mathbf{Z} - \mathbf{H}^T\mathbf{H}\right\|_F^2 + \gamma\sum_j \left\|\mathbf{H}_{.j}\right\|_1^2 \tag{12}$$

Interestingly, even though this algorithm is fairly similar to ours, their implementation performs poorly in our comparative study on simulated data. This may illustrate the importance of the graph regularization constraints.

3.3.5 Seurat

Seurat v.4 uses weighted nearest neighbor (WNN) to integrate bimodal single-cell data. Like BREM-SC and CiteFuse, it was developed for CITE-seq data; however it has also been used for scATAC-seq and scRNA-seq dual assay data. After quality control, normalization, and dimension reduction on each modality, Seurat constructs independent KNN graphs for both modalities. Next, it performs within and across-modality prediction and cell-specific modality weights:

$$\theta_{\text{weighted}}(i,j) = w_{\text{RNA}}(i)\theta_{\text{RNA}}\left(r_i, r_j\right) + w_{\text{protein}}(i)\theta_{\text{protein}}\left(p_i, p_j\right). \tag{13}$$

$\theta_{\text{weighted}}(i,j)$ is the weighted similarity between cells i and j, $w_{\text{rna}}(i)$ is the cell-specific RNA weight, $\theta_{\text{RNA}}(r_i, r_j)$ is the affinity between the RNA profiles of cells i and j, $w_{\text{protein}}(i)$ is the cell-specific ADT weight, $\theta_{\text{protein}}(r_i, r_j)$ is the affinity between the ADT profiles of cells i and j. Then, a final KNN graph is constructed using $\theta_{\text{weighted}}(i,j)$ as the similarity metric. To identify clusters, Seurat uses community detection algorithms on this graph.

3.4 Comparison to other methods on simulated data

We compare our method to the five methods discussed in the previous section. The numerical comparisons are illustrated in Table 2, with the best performing value in bold. For every level of noise, a version of jrSiCKLSNMF performed best in terms of ARI. We include four variants of jrSiCKLSNMF in our comparison: jrSiCKLSNMF-B:L2, jrSiCKLSNMF-B:SH, jrSiCKLSNMF-I:L2, and jrSiCKLSNMF-I:SH. The first variant, jrSiCKLSNMF-B:L2, is jrSiCKLSNMF with graph regularization term $\mathbf{L}^v$ constructed from a feature-feature KNN graph built from simulated bulk data (i.e., $\mathbf{L}^v$ is the same for all 400 datasets). jrSiCKLSNMF-B:L2 also has a unit L2 norm constraint on the rows of $\mathbf{H}$. For the second variant jrSiCKLSNMF-B:SH, the $\mathbf{L}^v$ used is the same as the one used in jrSiCKLSNMF-B:L2, but there is a sparsity constraint on $\mathbf{H}$. For the third variant jrSiCKLSNMF-I:L2, $\mathbf{L}^v$ is different for each of the 400 datasets and is constructed individually from each dataset's feature-feature KNN graph. It also, like jrSiCKLSNMF-B:L2, has a unit L2 norm constraint on the rows of $\mathbf{H}$. The final variation jrSiCKLSNMF-I:SH has individual $\mathbf{L}^v$ matrices for each dataset as in jrSiCKLSNMF-I:L2 and has a sparsity constraint on $\mathbf{H}$ as in jrSiCKLSNMF-B:SH. Except for MOFA+, which was run on a 16.0 RAM local machine due to difficulty with setting up `Python` modules from RETICULATE (Ushey et al., 2023) on the cluster, we ran all analyses on the HiPerGator 3.0 high performance cluster. As such, MOFA + may have slightly faster mean running times listed here than it would if it were run on the cluster. For the jrSiCKLSNMF analyses, we used 3 GB of RAM per node, and for the other methods, we used 10 GB of RAM on the high performance cluster for most analyses. BREM-SC sometimes had high RAM requirements and failed to run on all datasets, so we tested using up to 50 GB when needed. BREM-SC also required the manual re-setting of the random seed when it failed to converge for certain datasets. In addition to Table 2, in Figure 3, we also provide boxplots of results from our method along with results from other `R`-based methods. Not only is our method more accurate, it also has a very low variability, indicating that it works similarly over many different datasets. We can also see from this that our method is robust to increased noise; jrSiCKLSNMF, with graph Laplacian $\mathbf{L}^v$ constructed from each individual dataset's feature-feature similarity and a sparsity constraint on $\mathbf{H}$, consistently outperforms other methods for all noise levels.

3.5 Real data example

For our real data example, we use cell line dataset, GSE126074, which includes 1,047 cells from the H1, BJ, K562, and GM12878 cell lines. This dataset is not labeled with the true cell types; however, an R script to generate two sets of cell annotations for the dataset was graciously provided by Professor Song Chen, the first author of the paper describing SNARE-seq (Chen et al., 2019). To annotate the cells, Chen et al. (2019) separately cluster and then annotate the cells in the ATAC modality using cisTopic (Bravo González-Blas et al., 2019) and in the RNA modality using Pagoda2 (Barkas et al., 2021). The ARI between these two annotations was 0.867. We will compare our clustering results to these annotations. Since the data are already pre-processed, we remove 0 cells from the dataset. There are 18,666 genes and 136,771 peaks. We select genes and bins which appear in at least 10 cells and are left with 9,000 genes and 24,514 peaks. In Figure 4, we compare the performance on this dataset of jrSiCKLSNMF with a unit L2 norm constraint on the rows of $\mathbf{H}$ to the dimension reduction generated by Seurat's WNN. From these images, we can see that our dimension-reduction method does a better job of separating the cell types into distinct clusters in the UMAP space; one can easily see from this graph that there are 4 clusters. On the other hand, for the Seurat dimension reduction, H1-hESC is clearly separated from the other 3 cell types, but the clusters K-562, BJ, and GM12878 are very close in the UMAP space. Without these color annotations, it could be interpreted as one oblong cluster. Our clustering results are also better. After performing k-means on the H matrix generated here, we achieve an ARI of 0.923 with the

TABLE 2 Here, we provide the mean ARI, median ARI, standard deviation of ARI, and the mean running time for BREM-SC, Citefuse, jrSiCKLSNMF-B:L2, jrSiCKLSNMF-B:SH, jrSiCKLSNMF-I:L2,jrSiCKLSNMF-I:SH, MOFA+, scAI, and Seurat on 400 simulated datasets (100 datasets in each of 4 noise conditions). Bold entries indicate the best performance in each column. Note that these times include all normalization and pre-processing steps required to run each algorithm. We use the Seurat normalization workflow to normalize the data for MOFA+, so Seurat normalization is included as part of its computation time. A variant of jrSiCKLSNMF performs best for all examples, and CiteFuse, when compared using all pre-processing steps, has the fastest performance. Bold values indicate the best performing algorithm per column.

No added noise	Mean	Median	Standard deviation	Mean time
BREM-SC	0.827	0.913	0.178	615.75 s
CiteFuse	0.330	0.330	0.0449	**6.87 s**
jrSiCKLSNMF-B:L2	0.949	0.960	0.0540	84.46 s
jrSiCKLSNMF-B:SH	0.974	0.980	0.0199	65.82 s
jrSiCKLSNMF-I:L2	**0.992**	**0.990**	0.0095	60.04 s
jrSiCKLSNMF-I:SH	0.988	**0.990**	0.0127	46.77 s
MOFA+	0.321	0.320	0.0971	11.39 s
scAI	0.021	0.014	0.0240	149.25 s
Seurat	0.767	0.783	0.0857	31.01 s
$\mathcal{U}(1, 1.25), \mathcal{N}(-0.25, 0.25)$	Mean	Median	Standard Deviation	Mean Time
BREM-SC	0.427	0.505	0.179	1,181.96 s
CiteFuse	0.303	0.314	0.0540	**8.89 s**
jrSiCKLSNMF-B:L2	0.952	0.960	0.0327	84.93 s
jrSiCKLSNMF-B:SH	0.961	0.965	0.0223	58.43 s
jrSiCKLSNMF-I:L2	**0.982**	**0.980**	0.0149	78.32 s
jrSiCKLSNMF-I:SH	0.977	**0.980**	0.0180	37.52 s
MOFA+	0.306	0.312	0.112	10.24 s
scAI	0.024	0.015	0.0268	160.93 s
Seurat	0.694	0.710	0.0947	36.87 s
$\mathcal{U}(1, 1.5), \mathcal{N}(-0.5, 0.5)$	Mean	Median	Standard Deviation	Mean Time
BREM-SC	0.047	0.022	0.0934	1,253.69 s
CiteFuse	0.182	0.189	0.0520	**9.23 s**
jrSiCKLSNMF-B:L2	0.372	0.375	0.0945	115.72 s
jrSiCKLSNMF-B:SH	0.310	0.303	0.0821	49.31s
jrSiCKLSNMF-I:L2	0.465	0.467	0.112	123.00s
jrSiCKLSNMF-I:SH	**0.702**	**0.711**	0.0895	43.77 s
MOFA+	0.215	0.220	0.0593	14.34 s
scAI	0.019	0.012	0.0192	165.11 s
Seurat	0.387	0.394	0.0895	34.54 s
$\mathcal{U}(1, 2), \mathcal{N}(-1, 1)$	Mean	Median	Standard Deviation	Mean Time
BREM-SC*	0.017	0.009	0.0235	1,277.76 s
CiteFuse	0.130	0.130	0.0509	**8.10** s
jrSiCKLSNMF-B:L2	0.152	0.145	0.0453	197.28 s
jrSiCKLSNMF-B:SH	0.141	0.134	0.0393	82.47 s
jrSiCKLSNMF-I:L2	0.200	0.199	0.0380	123.22 s
jrSiCKLSNMF-I:SH	**0.295**	**0.278**	0.0943	59.44 s
MOFA+	0.143	0.140	0.0395	24.79 s
scAI	0.018	0.012	0.0194	158.97 s
Seurat	0.204	0.188	0.0751	57.36

*BREM-SC fails to run on the 99th simulated dataset with added noise $\mathcal{U}(1, 2)$ in the scRNA-seq modality and $\mathcal{N}(-1, 1)$ in the scATAC-seq modality. The values displayed in this row exclude the 99th simulated dataset.

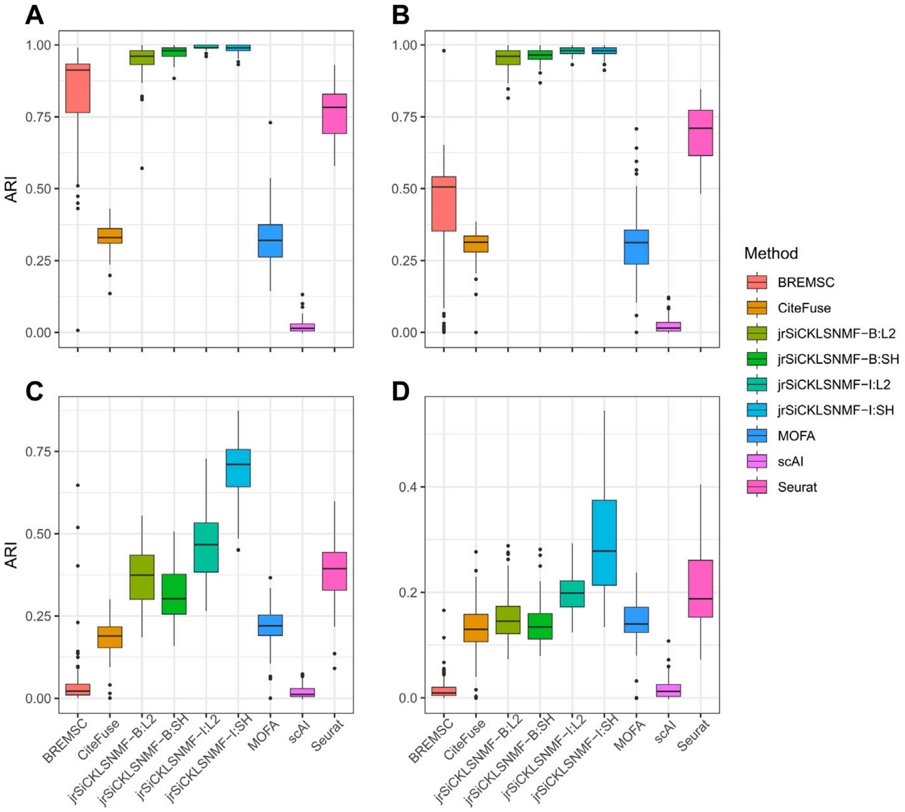


FIGURE 3
Comparison of different versions of jrSiCKLSNMF to other methods. A "B" in the method indicates that the regularizing graph is generated from bulk data while an "I" indicates that the regularizing graph is generated from the data itself. "SH" indicates that a sparsity parameter is included on **H** while "L2" indicates that the L2 norms of the rows of **H** are equal to 1. For all simulations, we generate 10 latent NMF factors. For all "SH," $\lambda \mathbf{W}^{RNA} = 10$, $\lambda \mathbf{W}^{ATAC} = 50$, $\lambda_H = 500$. For all "L2," $\lambda \mathbf{W}^{RNA} = 3$, $\lambda \mathbf{W}^{ATAC} = 15$. **(A)** Data simulated for the RNA and ATAC modalities from SPARSim and SimATAC, respectively, with no added noise. **(B)** The gene variability parameter is increased by up to 25% in the RNA simulation and noise simulated from $\mathcal{N}(-0.25, 0.25)$ distribution is added to the ATAC simulation. **(C)** The gene variability parameter is increased by up to 50% in the RNA simulation and noise simulated from $\mathcal{N}(-0.5, 0.5)$ distribution is added to the ATAC simulation. **(D)** The gene variability parameter is increased by up to 100% in the RNA simulation and noise simulated from $\mathcal{N}(-1, 1)$ distribution is added to the ATAC simulation. Note that here, BREMSC is unable to run on dataset 99.

annotations based on the RNA modality and 0.916 with the annotations based on the ATAC modality. For the Seurat multimodal WNN analysis, the ARI is 0.876 for the RNA modality and 0.872 for the ATAC modality.

We further use jrSiCKLSNMF to visualize data in the RNA modality and the ATAC modality by performing UMAP on the products $\mathbf{W}^{RNA}\mathbf{H}$ and $\mathbf{W}^{ATAC}\mathbf{H}$, respectively. In Figure 5, we plot the UMAP of $\mathbf{W}^{RNA}\mathbf{H}$ in (A), the UMAP of $\mathbf{W}^{ATAC}\mathbf{H}$ in (C) and compare it to the dimension reduction in Seurat based on the RNA modality alone (B) and the ATAC modality alone (C). The annotations for A and B correspond to the annotations derived purely from the RNA modality while the annotations for (C) and (D) correspond to the annotations derived purely from the ATAC modality. From this, in the first row, we can see that the Seurat UMAP on the RNA dimension reduction almost perfectly captures the four cell types identified by the annotation while our method does not have as clear of a separation in the RNA modality. However, for the ATAC modality, the UMAP of the Seurat dimension reduction fails to capture differences between BJ cells and K-562 cells in the first 2 UMAP dimensions. However, jrSiCKLSNMF is able to capture this difference better: there is a separation between the bulk of the BJ cells and the bulk of the K-562 cells.

The plotting performance of jrSiCKLSNMF using the L2 norm constraint is a bit more robust to specifying a larger D and obtains slightly better results than jrSiCKLSNMF with a sparsity constraint on **H**. To determine an appropriate number of D and k, we use diagnostic plots implemented in the jrSiCKLSNMF package. In Figure 6A, for $\lambda_{W^{RNA}} = 10, \lambda_{W^{ATAC}} = 50, \lambda_H = 500$ we show a plot of the loss function vs. D for 2 to 20 factors. We recommend identifying an appropriate elbow. Here, we identify 5 as an appropriate number of factors. After convergence, we perform diagnostics to determine an appropriate number of clusters. In Figure 6B, we provide a representative plot of the silhouette method (the plots using the gap statistic and within sum of squares method are available in the Supplementary Figure S5 while the output from CLVALID is in Supplementary Table S1). Then, in Figure 7A, we provide a UMAP plot colored by the k-means clusters with number of clusters $k = 5$. In Figure 7B, we provide a UMAP plot colored by clusters determined by k-means using the true number of clusters (4). Figures 7C, D show the RNA

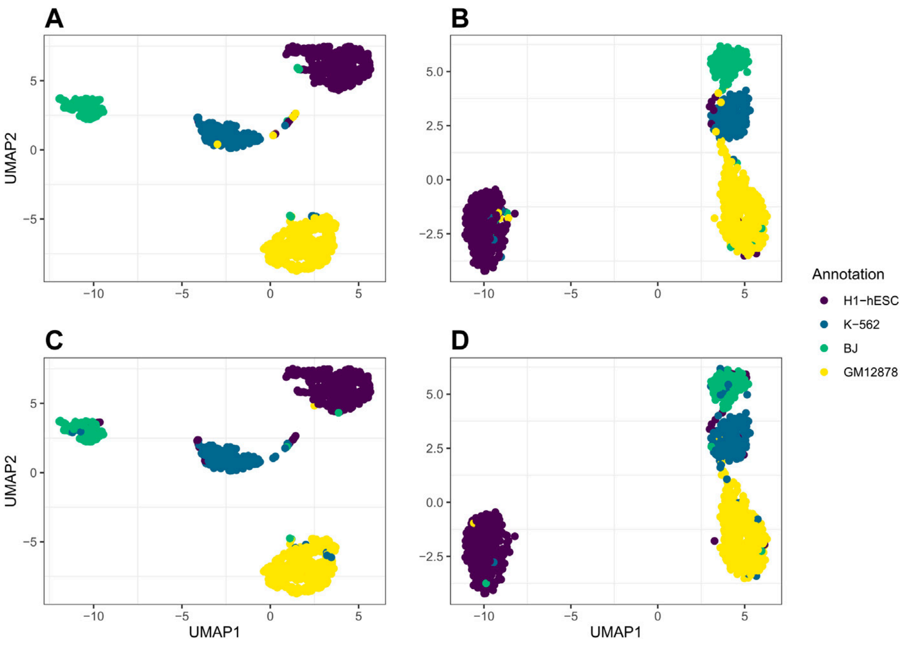


FIGURE 4
Comparison of UMAP graphs of the H matrix generated by jrSiCKLSNMF with $D = 10$, $\lambda W^{RNA} = 3$, $\lambda W^{ATAC} = 15$, and the unit L2 norm constraint on the rows of **H** **(A,C)** to the Seurat WNN dimension reduction **(B,D)**. The colors of the points in A and B correspond to the generated cell annotations from the RNA modality while the colors of the points in C and D correspond to the ATAC modality.

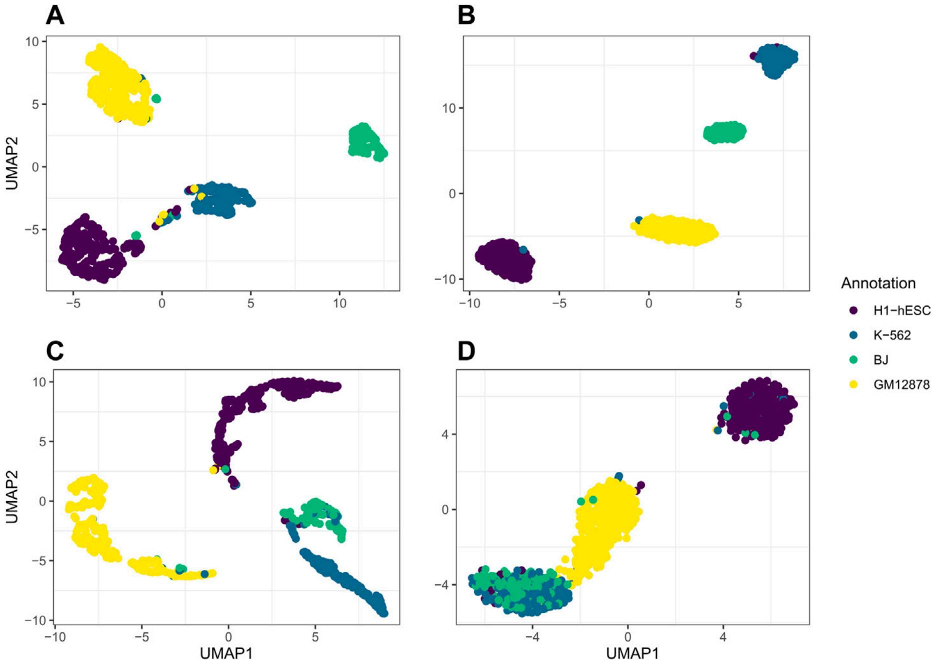


FIGURE 5
(A) is the UMAP of the product of $\mathbf{W}^{\mathbf{RNA}}\mathbf{H}$ generated by jrSiCKLSNMF with $D = 10$, $\lambda W^{RNA} = 3$, $\lambda W^{ATAC} = 15$, and the unit L2 norm constraint on the rows of **H**, **(B)** is Seurat's dimension reduction of the RNA modality alone, **(C)** is the UMAP of the product of $\mathbf{W}^{\mathbf{ATAC}}\mathbf{H}$, and **(D)** is Seurat's dimension reduction of the ATAC modality alone. The colors of the annotations for A and B correspond to the generated cell annotations in the RNA modality while the colors of the annotations for C and D correspond to the generated cell annotations in the ATAC modality.

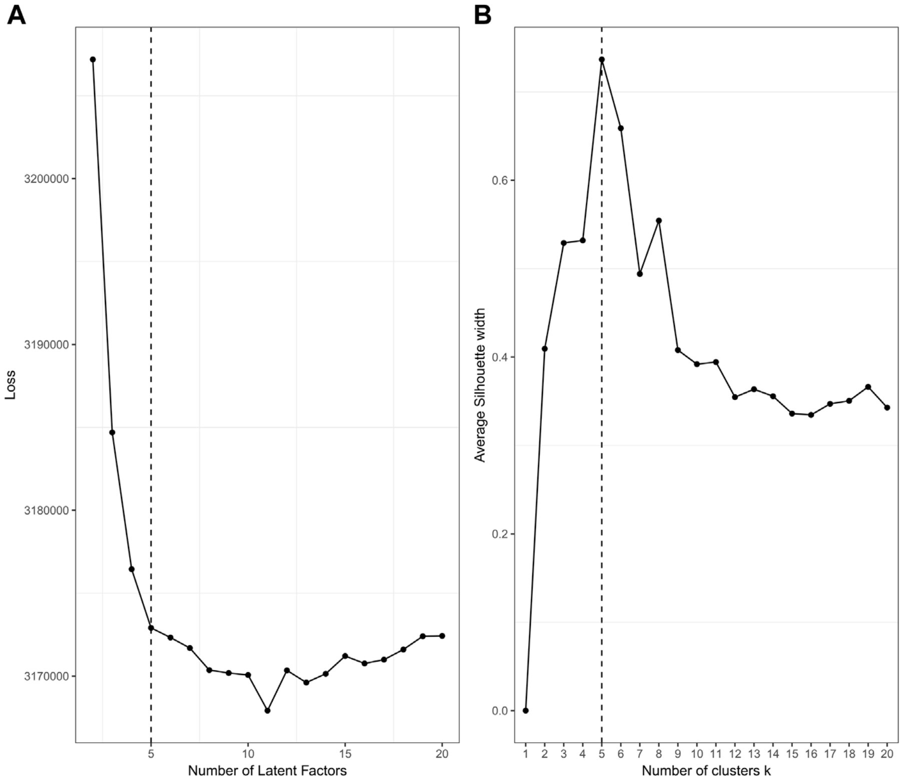


FIGURE 6
Diagnostic plots for jrSiCKLSNMF to determine the number of latent factors **(A)** and to determine the number of clusters **(B)**, with $\lambda_{W^{RNA}} = 10$, $\lambda_{W^{ATAC}} = 50$, and $\lambda_H = 500$. In **(A)**, the value of the loss function is after 100 iterations of jrSiCKLSNMF. In **(B)**, we show diagnostics for the silhouette score. Here, the dashed line indicates the ideal number of latent factors and number of clusters, which we determine to be 5 for both of these the number of factors and the number of clusters are coincidentally determined to be equal here. The true number of clusters is 4.

and ATAC annotations, respectively. Even though we determined an incorrect true number of clusters, the ARI dropped only from 0.904 to 0.885 in the RNA modality and from 0.918 to 0.875 in the ATAC modality.

4 Discussion

jrSiCKLSNMF is a promising method for the analysis of multimodal single-cell count data with many useful properties. First, this method utilizes all features shared across a pre-specified threshold of cells rather than a small subset of highly-variable features. We also do not introduce bias by performing $\log(x + 1)$ normalization and therefore preserve the count nature of the data in each modality (Townes et al., 2019; Elyanow et al., 2020). This NMF method can provide an intuitive way to summarize and describe data. There is potential for the use of jrSiCKLSNMF in the visualizations of multimodal data because it can extract relevant latent factors from high dimensional data and also provide a method of data compression.

For smaller datasets (i.e., $N \ll M^v$), we recommend using the I-SH variant of our algorithm. It is not recommended to construct KNN graphs from data where $N > M^v$ or $N \approx M^v$ because KNN is unreliable in these situations. In this case, we recommend constructing the KNN graph from bulk data or using a graph that is not based on the Euclidean distance. Additionally, when not confident about the number of latent factors, the L2 Norm regularization appears to be slightly more robust to choice of D for visualization purposes. Therefore, we recommend using it as a secondary method of data analysis if desired.

Though we show that our method performs well for cell-type clustering, even in the presence of increasing noise, there are a few limitations. These limitations can serve as directions for future research. Firstly, optimizing the choice of λ values is not trivial. Through extensive validation, when k-means is used to cluster **H**, we find that for both our simulated data and the real dataset, using $\lambda W^{RNA} = 10$, $\lambda W^{ATAC} = 50$, and $\lambda_H = 500$ work well when using a sparsity constraint on **H**, and using $\lambda W^{RNA} = 3$ and $\lambda W^{ATAC} = 15$ work well when enforcing a unit norm constraint on the L2 norms of the rows of **H**. However, we also find that, even when using λ values that are sub-optimal for clustering using k-means, jrSiCKLSNMF still can extract meaningful factors. We find that for some combinations of λ values where k-means performs poorly, the UMAP plot was still accurate, and Louvain clustering performs well.

While the *post hoc* clustering remains accurate while varying the number of latent factors D, the performance of the visualization using the first two UMAP dimensions depends on appropriate

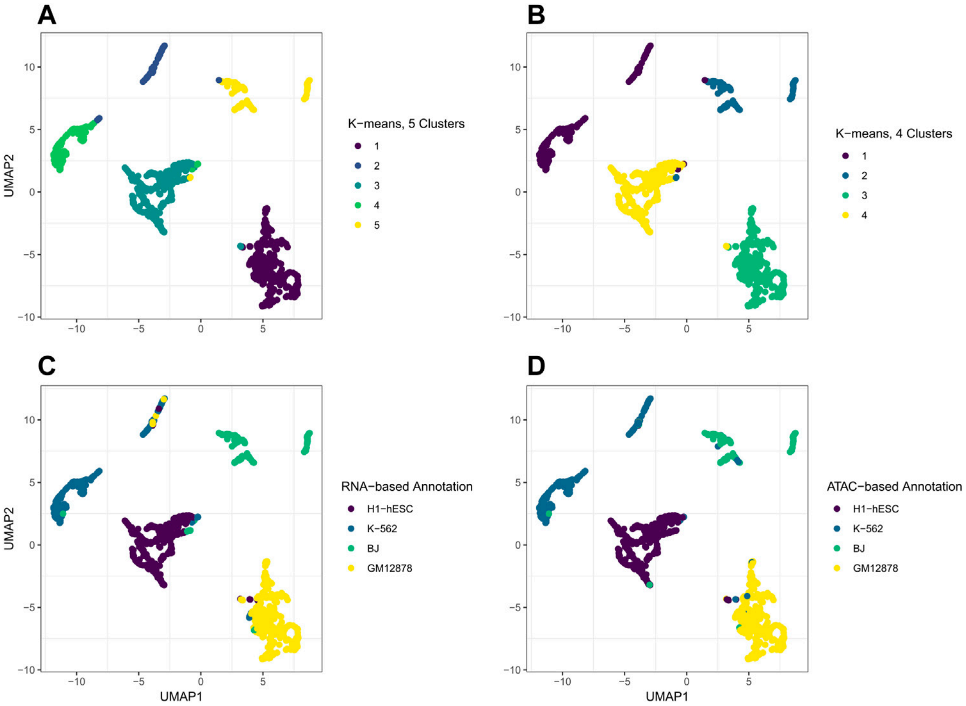


FIGURE 7
UMAP plots of the **H** matrix that resulted from jrSiCKLSNMF with $D = 5$, $\lambda_{W^{RNA}} = 10$, $\lambda_{W^{ATAC}} = 50$, and $\lambda_H = 500$, colored by various clustering or annotation results. **(A)** shows the results of clustering **H** using k-means with $k = 5$, as determined by the diagnostic plots in Figure 6. The ARI of these clusters with the RNA annotation is 0.885 and the ATAC annotation is 0.876. **(B)** shows the results of clustering **H** using k-means with $k = 4$, the correct number of cell types. The ARI of these clusters with the RNA annotation is 0.904 and with the ATAC annotation is 0.918. **(C)** plots the UMAP with colors based on the RNA annotations while **(D)** plots the UMAP with colors based on the ATAC annotations.

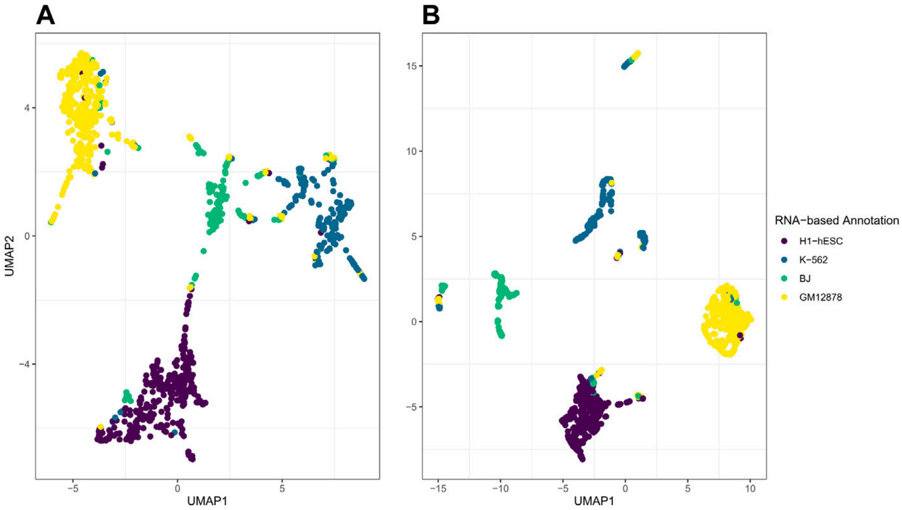


FIGURE 8
Illustrations of jrSiCKLS-NMF with too much noise captured in the generated **H**. On the left in **(A)** is the UMAP generated from **H** when $D = 20$, $\lambda_{W^{RNA}} = 3, \lambda_{W^{RNA}} = 15$ with the unit L2 norm constraint on the rows of **H**, while on the right in **(B)**, $D = 10$ and $\lambda_{W^{RNA}} = 10, \lambda_{W^{ATAC}} = 50, \lambda_H = 500$.

selection of D. In Figure 8, on the left in A, we illustrate what happens when the number of latent factors D is too large for the variation to be captured within the first 2 elements of UMAP. Here, we set $D = 20$, set $\lambda_{W^{RNA}} = 3$, $\lambda_{W^{ATAC}} = 15$, $\lambda_H = 0$ and used the unit constraint on the L2 norm of the rows of **H**. In contrast to A and C in Figure 4 where the clusters are well-separated in the UMAP space, here, except for GM12878, the clusters are not as clearly separated. Similarly, in B, which is generated from jrSiCKLSNMF with $D = 10$,

$\lambda_{W^{RNA}} = 10$, $\lambda_{W^{ATAC}} = 50$, and $\lambda_H = 500$, while H1-hESC and GM12878 form distinct clusters, K-562 and BJ appear to form multiple smaller clusters. If we contrast these plots with those in Figures 4, 7, we see that these results capture more noise. Although the plots of the first and second UMAP dimension are not ideal, the clusters determined by using k-means on the respective **H** matrices are still accurate. Optimizing *D* is not trivial and is still an active area of research for NMF (Maisog et al., 2021). While we do provide this method to determine an appropriate *D* visually as in Figure 6A, future research will further address this gap and potentially identify more suitable approaches for the selection of *D*.

Additionally, although our method outperforms existing methods in terms of accurately identifying clusters by a wide margin, the algorithmic implementation can be slower than desirable, especially when we need to determine an appropriate number of latent factors *D* and clusters *k*. Since the methods to determine the number of latent factors *D* and clusters *k* for any of the methods used on simulated data as outlined in Table 2 require pre-specification, for this simulation study, we use a fixed $D = 10$ for our method and the known $k = 3$ for all methods, except for Seurat, which requires a resolution parameter. We therefore fix Seurat's resolution parameter to a value which consistently results in 3 clusters. Therefore, for these time trials, we do not include the time required to determine the number of clusters for any method or the number of latent factors for our method. For large datasets, this means that it can be computationally demanding to use jrSiCKLSNMF. Although we have implemented sparse matrix functions to decrease memory load and increase speed, methods such as implementing a more efficient descent algorithm than MU, or exploring also using online algorithms as in the 2021 version of LIGER (Gao et al., 2021) may help to improve performance. Moreover, the choice of the KL-divergence itself has some drawbacks. Compared to the wide variety of methods that leverage block coordinate descent to increase the convergence speed of NMF algorithms that use the Frobenius norm, since the KL-divergence is not differentiable for **W** or **H** when $(\mathbf{WH})_{ij} = 0$, the KL-divergence lacks the appropriate smoothness requirements to implement block coordinate descent in many cases (Hien and Gillis, 2021). This adds restrictions to the extension of block coordinate descent to KL-NMF algorithms. Hien and Gillis (2021) further discuss that while MU is slow and should not be used in Frobenius NMF algorithms, MU is one of the three most reliable algorithms of the seven descent algorithms for KL-NMF compared in their work. Furthermore, as the technology progresses, datasets will become even larger and will contain more diverse cell types. Testing on a larger number of cell types may have other computational issues. Future works will focus on improving these computational aspects.

Finally, in this work, other than a brief discussion of using $\mathbf{W}^{\nu}\mathbf{H}$ to visualize data in different modalities, we do not address potential applications of the $\mathbf{W}^{\nu}$ matrices. Since our focus is on the integration of data from different modalities for the same set of single cells, discussion of applications of $\mathbf{W}^{\nu}$ is outside of the scope of this work. $\mathbf{W}^{\nu}$ belongs to the feature space rather than the observation space. However, there are many interesting potential avenues for future research involving these $\mathbf{W}^{\nu}$ matrices. One such potential application, with which we have had some preliminary success, is using the weighted average of $(\mathbf{W}\nu)^{+}\mathbf{X}^{\nu}_{new}$, where $(\mathbf{W}^{\nu})^{+}$ is the pseudoinverse of $\mathbf{W}^{\nu}$ fitted on the original data $\mathbf{X}^{\nu}$ and $\mathbf{X}^{\nu}_{new}$ is new data, to provide an approximation of $\mathbf{H}_{new}$, the latent factor observation matrix for the new observations. Other such applications include using $\mathbf{W}^{\nu}$ to identify co-expressed features or constructing feature networks and exploring whether $\mathbf{W}^{\nu}\mathbf{H}$ can have applications in downstream analyses like network analysis at the single-cell level.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

Author contributions

DE designed the study. SD and AR provided theoretical support when required, DE implemented the simulation and the analyses, DE wrote the manuscript. All authors contributed to the article and approved the submitted version.

Funding

This work was partially supported by NIH grant 1UL1TR000064 from the Center for Scientific Review.

Acknowledgments

The authors acknowledge University of Florida Research Computing for providing computational resources and support that have contributed to the research results reported in this publication. URL: http://www.rc.ufl.edu. Special thanks to Song Chen of the Sanger Institute for providing scripts to generate annotations for the cell line mix experiment for GSE126074.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2023.1179439/full#supplementary-material

References

Akata, Z., Thurau, C., and Bauckhage, C. (2011). "Non-negative matrix factorization in multi-modality data for segmentation and label Prediction," in 16th Computer Vision Winter Workshop, Mitterberg, Austria, February 2-4, 2011, 652879.

Argelaguet, R., Arnol, D., Bredikhin, D., Deloro, Y., Velten, B., Marioni, J. C., et al. (2020). MOFA+: A statistical framework for comprehensive integration of multi-modal single-cell data. *Genome Biol.* 21, 111–117. doi:10.1186/s13059-020-02015-1

Barkas, N., Petukhov, V., Karchenko, P., and Biederstedt, E. (2021). pagoda2: SIngle cell analysis and differential expression. Available at: https://cran.r-project.org/web/packages/pagoda2/pagoda2.pdf (Accessed October 14, 2022).

Baruzzo, G., Patuzzi, I., and Camillo, B. D. (2020). SPARSim single cell: A count data simulator for scRNA-seq data. *Bioinformatics* 36, 1468–1475. doi:10.1093/bioinformatics/btz752

Boutsidis, C., and Gallopoulos, E. (2008). SVD based initialization: A head start for nonnegative matrix factorization. *Pattern Recognit.* 41, 1350–1362. doi:10.1016/J.PATCOG.2007.09.010

Bravo González-Blas, C., Minnoye, L., Papasokrati, D., Aibar, S., Hulselmans, G., Christiaens, V., et al. (2019). cisTopic: cis-regulatory topic modeling on single-cell ATAC-seq data. *Nat. Methods* 16, 397–400. doi:10.1038/s41592-019-0367-1

Brock, G., Pihur, V., Datta, S., and Datta, S. (2008). clValid: An R package for cluster validation. *J. Stat. Softw.* 25, 1–22. doi:10.18637/jss.v025.i04

Buenrostro, J. D., Wu, B., Litzenburger, U. M., Ruff, D., Gonzales, M. L., Snyder, M. P., et al. (2015). Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature* 523, 486–490. doi:10.1038/nature14590

Cai, D., He, X., Han, J., and Huang, T. S. (2011). Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. Pattern Analysis Mach. Intell.* 33, 1548–1560. doi:10.1109/TPAMI.2010.231

Cai, D., He, X., Wu, X., and Han, J. (2008). "Non-negative matrix factorization on manifold," in Proceedings - IEEE International Conference on Data Mining, ICDM, Pisa, Italy, December 15-19, 2008, 63–72.

Cao, J., Cusanovich, D. A., Ramani, V., Aghamirzaie, D., Pliner, H. A., Hill, A. J., et al. (2018). Joint profiling of chromatin accessibility and gene expression in thousands of single cells. *Science* 361, 1380–1385. doi:10.1126/science.aau0730

Chalise, P., and Fridley, B. L. (2017). Integrative clustering of multi-level 'omic data based on non-negative matrix factorization algorithm. *PLOS ONE* 12, 01762788–e176318. doi:10.1371/journal.pone.0176278

Charrad, M., Ghazzali, N., Boiteau, V., and Niknafs, A. (2014). NbClust: An R package for determining the relevant number of clusters in a data set. *J. Stat. Softw.* 61, 1–36. doi:10.18637/jss.v061.i06

Chen, S., Lake, B. B., and Zhang, K. (2019). High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nat. Biotechnol.* 37, 1452–1457. doi:10.1038/s41587-019-0290-0

Costa Dos Santos, G., Renovato-Martins, M., and Mesquita De Brito, N. (2021). The remodel of the "central dogma": A metabolomics interaction perspective. *Metabolomics* 17, 48. doi:10.1007/s11306-021-01800-8

Dai, L. Y., Zhu, R., and Wang, J. (2020). Joint nonnegative matrix factorization based on sparse and graph laplacian regularization for clustering and Co-differential expression genes analysis. *Complexity* 2020, 1–10. doi:10.1155/2020/3917812

Douglas, S. C., Amari, S. I., and Kung, S. Y. (2000). On gradient adaptation with unit-norm constraints. *IEEE Trans. Signal Process.* 48, 1843–1847. doi:10.1109/78.845952

Eddelbuettel, D., and François, R. (2011). Rcpp: Seamless R and C++ integration. *J. Stat. Softw.* 40, 1–18. doi:10.18637/jss.v040.i08

Eddelbuettel, D., and Sanderson, C. (2014). RcppArmadillo: Accelerating R with high-performance C++ linear algebra. *Comput. Statistics Data Analysis* 71, 1054–1063. doi:10.1016/j.csda.2013.02.005

Ellis, D., Wu, D., and Datta, S. (2021). Sarev: A review on statistical analytics of single-cell rna sequencing data. *Wiley Interdiscip. Rev. Comput. Stat.* 14, e1558. doi:10.1002/WICS.1558

Elyanow, R., Dumitrascu, B., Engelhardt, B. E., and Raphael, B. J. (2020). NetNMF-SC: Leveraging gene-gene interactions for imputation and dimensionality reduction in single-cell expression analysis. *Genome Res.* 30, 195–204. doi:10.1101/gr.251603.119

Esposito, F. (2021). A review on initialization methods for nonnegative matrix factorization: Towards omics data experiments. *Mathematics* 9, 1006. doi:10.3390/MATH9091006

Févotte, C., Bertin, N., and Durrieu, J. L. (2009). Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis. *Neural Comput.* 21, 793–830. doi:10.1162/NECO.2008.04-08-771

Févotte, C., and Idier, J. (2011). Algorithms for nonnegative matrix factorization with the β-divergence. *Neural Comput.* 23, 2421–2456. doi:10.1162/NECO_a_00168

Fu, X., Huang, K., Sidiropoulos, N. D., and Ma, W.-K. (2019). Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications. *IEEE Signal Process. Mag.* 36, 59–80. doi:10.1109/MSP.2018.2877582

Fu, X., Huang, K., and Sidiropoulos, N. D. (2018). On identifiability of nonnegative matrix factorization. *IEEE Signal Process. Lett.* 25, 328–332. doi:10.1109/LSP.2018.2789405

Gao, C., Liu, J., Kriebel, A. R., Preissl, S., Luo, C., Castanon, R., et al. (2021). Iterative single-cell multi-omic integration using online learning. *Nat. Biotechnol.* 1, 1000–1007. doi:10.1038/s41587-021-00867-x

Gillis, N., and Rajkó, R. (2023). Partial identifiability for nonnegative matrix factorization. *SIAM J. Matrix ANalysis Appl.* 44, 27–52. doi:10.1137/22M1507553

Gillis, N. (2012). Sparse and unique nonnegative matrix factorization through data preprocessing. *J. Mach. Learn. Res.* 13, 3349–3386. doi:10.5555/2503308.2503349

Greene, D., and Cunningham, P. (2009). "A matrix factorization approach for integrating multiple data views," in *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Berlin: Springer), 423–438.

Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W. M., Zheng, S., Butler, A., et al. (2021). Integrated analysis of multimodal single-cell data. *Cell.* 184, 3573–3587.e29. doi:10.1016/J.CELL.2021.04.048

Haque, A., Engel, J., Teichmann, S. A., and Lönnberg, T. (2017). A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. *Genome Med.* 9, 75–12. doi:10.1186/s13073-017-0467-4

Hien, L. T. K., and Gillis, N. (2021). Algorithms for nonnegative matrix factorization with the Kullback-Leibler divergence. *J. Sci. Comput.* 87, 93. doi:10.1007/s10915-021-01504-0

Hubert, L., and Arabie, P. (1985). Comparing partitions. *J. Classif.* 2, 193–218. doi:10.1007/BF01908075

Jin, S., Zhang, L., and Nie, Q. (2020). scAI: An unsupervised approach for the integrative analysis of parallel single-cell transcriptomic and epigenomic profiles. *Genome Biol.* 21, 25–19. doi:10.1186/s13059-020-1932-8

Kim, H. J., Lin, Y., Geddes, T. A., Yee, J., Yang, H., and Yang, P. (2020). CiteFuse enables multi-modal analysis of CITE-seq data. *Bioinformatics* 36, 4137–4143. doi:10.1093/bioinformatics/btaa282

Kimura, K., and Yoshida, T. (2011). "Non-negative matrix factorization with sparse features," in Proceedings - 2011 IEEE International Conference on Granular Computing, GrC, Kaohsiung, Taiwan, November 8-10, 2011, 324–329. doi:10.1109/GRC.2011.6122616

Krassowski, M., Das, V., Sahu, S. K., and Misra, B. B. (2020). State of the field in multi-omics research: From computational needs to data mining and sharing. *Front. Genet.* 11, 1598. doi:10.3389/fgene.2020.610798

Le Roux, J., Weniger, F., and Hershey, J. R. (2015). Tech. rep. Mitsubishi Electric Research Laboratories (MERL),.Sparse NMF: Half-baked or well done?

Lee, D. D., and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 788–791. doi:10.1038/44565

Lee, J., Hyeon, D. Y., and Hwang, D. (2020). Single-cell multiomics: Technologies and data analysis methods. *Exp. Mol. Med.* 52, 1428–1442. doi:10.1038/s12276-020-0420-2

Li, J. J., and Biggin, M. D. (2015). Gene expression. Statistics requantitates the central dogma. *Science* 347, 1066–1067. doi:10.1126/SCIENCE.AAA8332

Liu, J., Gao, C., Sodicoff, J., Kozareva, V., Macosko, E. Z., and Welch, J. D. (2020). Jointly defining cell types from multiple single-cell datasets using LIGER. *Nat. Protoc.* 15, 3632–3662. doi:10.1038/s41596-020-0391-8

Liu, J., Wang, C., Gao, J., and Han, J. (2013). "Multi-view clustering via joint nonnegative matrix factorization," in Proceedings of the 2013 SIAM International Conference on Data Mining, Austin, Texas, May 2-4, 2013, 252–260.

Lun, A. T., Bach, K., and Marioni, J. C. (2016). Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biol.* 17, 75–14. doi:10.1186/s13059-016-0947-7

Luo, Y., Mao, C., Yang, Y., Wang, F., Ahmad, F. S., Arnett, D., et al. (2019). Integrating hypertension phenotype and genotype with hybrid non-negative matrix factorization. *Bioinformatics* 35, 1395–1403. doi:10.1093/bioinformatics/bty804

Ma, S., Zhang, B., LaFave, L. M., Earl, A. S., Chiang, Z., Hu, Y., et al. (2020). Chromatin potential identified by shared single-cell profiling of RNA and chromatin. *Cell.* 183, 1103–1116. doi:10.1016/J.CELL.2020.09.056

Maisog, J. M., Demarco, A. T., Devarajan, K., Young, S., Fogel, P., and Luta, G. (2021). Assessing methods for evaluating the number of components in non-negative matrix factorization. *Math. (Basel, Switz.* 9, 2840. doi:10.3390/MATH9222840

McInnes, L., Healy, J., Saul, N., and Großberger, L. (2018). Umap: Uniform manifold approximation and projection. *J. Open Source Softw.* 3, 861. doi:10.21105/JOSS.00861

Merris, R. (1994). Laplacian matrices of graphs: A survey. *Linear Algebra its Appl.* 197-198, 143–176. doi:10.1016/0024-3795(94)90486-3

Miao, Z., Humphreys, B. D., McMahon, A. P., and Kim, J. (2021). Multi-omics integration in the age of million single-cell data. *Nat. Rev.* 17, 710–724. doi:10.1038/s41581-021-00463-x

Navidi, Z., Zhang, L., and Wang, B. (2021). simATAC: a single-cell ATAC-seq simulation framework. *Genome Biol.* 22, 74–16. doi:10.1186/s13059-021-02270-w

Ogbeide, S., Giannese, F., Mincarelli, L., and Macaulay, I. C. (2022). Into the multiverse: Advances in single-cell multiomic profiling. *Trends Genet.* 38, 831–843. doi:10.1016/J.TIG.2022.03.015

Park, M., Keung, A. J., and Khalil, A. S. (2016). The epigenome: The next substrate for engineering. *Genome Biol.* 17, 183. doi:10.1186/S13059-016-1046-5

Peng, S., Lin, Z., and Chen, B. (2019). "Dual graph regularized sparse nonnegative matrix factorization for data representation," in IEEE International Symposium on Circuits and Systems, Sapporo, Japan, May 26-29, 2019, 1–5. doi:10.1109/ISCAS.2019.8702585

Qiao, H. (2015). New SVD based initialization strategy for non-negative matrix factorization. *Pattern Recognit. Lett.* 63, 71–77. doi:10.1016/J.PATREC.2015.05.019

Quinn, T. P., Richardson, M. F., Lovell, D., and Crowley, T. M. (2017). propr: An R-package for identifying proportionally abundant features using compositional data analysis. *Sci. Rep.* 7, 16252. doi:10.1038/s41598-017-16520-0

R Core Team (2022). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

Rappoport, N., and Shamir, R. (2018). Multi-omic and multi-view clustering algorithms: Review and cancer benchmark. *Nucleic Acids Res.* 46, 10546–10562. doi:10.1093/nar/gky889

Reel, P. S., Reel, S., Pearson, E., Trucco, E., and Jefferson, E. (2021). Using machine learning approaches for multi-omics data analysis: A review. *Biotechnol. Adv.* 49, 107739. doi:10.1016/j.biotechadv.2021.107739

Satija, R., Farrell, J. A., Gennert, D., Schier, A. F., and Regev, A. (2015). Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.* 33, 495–502. doi:10.1038/nbt.3192

Shiga, M., Seno, S., Onizuka, M., and Matsuda, H. (2021). SC-JNMF: Single-cell clustering integrating multiple quantification methods based on joint non-negative matrix factorization. *PeerJ* 9, e12087. doi:10.7717/peerj.12087

Stoeckius, M., Hafemeister, C., Stephenson, W., Houck-Loomis, B., Chattopadhyay, P. K., Swerdlow, H., et al. (2017). Simultaneous epitope and transcriptome measurement in single cells. *Nat. Methods* 14, 865–868. doi:10.1038/NMETH.4380

Stuart, T., Srivastava, A., Madad, S., Lareau, C. A., and Satija, R. (2021). Single-cell chromatin state analysis with Signac. *Nat. Methods* 18, 1333–1341. doi:10.1038/s41592-021-01282-5

Subramanian, I., Verma, S., Kumar, S., Jere, A., and Anamika, K. (2020). Multi-omics data integration, interpretation, and its application. *Bioinforma. Biol. Insights* 14, 1177932219899051–1177932219899059. doi:10.1177/1177932219899051

Swanson, E., Lord, C., Reading, J., Heubeck, A. T., Genge, P. C., Thomson, Z., et al. (2021). Simultaneous trimodal single-cell measurement of transcripts, epitopes, and chromatin accessibility using tea-seq. *eLife* 10, e63632. doi:10.7554/ELIFE.63632

Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., et al. (2009). mRNA-Seq whole-transcriptome analysis of a single cell. *Nat. Methods* 6, 377–382. doi:10.1038/nmeth.1315

Townes, F. W., Hicks, S. C., Aryee, M. J., and Irizarry, R. A. (2019). Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model. *Genome Biol.* 20, 295–316. doi:10.1186/s13059-019-1861-6

Ushey, K., Allaire, J. J., and Tang, Y. (2023). reticulate: Interface to 'Python'. Available at: https://cran.r-project.org/web/packages/reticulate/index.html (Accessed January 27, 2023)

Van Rossum, G., and Drake, F. L. (2009). *Python 3 reference manual*. Scotts Valley, CA: CreateSpace.

Wang, B., Mezlini, A. M., Demir, F., Fiume, M., Tu, Z., Brudno, M., et al. (2014). Similarity network fusion for aggregating data types on a genomic scale. *Nat. Methods* 11, 333–337. doi:10.1038/nmeth.2810

Wang, X., Sun, Z., Zhang, Y., Xu, Z., Xin, H., Huang, H., et al. (2020). BREM-SC: A bayesian random effects mixture model for joint clustering single cell multi-omics data. *Nucleic Acids Res.* 48, 5814–5824. doi:10.1093/NAR/GKAA314

Wang, Z., Kong, X., Fu, H., Li, M., and Zhang, Y. (2015). "Feature extraction via multi-view non-negative matrix factorization with local graph regularization," in 2015 IEEE International Conference on Image Processing (ICIP), Quebec, QC, Canada, September 27-30, 2015, 3500–3504.

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. 2nd Edn. Cham, Switzerland: Springer International Publishing. Available at: https://ggplot2.tidyverse.org/.

Xuan Vinh, N., Epps, J., and Bailey, J. (2009). "Information theoretic measures for clusterings comparison: Is a correction for chance necessary?," in Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09, Montreal Quebec. Canada, June 14 - 18, 2009.

Yan, F., Powell, D. R., Curtis, D. J., and Wong, N. C. (2020). From reads to insight: A hitchhiker's guide to ATAC-seq data analysis. *Genome Biol.* 21, 22–16. doi:10.1186/S13059-020-1929-3

Zheng, G. X., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., et al. (2017). Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* 8, 14049. doi:10.1038/NCOMMS14049

Zhou, L., Du, G., Lü, K., and Wang, L. (2021). A network-based sparse and multi-manifold regularized multiple non-negative matrix factorization for multi-view clustering. *Expert Syst. Appl.* 174, 114783. doi:10.1016/j.eswa.2021.114783

Zhu, C., Yu, M., Huang, H., Juric, I., Abnousi, A., Hu, R., et al. (2019). An ultra high-throughput method for single-cell joint analysis of open chromatin and transcriptome. *Nat. Struct. Mol. Biol.* 26, 1063–1070. doi:10.1038/S41594-019-0323-X