

# Scalable clustering of single-cell RNA-seq data with co-sparse non-negative matrix factorization

Tianci Song

Department of Computer Science and Engineering  
University of Minnesota Twin Cities  
Minneapolis, MN USA  
song0309@umn.edu

Arun Sharma

Department of Computer Science and Engineering  
University of Minnesota Twin Cities  
Minneapolis, MN USA  
sharm485@umn.edu

## ABSTRACT

After single-cell RNA sequencing (scRNA-seq) was first introduced to enable transcriptome profiling of individual cells in large scale, single-cell profiling technologies are now capable of capturing more than a million cells to provide a detailed mapping of the cell population in a biological sample. Since, scRNA-seq data analysis is becoming a routine step in biology and biomedical research, one of the major obstacle is limited scalability of the existing tools for analyzing such large-scale scRNA-seq datasets. In this study, we formulate a sparse non-negative matrix factorization (NMF) problem, called co-sparse non-negative matrix factorization (cosNMF) for exploring the sparseness of scRNA-seq data and computation of the factor matrices while improving scalability of clustering scRNA-seq data. The cosNMF formulation can be efficiently solved with alternative non-negative least square with block principal pivoting (ANLS-BPP) in the number of non-zero entries in the matrices in each iteration. In the experiment, we tested clustering three large-scale scRNA-seq datasets with 90K, 400K and 1.3M cell profiles generated with Microwell-seq or 10xGenomics and observed that cosNMF provides much better scalability while achieving competitive clustering performance compared with several commonly used the-state-of-the-arts methods for clustering scRNA-seq data. In particular, cosNMF is able to cluster 400K cells into 98 clusters in 1.67 hour and 1.3M cells into 10 clusters in less than 20 minutes on a regular server with a requirement of around 40G memory.

## KEYWORDS

single cell RNA-seq, sparse non-negative matrix factorization, clustering

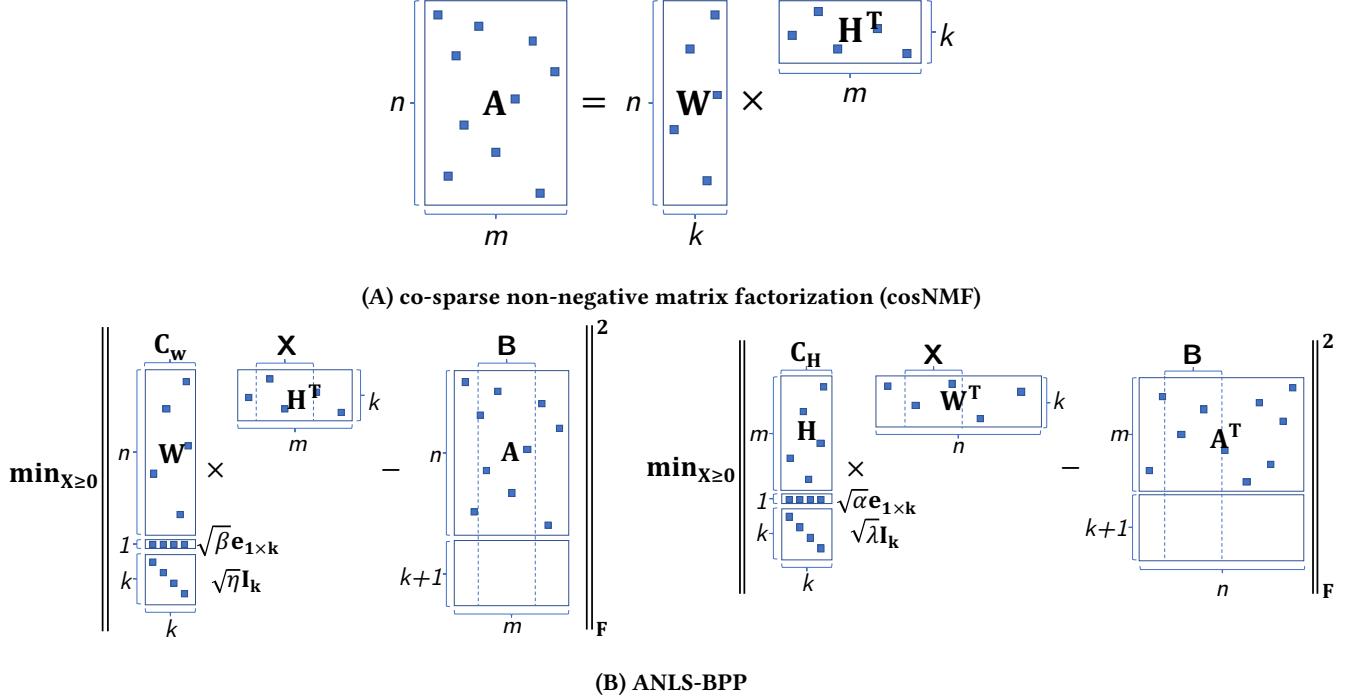
## 1 INTRODUCTION

Complex biological systems are made up of a heterogeneous mixture of distinct types of cells playing different functional roles. To investigate such a complex system, it is essential to reveal the cell identities for studying their biological functions. For instance, tumor tissues consist of different types of tumour cells created by imperfect DNA replication and these different cells lead to distinct morphological and phenotypic characteristics of the tumor playing different roles in cancer initiation, development and metastasis [11, 26]. Conventional bulk gene expression analysis only measures the average of the transcription levels of the genes in a bulk population of cells in a sample. Hence, it will not be able to reach the single cell resolution for studying a complex biological system.

More recently introduced single-cell RNA sequencing (scRNA-seq) are widely used for large-scale transcriptome profiling in individual cells [18, 36]. In a scRNA-seq analysis, single cells are isolated with a capture method and captured RNAs are reverse transcribed and amplified for sequencing [18]. Early capture methods such as flow-activated cell sorting (FACS) and Fluidigm C1 are only capable of capturing less than a thousand cells per run with a limited sampling resolution of the cells [18]. For example a typical early study [37] in 2009 only captured 8 mouse blastomere cells for profiling. The emerging high-throughput scRNA-seq protocols have exponentially increased the number of captured cells in each scRNA-seq experiment. For instances, droplet-based platforms such as 10xGenomics can capture 1.3 million mouse brain cells in one study [25].

Similar to the traditional clustering analysis of bulk gene expressions [2, 13], the scRNA-seq profiles of the individual cells are also clustered to reveal the cell subpopulations (types) and other downstream analysis including trajectory inference, pseudotime analysis, rare cell detection and cell gene markers discovery [28]. However, there are two unique challenges in clustering analysis of scRNA-seq data. First, several factors including cell-specific characteristics such as cell-cycle and cell size variation, and technical sources (e.g. library size difference, RNA capture inefficiency, amplification biases and sequencing depth) introduce substantially more noise and technical biases. These noise and biases lead to highly sparse data with many zero-coverage region and drop-out genes. Secondly, with the increasing size of scRNA-seq data, both in terms of the number of genes and cells can be assayed simultaneously resulting the scalability issues which renders most of the existing clustering methods inapplicable. In our recent survey analysis in [28], the comparative study clustering methods are only scalable to tens of thousands of scRNA-seq profiles unless subsampling is applied [17].

To address these two limitations, we formulate sparse non-negative matrix factorization (NMF) as co-sparse non-negative matrix factorization (cosNMF) to explore the sparseness of scRNA-seq data and the sparseness in both factor matrices to improve the scalability of clustering scRNA-seq data. In particular, we coupled non-negative factorization (NMF) on sparse matrix with two sparse constraints on both the resulting coefficient matrix and basis matrix to leverage the co-sparseness between the factor matrices such that cosNMF not only simultaneously clusters cells to identify the cell populations and the marker genes of each cluster, but also exploits the inherent sparsity to solve the scalability issues of clustering very large-scale scRNA-seq data. The cosNMF formulation can be efficiently solved with alternative non-negative least square with block



**Figure 1: Co-sparse non-negative matrix factorization (cosNMF) and alternative non-negative least square with block principal pivoting (ANLS-BPP) algorithm.** (A) Illustration of cosNMF for clustering single cell RNA-seq (scRNA-seq) data: sparse scRNA-seq matrix  $A$  is decomposed into two sparse factors, cell membership matrix  $W$  and gene contribution matrix  $H$ , where  $n$  denotes the number of cells,  $m$  denotes the number of genes and  $k$  denotes the number of clusters. (B) The building blocks of alternative non-negative least square with block principal pivoting (ANLS-BPP) algorithm. The the time complexity of each iteration of ANLS-BPP for solving cosNMF is also given at the bottom.

principal pivoting (ANLS-BPP) in the number of non-zero entries in the matrices in each iteration. In addition, scRNA-seq gene expressions are typically quantified as RPKM (Reads Per Kilobase Million), FPKM (Fragments Per Kilobase Million), TPM (Transcripts Per Kilobase Million) or the counts of unique molecular identifiers (UMI), which are all non-negative measures for natural application of cosNMF. In the experiment, we clustered three large-scale scRNA-seq datasets, two 10xGenomics datasets with 90K and 1.3M cells, respectively [1] and a Microwell-seq dataset with 1.3M cells [6], with cosNMF and other existing scRNA-seq data clustering methods. We observed that cosNMF provides much better scalability while achieving competitive clustering performance.

## 2 METHOD

In this section, we first introduce the formulation of co-sparse non-negative matrix factorization (cosNMF) and then describe the non-negativity constrained least square algorithm for solving the optimization problem. Finally, we provide the detail of the implementation of cosNMF.

### 2.1 cosNMF

Given a scRNA-seq data matrix  $A \in \mathbb{R}^{n \times m}$  where each row represents a cell, each column represents a gene and  $k$  denotes the number of clusters where  $k \ll \min\{m, n\}$ , NMF aims to find two

non-negative matrices  $W \in \mathbb{R}^{n \times k}$  and  $H \in \mathbb{R}^{m \times k}$  such that:

$$A \approx WH^T. \quad (1)$$

Then the solution  $W$  and  $H$  can be found by solving the non-convex optimization problem:

$$\min_{W,H} f_k(W,H) = \|A - WH^T\|_F^2 \text{ s.t. } W, H \geq 0, \quad (2)$$

where  $\|\cdot\|_F$  denotes Frobenius norm and  $W, H \geq 0$  means that each element of  $W$  and  $H$  is non-negative. Often  $W$  is called a basis matrix, and  $H$  is called a coefficient matrix. Note that NMF does not have a unique solution: if we have a solution  $(W, H)$ , then  $(WD, D^{-1}H)$  with any diagonal matrix  $D$  consisting of positive diagonal elements is also solution.

$L_1$ -norm regularization is commonly used to achieve the sparsity of solution [38]. As illustrated in Figure 1(A), we impose  $L_1$ -norm square constraints on both  $W$  and  $H$  so that we can leverage the co-sparsity on  $W$  and  $H$  to involve sparse computations for addressing the scalability issues on large-scale sparse scRNA-seq data, and also enhance the model interpretability—the sparsities on  $W$  and  $H$  indicate the cluster memberships of the cells and the gene marker contributions, respectively. The objective function of cosNMF is

given as follows:

$$\begin{aligned} \min_{W,H} & \frac{1}{2} \|A - WH^T\|_F^2 + \frac{\lambda}{2} \|W\|_F^2 + \frac{\eta}{2} \|H\|_F^2 \\ & + \frac{\alpha}{2} \sum_{i=1}^n \|W(i,:)\|_1^2 + \frac{\beta}{2} \sum_{j=1}^m \|H(j,:)\|_1^2, \\ \text{s.t.} & W, H \geq 0 \end{aligned} \quad (3)$$

where  $W(i,:)$  denotes  $i$ -th row vector of  $W$ , and  $H(j,:)$  denotes the  $j$ -th row vector of  $H$ . The hyperparameters  $\lambda > 0$  and  $\eta > 0$  control the scale of the elements of  $W$  and  $H$  respectively, and  $\alpha > 0$  and  $\beta > 0$  balance the accuracy of approximation and the sparseness of  $W$  and  $H$ . Larger values of  $\alpha$  and  $\beta$  imply stronger sparsity while smaller values lead to better approximation. Note that in the formulation in Eq. (3), we use  $L_1$ -norm square instead of the commonly used  $L_1$ -norm to achieve both the sparsity and a scalable solution as the previous work for dense microarray gene expression analysis [14].

The most intuitive way to solve this optimization problem is the multiplicative update (MU) rule [21], which is a matrix-wise update algorithm optimizing the objective function with respect to  $W$  and  $H$  alternatively. However, MU cannot maintain the sparse properties of  $W$  and  $H$  simultaneously in each iteration. To facilitate the sparse computation for efficient and scalable NMF, we applied alternative non-negative least square (ANLS) [15, 22] to solve cosNMF defined in Eq. (3). ANLS also optimizes the objective function with respect to one matrix while keeping the another matrix fixed, but it decomposes the optimization problem as several independent non-negativity constrained least squares (NNLS) with multiple right-hand sides.

In the ANLS formulation, the cosNMF optimization can be solved by iterating the subproblems as Eq. 4 and Eq. 5 shown below until convergence.

$$\min_H \left\| \begin{pmatrix} W \\ \sqrt{\beta}e_{1 \times k} \\ \sqrt{\eta}I_k \end{pmatrix} H^T - \begin{pmatrix} A \\ 0_{(k+1) \times m} \end{pmatrix} \right\|_F^2, \text{s.t. } H \geq 0, \quad (4)$$

$$\min_W \left\| \begin{pmatrix} H \\ \sqrt{\alpha}e_{1 \times k} \\ \sqrt{\lambda}I_k \end{pmatrix} W^T - \begin{pmatrix} A^T \\ 0_{(k+1) \times n} \end{pmatrix} \right\|_F^2, \text{s.t. } W \geq 0, \quad (5)$$

where  $e_{1 \times k} \in \mathbb{R}^{1 \times k}$  is a row vector with all elements as ones,  $I_k$  is an  $k \times k$  identity matrix, and  $0_{(k+1) \times m}$  and  $0_{(k+1) \times n}$  are zero matrices. And each subproblem in Eq. 4 and Eq. 5 can be further decomposed into a series of NNLS problems.

## 2.2 NNLS-BPP

Regarding the NNLS subproblem, many efficient algorithms have been developed, which enables ANLS algorithm to tackle the large sparse non-negative matrix. In this work, we employed the NNLS with block principal pivoting (NNLS-BPP) [16] to efficiently utilize the sparsity of  $W$  and  $H$  for solving the optimization problem.

For each NNLS problem in one iteration, it can be formulated as the following:

$$\min_{x \geq 0} \|Cx - b\|_F^2. \quad (6)$$

We first need to fix either  $C_W$  or  $C_H$ , which is an extension of  $W$  or  $H$  by corresponding  $L_1$ -norm and  $F$ -norm regularizers defined in the Eq. 4 and Eq. 5, and solve  $x$ , one column of  $H$  or  $W$  with respect to one right-hand side, a concatenation of the corresponding column in  $A$  and zeros, whose size exactly matches the  $x$ . The problem in Eq. 6 can be rewritten to satisfy the Karush–Kuhn–Tucker (KKT) conditions:

$$\begin{aligned} y &= C^T C x - C^T b, \\ \text{s.t.} & x \geq 0, y \geq 0, x_i y_i = 0, i = 1, \dots, k. \end{aligned} \quad (7)$$

In the block principal pivoting method, the index set  $\{1, \dots, k\}$  is simply divided into two groups  $F$  (inactive set) and  $G$  (active set), where  $x_G = 0$ ,  $y_F = 0$ ,  $F \cap G = \emptyset$  and  $F \cup G = \{1, \dots, k\}$ , the  $x_F$  and  $y_G$  are repeatedly calculated by the Eq. 8:

$$\begin{aligned} C_F^T C_F x_F &= C_F^T b, \\ y_G &= C_G^T C_F x_F - C_G^T b. \end{aligned} \quad (8)$$

and then the indices in  $F$  and  $G$  are updated by selecting and exchanging a set of indices which makes  $x_F < 0$  and  $y_G < 0$ . The updating repeats until  $x_F \geq 0$  and  $y_G \geq 0$ .

However, it is indeed computationally expensive to iterate all columns either in  $W$  or  $H$  for each iteration, Kim et al [16] proposed a grouping strategy on multiple right-hand sides sharing the same index sets  $x_F$  and  $y_G$ , which simultaneously solve the  $X$  with respect to multiple right-hand sides to further improves the efficiency. Figure 1(B) shows the basic structure of ANLS with block principal pivoting algorithm for cosNMF in each iteration. For each set of NNLS problems in one iteration, it can be formulated as the following:

$$\begin{aligned} \min_{X \geq 0} & \|CX - B\|_F^2. \\ C_F^T C_F X_F &= C_F^T B, \\ Y_G &= C_G^T C_F X_F - C_G^T B. \end{aligned} \quad (9)$$

here  $X$  is a part of  $H$  or  $W$  with respect to the multiple right-hand sides from the corresponding columns in  $B$ , where  $B$  consists of  $r$  columns from the matrix extending  $A$  with a zero matrix, and these  $r$  columns share the same index pattern. Moreover, the redundant computation in updating  $x_F$  and  $y_G$  can be avoided by calculating  $C^T C$  and  $C^T B$  ahead such that every iteration of the updating just extracts the sub-matrices from  $C^T C$  and  $C^T B$  directly. Since we enforce both  $W$  and  $H$  to be sparse,  $C^T C$  and  $C^T B$  are computationally cheap.

## 2.3 Implementation details

For the details of implementation, we modified the NNLS-BPP solver provided by Kim et al [16], and fit it into the cosNMF ANLS framework, the corresponding objective function is given as follows:

$$\begin{aligned} f^{(i)} = & \frac{1}{2} \|A - W^{(i)} H^{T(i)}\|_F^2 + \frac{\lambda}{2} \|W^{(i)}\|_F^2 + \frac{\eta}{2} \|H^{(i)}\|_F^2 \\ & + \frac{\alpha}{2} \sum_{i=1}^n \|W^{(i)}(i,:)\|_1^2 + \frac{\beta}{2} \sum_{j=1}^m \|H^{(i)}(j,:)\|_1^2, \end{aligned} \quad (10)$$

Note that for the initialization of  $W$  and  $H$ , commonly used PCA-based [44], kMeans-based [39] and SVD-based [3, 29] initialization are not easily applicable to large data. Hence, we adopted a simple strategy of initializing the  $W$  and  $H$  with random sparse non-negative elements.

To obtain a stopping criteria, we followed the same idea in projected NMF [22] which examines whether the change of relative projected gradient on both  $W$  and  $H$  of cosNMF (Eq. 10) has reached a predetermined threshold  $\epsilon$  at a checkpoint  $i$  (Eq. 11) as follows:

$$\begin{aligned} \Delta^{(i)} &= \sqrt{\text{proj}(\|\nabla f_{W^{(i)}}\|_F^2) + \text{proj}(\|\nabla f_{H^{(i)}}\|_F^2)}, \\ \text{proj}(\nabla_{jk} f_{W^{(i)}}) &= \begin{cases} \nabla_{jk} f_{W^{(i)}}, W_{jk}^{(i)} \geq 0 \\ 0, \text{otherwise} \end{cases} \\ \text{proj}(\nabla_{ij} f_{H^{(i)}}) &= \begin{cases} \nabla_{ij} f_{H^{(i)}}, H_{ij}^{(i)} \geq 0 \\ 0, \text{otherwise} \end{cases} \end{aligned} \quad (11)$$

$$\frac{\Delta}{\Delta^{(0)}} \leq \epsilon.$$

where  $\Delta^{(0)}$  denotes the projected gradient of using the initial  $W^{(0)}$  and  $H^{(0)}$ .

Finally, for the parameters  $\alpha$  and  $\beta$  controlling the sparsity of  $W$  and  $H$ , we applied grid search to identify the optimal ones which can obtain the best clustering performance. In the experiments, we set  $\epsilon = 10^{-5}$  and used grid search with exponentially growing sequences of  $\alpha$  and  $\beta$  as  $10^c$ ,  $c \in \{-4, \dots, 4\}$ . And  $\lambda$  and  $\eta$  are tested by three values  $0, 10^2, 10^4$ . cosNMF ANLS is implemented with matlab (R2018a).

### 3 EXPERIMENTS

In the experiments, we evaluated both the clustering scalability and performance of cosNMF on three large-scale scRNA-seq datasets, peripheral blood mononuclear cells (PBMC) dataset<sup>1</sup> and mouse brain cell (MBC) scRNA-seq dataset<sup>2</sup> downloaded from the 10x Genomics data repository and Mouse Cell Atlas (MCA) dataset<sup>3</sup> [10]. The PBMC dataset consists of 93,802 cells in 10 different cell types; the MCA dataset contains around 407,021 cells in 98 cell types; and the MBC dataset contains around 1,306,127 cells in 10 cell types. In the experiments, cosNMF are compared with 13 scRNA-seq methods briefly summarized in the Table 1 when the methods are applicable to the data of a certain size. The implementation detail of the baseline methods were described in detail in the review paper [28]. All the experiments on PBMC dataset were performed on a server with Intel Xeon E52687W v3 3.10 GHz 25M L3 Cache, and 256GB RAM, and all the experiments on MCA and MBC dataset were performed on a server with Intel Xeon E5-2540 2.10 GHz, 20M L3 Cache, and 192GB RAM. Note that in all the experiments, the actual usage of memory by cosNMF is up to around 40G, far from the maximum limit of the servers.

<sup>1</sup>PBMC dataset: [https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/pbmc\\_10k\\_protein\\_v3](https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/pbmc_10k_protein_v3)

<sup>2</sup>MBC dataset: [https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.3.0/1M\\_neurons](https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.3.0/1M_neurons)

<sup>3</sup>MCA dataset: [https://figshare.com/articles/MCA\\_DGE\\_Data/5435866](https://figshare.com/articles/MCA_DGE_Data/5435866)

**Table 1: Summaries of the clustering methods**

| Algorithm        | Method type                            |
|------------------|--|
| kMeans           | kMeans                                 |
| BackSPIN [42]    | hierarchical biclustering              |
| cellTree [41]    | LDA + hierarchical clustering          |
| CIDR [23]        | PCA + hierarchical clustering          |
| DendroSplit [43] | hierarchical clustering                |
| ICGS [27]        | hierarchical clustering                |
| Monocle 2 [30]   | t-SNE + density peaks clustering       |
| pcaReduce [40]   | PCA + kMeans + hierarchical clustering |
| SC3 [17]         | PCA + graph-based + kMeans             |
| SCRAT [5]        | SOM + hierarchical clustering          |
| Seurat [4, 31]   | PCA + graph-based                      |
| SNN-Cliq [8]     | graph-based                            |
| TSCAN [12]       | PCA + GMM                              |
| cosNMF           | non-negative matrix factorization      |

\*GMM indicates gaussian mixture model

### 3.1 Evaluation

To evaluate the performance and the scalability of the scRNA-seq clustering methods, adjusted rand index (ARI) and running time are measured as follows,

- (1) Rand Index (RI) is a measure of the level of agreement between the clustering results and the ground truths, and its generalized form ARI is a variation of RI corrected by the index expected by chance. Given clustering results  $X = \{X_1, \dots, X_r\}$  and ground truths  $Y = \{Y_1, \dots, Y_s\}$ , both  $X_i, 1 \leq i \leq r$  and  $Y_j, 1 \leq j \leq s$  are the clusters in  $X$  and  $Y$ , the ARI is defined as follows:

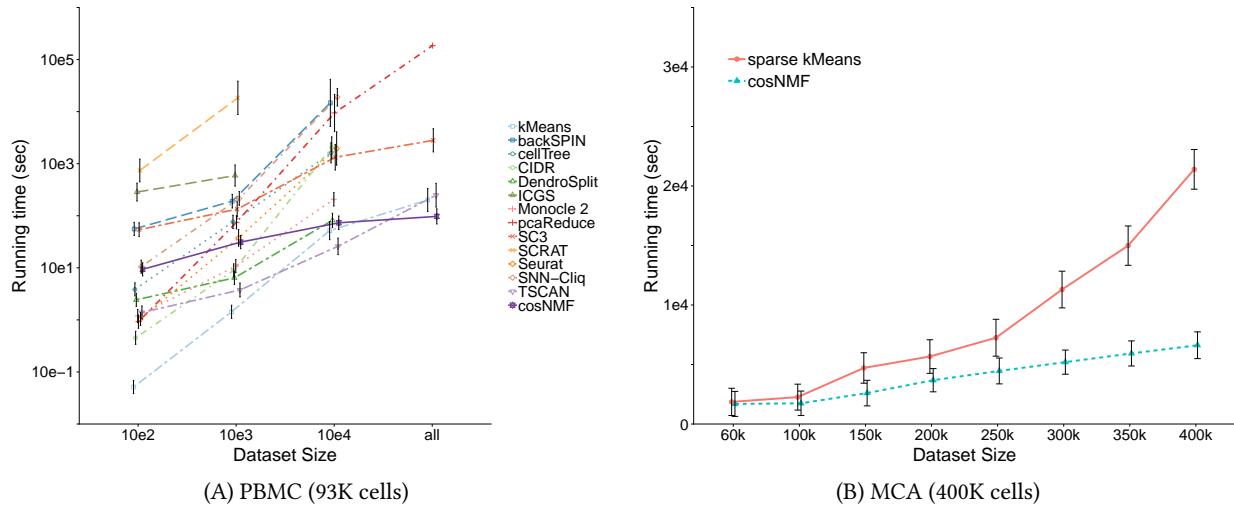
$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[ \sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{n}{2}}{\frac{1}{2} \left[ \sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right] - \left[ \sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{n}{2}}, \quad (12)$$

where  $n_{ij} = |X_i \cap Y_j|$  is the number of common samples in  $X_i$  and  $Y_j$ ,  $a_i = \sum j n_{ij}$ , and  $b_i = \sum i n_{ij}$ . The larger value of ARI indicates that the clustering results are more consistent with the ground truth while the smaller value of ARI indicates that the clustering results are more approximate to the random results. Note that ARI also could be negative if the index  $\sum_{ij} \binom{n_{ij}}{2}$  is less than the expected index  $\left[ \sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{n}{2}$ .

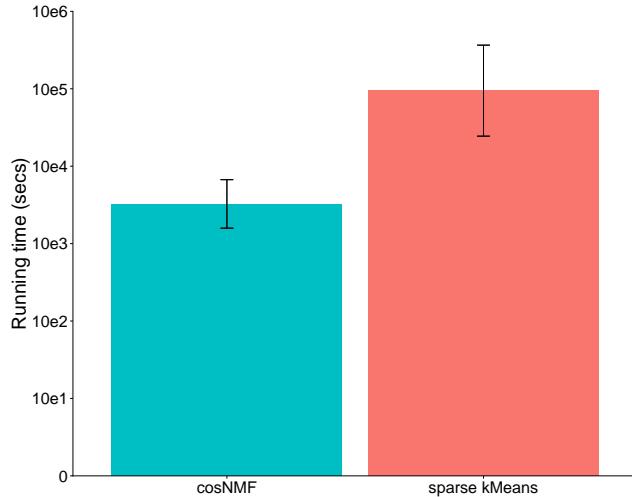
- (2) Running time measures the efficiency of the algorithms. Other than the total running time, we also measured the running time at every iteration up to the convergence as we defined in Eq. (11) to fully understand the convergence trajectory of cosNMF and the compared methods.

### 3.2 Scalability on large scRNA-seq datasets

To measure the scalability of cosNMF, we measured the running time for clustering the PBMC, MCA and MBC datasets. First, we measured the running time on the PBMC scRNA-seq dataset with 19630 genes, 10 cell types, and sampled data in 3 different dataset size, 100, 1,000, 10,000 and all cells (93,802 cells) in clustering by the 13 other clustering methods and cosNMF. For a fair comparison, we



**Figure 2: Comparison of the scalability of cosNMF and other scRNA-seq clustering methods. (A)** The plot shows the running time of cosNMF and 13 other clustering methods with respect to the three different sizes of sampling of the PBMC data and the complete PBMC dataset. **(B)** The plot shows the running time of cosNMF and sparse kMeans with respect to 7 different sizes of sampling of the the MCA data the complete MCA dataset. Note, in this experiment, only sparse implementation of kMeans is scalable for comparison with cosNMF.



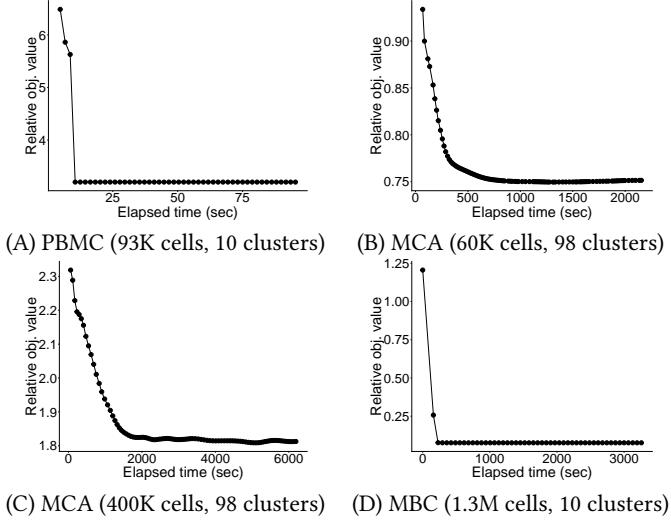
**Figure 3: Comparison of running time of clustering MBC dataset (1.3M cells in 10 clusters).**

applied the same stopping criteria to the similar methods such as the maximal number of iterations, and threshold of relative gradient or objective function, and fine-tuned each method with different parameters using the same strategy explained in the paper [28] to select the optimal parameters which reported the best clustering performance. We repeated the same experiment 10 times to obtain the average and standard deviation of the running time.

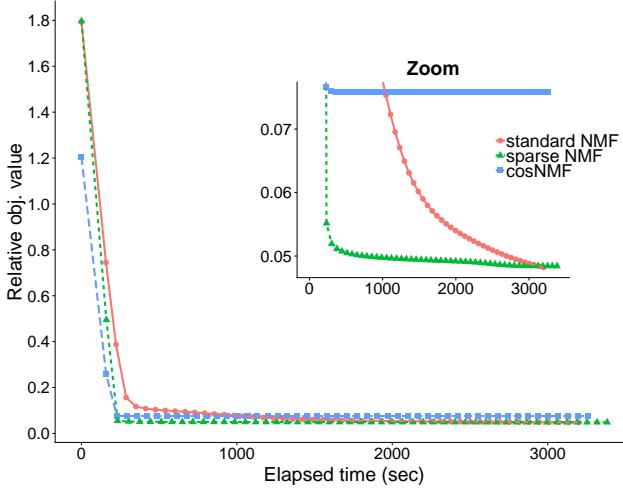
Figure 2(A) shows the comparison of running time on the PBMC data. We clearly observed that most of the methods can deal with

relatively small dataset ( $\leq 10,000$ ) efficiently. ICGS and SCART performed poorly even when there are only 1,000 cells in the dataset: ICGS took more than 20 minutes and SCRAT took more than 5 hours. Note that SCART as a pipeline also involves extra computational cost for downstream analysis, such as inference of lineage relationships and gene enrichment analysis, but the overall running time is still dominated by the clustering analysis. As the dataset size increases, the scalability issue is gradually unfolded: Monocle 2, cellTree, SC3 and Seurat are not scalable on all cells in PBMC dataset since the space they required exceeded the maximum limit of the main memory. pcaReduce can be applied to clustering all the 93K cells in PBMC dataset; however, the associated running time is at least 2 days to complete. kMeans and TSCAN shows better scalability. Compared with the other methods, cosNMF was among the methods with the lowest running time on the small datasets, but it only took around 100 seconds to cluster the entire dataset of 93K, faster than kMeans and TSCAN. Hence, cosNMF achieved the best scalability on the PBMC dataset.

Next, we evaluated the scalability of cosNMF on the MCA data with 25,133 genes, 98 cell types and 8 different size from 60,000 to 407,021 cells. Since the other 13 methods including standard kMeans are not scalable on this large dataset to cluster the cells into 98 clusters, we only compared cosNMF with our implementation of sparse kMeans, which is basically the same as standard kMeans except the data matrix is stored as a sparse matrix for computation. The clustering running time is shown in Figure 2(B). Overall, cosNMF is faster than sparse kMeans which is the only scalable method for this comparison. cosNMF spent approximately 30 minutes clustering the MCA data with 60,000 cells and 1.67 hours on the entire MCA dataset as compared to 6 hours by sparse kMeans. Finally, we measured the running time of clustering the largest scRNA-seq dataset MBC. The experimental results are shown in Figure 3 where



**Figure 4: Relative objective function value of cosNMF against running time measured at each iteration on PBMC, MCA 60k, MCA and MBC datasets.**



**Figure 5: Relative objective function value of standard NMF, sparse NMF and cosNMF are plotted against their corresponding running time at each iteration on MBC dataset.**

cosNMF (1 hour to cluster the 1.3M cells into 10 clusters) was much faster than sparse kMeans (more than 10 times slower). It appears in the experiments, larger the scRNA-seq dataset, the better is the scalability of cosNMF as compared with other methods including the highly scalable sparse kMeans.

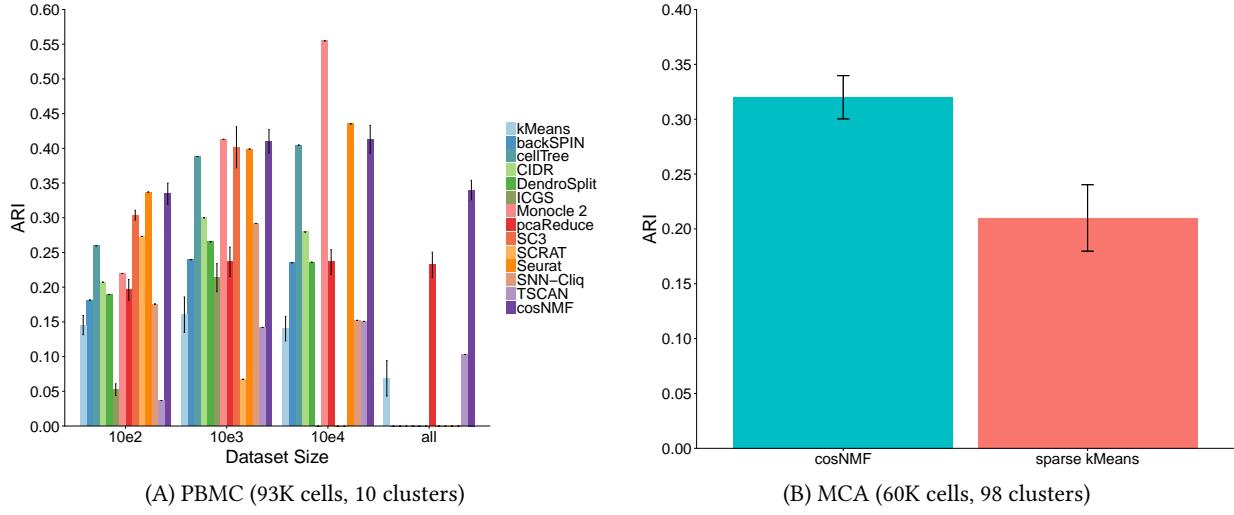
In terms of validation, the scalability of cosNMF is also tested based on its convergence while clustering the three datasets. Figure 4 shows the average relative objective function value defined in Eq. 3 of  $i$ -th iteration against cumulative average running time of first  $i$ -th iteration from 10 times of repeated experiments. We found that cosNMF converges eventually after a relatively small

number of iterations, which further explains why cosNMF is faster than the other methods. In addition, to verify whether the sparsity on both  $W$  and  $H$  plays a pivotal role in fast convergence, we compared the convergence of cosNMF with standard NMF and sparse NMF using NNLS-BPP on this largest dataset, as shown in Figure 5, shows sparse NMF only has  $L_1$ -norm square constraint on  $H$ . Although the difference is not significant between three NMF-based methods, but it is highlighted when zooming into the specific range of relative objective function value: in contrast to the relative objective function values of sparse NMF and standard NMF consistently tending to decrease, the relative objective function value of cosNMF becomes quite stable after first 1,000 seconds of running. Hence, cosNMF clearly exploits the co-sparsity in both  $W$  and  $H$  to achieve better convergence than the other two NMF-based methods. Note that since cosNMF and sparse NMF have additional regularization terms in addition to the reconstruction error, the relative objective function value of cosNMF, sparse NMF and NMF might be different at the convergence but the difference does not imply any relation to the clustering accuracy, meanwhile, sparse NMF and NMF have much larger memory consumption than the cosNMF.

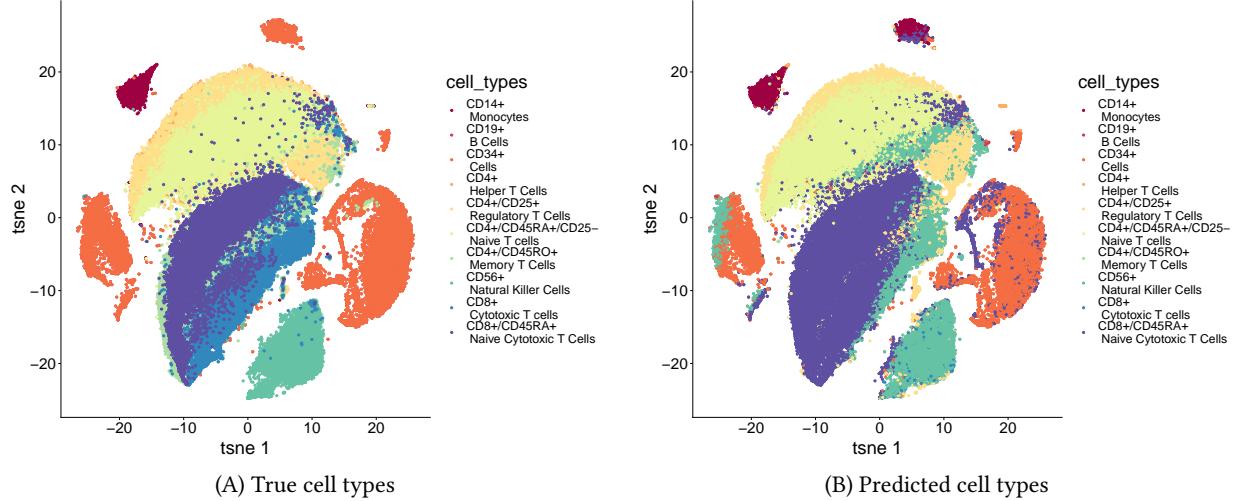
### 3.3 Clustering performance on large scRNA-seq datasets

We quantified the clustering accuracy by ARI on PBMC data. Figure 6 (A) shows the comparison of the clustering performance of all the tested methods. The results show that Seurat, Monocle and cellTree achieved similarly good performance on relatively small datasets while the three methods are not applicable to the entire PBMC dataset. Despite SC3 could be used on more than 5,000 cells, it clusters 5,000 cells first and classifies the remaining cells in a second stage of supervised learning. So, we did not add SC3 to the comparisons when the number of cells exceeds 5,000 cells. Again, kMeans, pcaReduce and TSCAN indeed obtained clustering results on the entire PBMC dataset but the ARIs are lower than the cosNMF. While cosNMF performed the best on the entire dataset, it also achieved good ARIs on relatively small datasets close to the best ARIs from Seurat. Note that Monocle 2 performed the best on the dataset of size 10,000, and performed similarly as cosNMF and Seurat on the other datasets of smaller sizes. Further, we visualized both the clustering by cosNMF and the true cell types to 2 dimensional space by t-SNE in Figure 7.

We also performed a similar clustering experiment on the MCA data. Since most of the methods are not scalable to the large dataset with more than 60,000 cells and TSCAN also has difficult in yielding clustering results of the desired large number of 98 clusters, we again only compared the clustering performance between sparse kMeans and cosNMF. Also, since only 60,000 cells in the MCA have the associated cell type information, we measured the average ARIs and standard deviations on the dataset with 60,000 annotated cells by repeating the clustering 10 times. Figure 6(B) shows that cosNMF achieves a better ARI=0.32 than sparse kMeans' ARI=0.21. Again, we mapped the clustering by cosNMF and the true cell types to the projected 2 dimensional space by t-SNE in the Figure 8. Since it is difficult to distinguish 98 different cell types in the MCA data by



**Figure 6: Comparison of clustering performance between cosNMF and 13 other scRNA-seq clustering methods.** (A) The plot shows the ARI of cosNMF and the 13 clustering methods on 3 different sampling size of PBMC dataset and the complete PBMC dataset. (B) The plot shows the ARI of cosNMF and sparse kMeans on 7 different sampling sizes of MCA data and the complete MCA dataset.



**Figure 7: t-SNE analysis of PBMC dataset with the true and predicted cell types. The 10 tissue types are annotated in both t-SNE maps.**

colors, we labelled the corresponding clusters by numbers in both plots.

### 3.4 Biological function analysis

As the matrix  $H$  can be interpreted as gene markers of clusters, we also did the Gene Ontology (GO) enrichment analysis on each cluster by using the genes with the large coefficients in the corresponding row of the  $H$  to further validate whether the discovered clusters are associated with specific cell types. Since genes may be active markers in different cell types, we defined z-score for each gene to identify the relative contributions over clusters, the

higher gene z-score indicates the larger contribution to the cluster. In GO enrichment analysis, top 500 genes were chosen by ranking the gene z-score in each cluster, and biological process GO terms whose Bonferroni adjusted P-value less than 0.05 were reported. We selected top 3 clusters ranked by the number of enriched GO terms, and presented top the 6 significant biological processes in each clusters on PBMC data (Table 2) and MCA data (Table 3). For the PBMC dataset, the GO enrichment results of cluster 1 shows the cluster is highly correlated with the regulation of T cells, and the true cell types of majority of the cells in cluster 1 are likely Regulatory T Cells, which act to suppress immune response [33],

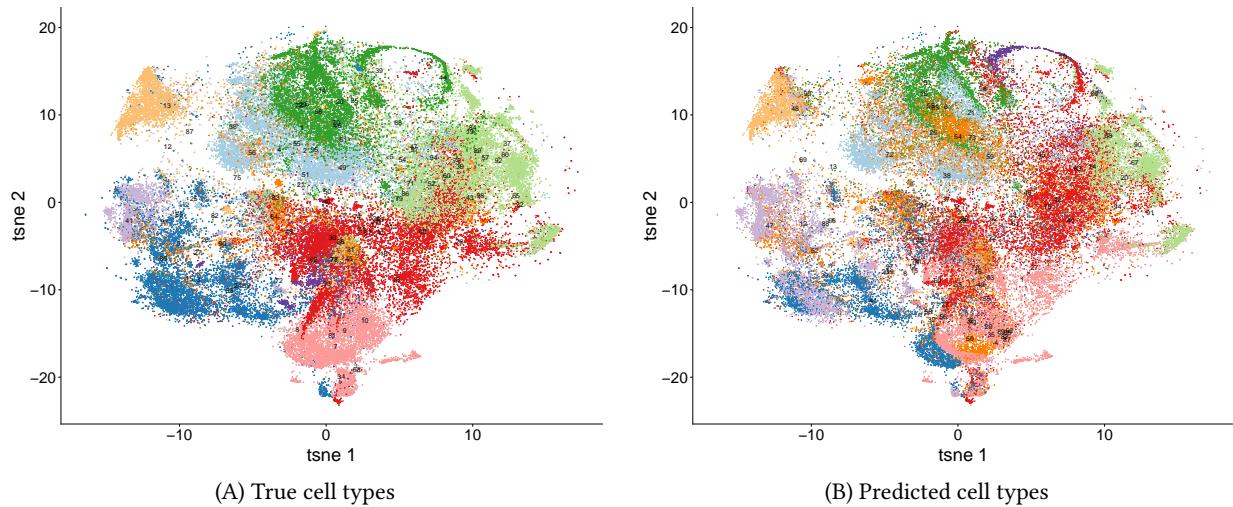


Figure 8: t-SNE analysis of MCA 60k dataset with the true and predicted cell types. The 98 tissue types are labeled with the cluster number in both t-SNE maps.

**Table 2:** The GO enrichment of top 500 genes in selected clusters on PBMC data

| Cluster               | Biological process   | Number of genes | Adjusted P-value |
|-----------------------|--|-----------------|------------------|
| Cluster 1 (382 genes) | Regulation of T cells activation (GO:0050863)  | 26 genes        | 1.03e-5          |
|                       | T cell activation (GO:0042110)   | 31 genes        | 3.00e-5          |
|                       | Regulation of lymphocyte activation (GO:0051249)   | 29 genes        | 4.50e-5          |
|                       | Regulation of cell-cell adhesion (GO:0022407)  | 28 genes        | 9.00e-5          |
|                       | Leukocyte cell-cell adhesion (GO:0007159)  | 25 genes        | 1.72e-4          |
|                       | Response to chemokine (GO:1990868)   | 11 genes        | 1.99e-4          |
| Cluster 2 (401 genes) | Neutrophil activation (GO:0042119)   | 49 genes        | 9.06e-15         |
|                       | Neutrophil degranulation (GO:0043312)  | 48 genes        | 1.51e-14         |
|                       | Neutrophil activation involved in immune response (GO:0002283)                                 | 48 genes        | 1.94e-14         |
|                       | Neutrophil mediated immunity (GO:0002446)  | 48 genes        | 4.79e-14         |
|                       | Positive regulation of response to external stimulus (GO:0032103)                              | 27 genes        | 1.06e-5          |
|                       | Positive regulation of cytokine production (GO:0001819)  | 11 genes        | 1.99e-4          |
| Cluster 3 (380 genes) | Antigen processing and presentation of peptide antigen via MHC class II (GO:0048002)           | 18 genes        | 1.24e-8          |
|                       | antigen processing and presentation of exogenous peptide antigen via MHC class II (GO:0019886) | 17 genes        | 1.48e-8          |
|                       | Interferon-gamma-mediated signaling pathway (GO:0002283)                                       | 13 genes        | 2.40e-4          |
|                       | Response to interferon-gamma (GO:0034341)  | 17 genes        | 6.64e-4          |
|                       | Regulation of leukocyte activation (GO:0002694)  | 29 genes        | 2.98e-3          |
|                       | Cellular response to interferon-gamma (GO:0071346)   | 14 genes        | 9.02e-3          |

and the behavior of regulatory T cells in immune response will be critically influenced by chemokine [9]. It is likely that most of the cells in cluster 2 are Monocytes since we found monocytes and neutrophils share a complex relationship to generate an enhanced immune response by regulating each other [19], and innate immune cell usually generate cytokine to mediate the activities of monocytes [35]. The cluster 3 is linked to Helper T cells which can

recognize the antigen-presenting cells expressing MHC class II and then aid other immune cells to generate immune response [7, 24], and Interferon-gamma has different effects on the differentiation and proliferation of distinct helper T cells [32, 34].

Similarly, for the MCA dataset, we observed that the 3 clusters are strongly associated with Muscle cells, T cells, and Neuron cells respectively. In cluster 1, aside from the common biological processes

**Table 3: The GO enrichment of top 500 genes in selected clusters on MCA data**

| Cluster               | Biological process                                       | Number of genes | Adjusted P-value |
|-----------------------|--|-----------------|------------------|
| Cluster 1 (477 genes) | Muscle cell development (GO:0055001)                     | 25 genes        | 4.94e-9          |
|                       | Muscle contraction (GO:0006936)                          | 29 genes        | 9.07e-9          |
|                       | Muscle tissue development (GO:0060537)                   | 36 genes        | 6.89e-8          |
|                       | Muscle cell differentiation (GO:0042692)                 | 32 genes        | 2.02e-7          |
|                       | Calcium ion transport (GO:0006816)                       | 33 genes        | 3.28e-7          |
|                       | Cellular calcium ion homeostasis (GO:0006874)            | 34 genes        | 1.391e-6         |
| Cluster 2 (466 genes) | T cell activation (GO:0042110)                           | 50 genes        | 1.04e-17         |
|                       | Regulation of adaptive immune response (GO:0002819)      | 30 genes        | 7.22e-14         |
|                       | Regulation of cell killing (GO:0031341)                  | 23 genes        | 6.26e-12         |
|                       | Regulation of lymphocyte mediated immunity (GO:0002706)  | 27 genes        | 1.24e-11         |
|                       | Regulation of leukocyte mediated immunity(GO:0002703)    | 30 genes        | 1.84e-11         |
|                       | Positive regulation of cell killing (GO:0031343)         | 20 genes        | 3.11e-11         |
| Cluster 3 (482 genes) | Regulation of synapse assembly (GO:0051963)              | 23 genes        | 1.69e-12         |
|                       | Regulation of synapse structure or activity (GO:0050803) | 33 genes        | 2.02e-12         |
|                       | Regulation of neurotransmitter levels (GO:0002283)       | 38 genes        | 2.58e-12         |
|                       | Regulation of synapse organization (GO:0050807)          | 32 genes        | 4.74e-12         |
|                       | Neurotransmitter secretion (GO:0007269)                  | 25 genes        | 5.70e-10         |
|                       | Signal release from synapse (GO:0099643)                 | 25 genes        | 6.45e-10         |

related to muscle cell, such as muscle cell development, differentiation and contractions, two additional biological processes calcium ion transport and cellular calcium ion homeostasis were detected. It has been previously reported that the two biological processes can control the concentration of calcium ion to regulate muscle cell contraction [20]. In cluster 2 of T cells, all the corresponding biological processes are related to various general regulation of immune response. Most neuron cells are grouped in the cluster 3, in which all processes are relevant to the regulation of synapse and neurotransmitter, and their primary functions in the neuron cells.

## 4 CONCLUSIONS AND DISCUSSIONS

The goal of this study is to explore the sparseness in the scRNA-seq data and the  $L_1$ -norm sparseness computation for improving the scalability of clustering. In our analysis we observed that by introducing sparseness into the factor matrices, cosNMF is more scalable than k-means or NMF implemented for sparse input data while cosNMF also achieved much better clustering performance by performing gene selection during the clustering.

Due to the low sequence depth in the scRNA-seq data, the expression profiles are typically sparse detecting only a few thousands of expressed genes. Especially, in the droplet-based platforms, unique molecular identifiers rather than the mapped reads are sequenced to detect and quantify unique mRNA transcripts. Thus, we expect that high sparseness will continue to be an inherent characteristics of scRNA-seq data in the near future and exploring the sparseness of the data will be a major approach to improve the scalability.

In the literature, typically, embedding methods such as Principal Component Analysis (PCA), Local Linear Embedding (LLE), Multi-Dimensional Scaling (MDS), t-distributed Stochastic Neighbor Embedding (t-SNE), Canonical Correlation Analysis (CCA), Latent Dirichlet Allocation (LDA) and etc. are applied to reduce

the scalability issue by reducing the dimensionality in clustering. However, even if some the embedding methods can also be implemented for the sparse data, they will still not be able to scale to millions of cells. Moreover, cosNMF directly applies gene selection on the original gene features rather than embedding with latent features which are difficult to interprets for functional analysis.

The results collectively support that cosNMF is useful in analyzing large-scale scRNA-seq datasets. In the future, we plan to implement parallel cosNMF using the most popular programming platforms including R and python for better applicability in a broader research community.

## REFERENCES

- [1] 10x Genomics. 2019. 10X Genomics single cell gene expression datasets. <https://support.10xgenomics.com/single-cell-gene-expression/datasets>.
- [2] Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. 1999. Clustering gene expression patterns. *Journal of computational biology* 6, 3-4 (1999), 281–297.
- [3] Christos Boutsidis and Efstratios Gallopoulos. 2008. SVD based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition* 41, 4 (2008), 1350–1362.
- [4] Andrew Butler, Paul Hoffman, Peter Smibert, et al. 2018. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology* 36, 5 (2018), 411.
- [5] J Gray Camp, Keisuke Sekine, Tobias Gerber, et al. 2017. Multilineage communication regulates human liver bud development from pluripotency. *Nature* 546, 7659 (2017), 533.
- [6] Junyue Cao, Darren A Cusanovich, Vijay Ramani, Delasa Aghamirzaie, Hannah A Pliner, Andrew J Hill, Riza M Daza, Jose L McFaline-Figueroa, Jonathan S Packer, Lena Christiansen, et al. 2018. Joint profiling of chromatin accessibility and gene expression in thousands of single cells. *Science* 361, 6409 (2018), 1380–1385.
- [7] David Kenneth Cole. 2013. Re-directing CD4+ T cell responses with the flanking residues of MHC class II-bound peptides: the core is not enough. *Frontiers in immunology* 4 (2013), 172.
- [8] Yanglan Gan, Ning Li, Guobing Zou, et al. 2018. Identification of cancer subtypes from single-cell RNA-seq data using a consensus clustering method. *BMC medical genomics* 11, 6 (2018), 117.
- [9] Jason W Griffith, Caroline L Sokol, and Andrew D Luster. 2014. Chemokines and chemokine receptors: positioning cells for host defense and immunity. *Annual review of immunology* 32 (2014), 659–702.

- [10] Xiaoping Han, Renying Wang, Yinchong Zhou, Lijiang Fei, Huiyu Sun, Shujing Lai, Assieh Saadatpour, Ziming Zhou, Haide Chen, Fang Ye, et al. 2018. Mapping the mouse cell atlas by microwell-seq. *Cell* 172, 5 (2018), 1091–1107.
- [11] Gloria H Heppner. 1984. Tumor heterogeneity. *Cancer research* 44, 6 (1984), 2259–2265.
- [12] Zhicheng Ji and Hongkai Ji. 2016. TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic acids research* 44, 13 (2016), e117–e117.
- [13] Daxin Jiang, Chun Tang, and Aidong Zhang. 2004. Cluster analysis for gene expression data: a survey. *IEEE Transactions on knowledge and data engineering* 16, 11 (2004), 1370–1386.
- [14] Hyunsoo Kim and Haesun Park. 2007. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics* 23, 12 (2007), 1495–1502.
- [15] Hyunsoo Kim and Haesun Park. 2008. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM journal on matrix analysis and applications* 30, 2 (2008), 713–730.
- [16] Jingu Kim and Haesun Park. 2011. Fast nonnegative matrix factorization: An active-set-like method and comparisons. *SIAM Journal on Scientific Computing* 33, 6 (2011), 3261–3281.
- [17] Vladimir Yu Kiselev, Kristina Kirschner, Michael T Schaub, et al. 2017. SC3: consensus clustering of single-cell RNA-seq data. *Nature methods* 14, 5 (2017), 483.
- [18] Aleksandra A Kolodziejczyk, Jong Kyoung Kim, Valentine Svensson, et al. 2015. The technology and biology of single-cell RNA sequencing. *Molecular cell* 58, 4 (2015), 610–620.
- [19] Kathryn Prame Kumar, Alyce J Nicholls, and Connie HY Wong. 2018. Partners in crime: neutrophils and monocytes/macrophages in inflammation and disease. *Cell and tissue research* 371, 3 (2018), 551–565.
- [20] Ivana Y Kuo and Barbara E Ehrlich. 2015. Signaling in muscle contraction. *Cold Spring Harbor perspectives in biology* 7, 2 (2015), a006023.
- [21] Daniel D Lee and H Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*. 556–562.
- [22] Chih-Jen Lin. 2007. Projected gradient methods for nonnegative matrix factorization. *Neural computation* 19, 10 (2007), 2756–2779.
- [23] Pejje Lin, Michael Troup, and Joshua WK Ho. 2017. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome biology* 18, 1 (2017), 59.
- [24] Rishi Vishal Luckheram, Rui Zhou, Asha Devi Verma, and Bing Xia. 2012. CD4+ T cells: differentiation and functions. *Clinical and developmental immunology* 2012 (2012).
- [25] Evan Z Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R Bialas, Nolan Kamitaki, Emily M Martersteck, et al. 2015. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* 161, 5 (2015), 1202–1214.
- [26] Andriy Marusyk and Kornelia Polyak. 2010. Tumor heterogeneity: causes and consequences. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer* 1805, 1 (2010), 105–117.
- [27] Andre Olsson, Meenakshi Venkatasubramanian, Viren K Chaudhri, et al. 2016. Single-cell analysis of mixed-lineage states leading to a binary cell fate choice. *Nature* 537, 7622 (2016), 698.
- [28] Raphael Petegrossi, Zhiliu Li, and Rui Kuang. 2019. Machine Learning and Statistical Methods for Clustering Single-cell RNA-sequencing Data. *Briefings in Bioinformatics* (2019). <http://compbio.cs.umn.edu/wp-content/uploads/2019/05/SingleCellReview.pdf>
- [29] Hanli Qiao. 2015. New SVD based initialization strategy for non-negative matrix factorization. *Pattern Recognition Letters* 63 (2015), 71–77.
- [30] Xiaojie Qiu, Qi Mao, Ying Tang, et al. 2017. Reversed graph embedding resolves complex single-cell trajectories. *Nature methods* 14, 10 (2017), 979.
- [31] Rahul Satija, Jeffrey A Farrell, David Gennert, et al. 2015. Spatial reconstruction of single-cell gene expression data. *Nature biotechnology* 33, 5 (2015), 495.
- [32] Jamie R Schoenborn and Christopher B Wilson. 2007. Regulation of interferon- $\gamma$  during innate and adaptive immune responses. *Advances in immunology* 96 (2007), 41–101.
- [33] Amir Sharabi, Maria G Tsokos, Ying Ding, Thomas R Malek, David Katzmann, and George C Tsokos. 2018. Regulatory T cells in the treatment of disease. *Nature Reviews Drug Discovery* (2018).
- [34] Dorothy K Sojka and Deborah J Fowell. 2011. Regulatory T cells inhibit acute IFN- $\gamma$  synthesis without blocking T-helper cell type 1 (Th1) differentiation via a compartmentalized requirement for IL-10. *Proceedings of the National Academy of Sciences* 108, 45 (2011), 18336–18341.
- [35] Amanda C Stanley and Paige Lacy. 2010. Pathways for cytokine secretion. *Physiology* 25, 4 (2010), 218–229.
- [36] Oliver Stegle, Sarah A Teichmann, and John C Marioni. 2015. Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics* 16, 3 (2015), 133.
- [37] Fuchou Tang, Catalin Barbacioru, Yangzhou Wang, Ellen Nordman, Clarence Lee, Nanlan Xu, Xiaohui Wang, John Bodeau, Brian B Tuch, Asim Siddiqui, et al. 2009. mRNA-Seq whole-transcriptome analysis of a single cell. *Nature methods* 6, 5 (2009), 377.
- [38] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58, 1 (1996), 267–288.
- [39] Yun Xue, Chong Sze Tong, Ying Chen, and Wen-Sheng Chen. 2008. Clustering-based initialization for non-negative matrix factorization. *Appl. Math. Comput.* 205, 2 (2008), 525–536.
- [40] Christopher Yau et al. 2016. pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC bioinformatics* 17, 1 (2016), 140.
- [41] Sohiya Yotsukura, Seitaro Nomura, Hiroyuki Aburatani, et al. 2016. CellTree: an R/biocductor package to infer the hierarchical structure of cell populations from single-cell RNA-seq data. *BMC bioinformatics* 17, 1 (2016), 363.
- [42] Amit Zeisel, Ana B Muñoz-Manchado, Simone Codeluppi, et al. 2015. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* 347, 6226 (2015), 1138–1142.
- [43] Jesse M Zhang, Jue Fan, H Christina Fan, et al. 2018. An interpretable framework for clustering single-cell RNA-Seq datasets. *BMC bioinformatics* 19, 1 (2018), 93.
- [44] Lihong Zhao, Guibin Zhuang, and Xinhe Xu. 2008. Facial expression recognition based on PCA and NMF. In *2008 7th World Congress on Intelligent Control and Automation*. IEEE, 6826–6829.