Supplementary Materials to BANMF-S

Contents

1 Interpretation of Regularizations.

We followed geometric interpretation of the graph regularization $Tr(HLH^T)$ given in [2]. Recall that $S \in \mathbb{R}^{n\times n}$ refers to the adjacency matrix of the gene similarity network and $L \in \mathbb{R}^{n\times n}$ is the graph Laplacian matrix defined by $L = \mathrm{diag}(S \cdot \mathbf{1}) - S$. We then illustrate that the regularization term $Tr(HLH^T)$ enforces the coherence of gene similarity structure.

Denote $\boldsymbol{h}_i \in \mathbb{R}^{p\times 1}$ as the $i-$th column vector of gene matrix H, which represents Gene i's expression in the latent space. Hence, $\|\boldsymbol{h}_i - \boldsymbol{h}_j\|^2$ characterizes the dissimilarity of Gene i and Gene j in the latent space. Given the *gene higher-order similarity* S_{ij} for Gene i and Gene j, we obtained a weighted-sum of overall dissimilarity across gene pairs by $\mathcal{R}_1$ in Eq. (1),

$$\begin{aligned}
\mathcal{R}_1 &= \frac{1}{2}\sum_{i,j} \|\boldsymbol{h}_i - \boldsymbol{h}_j\|^2 \cdot S_{ij} \\
&= \sum_i \boldsymbol{h}_i^T \boldsymbol{h}_i \cdot \left(\sum_j S_{ij}\right) - \sum_{i,j} \boldsymbol{h}_i^T \boldsymbol{h}_j \cdot S_{ij} \\
&= Tr(H \cdot \mathrm{diag}(S \cdot \mathbf{1}) \cdot H^T) - Tr(HSH^T) \\
&= Tr(H \cdot (\mathrm{diag}(S \cdot \mathbf{1}) - S) \cdot H^T) \\
&= Tr(HLH^T),
\end{aligned} \tag{1}$$

which demonstrates that minimizing $Tr(HLH^T)$ is equivalent to minimizing the overall dissimilarity of gene pairs in the latent space.

The cell regularization term $\|A - WW^T\|_F^2$ is highly related to the graph regularization term, it can be proved that $\min\limits_{W \geq 0, W^T W = I} \|A - WW^T\|_F^2$ can be transferred to the graph regularization form $\min \mathrm{tr}(W^T\tilde{L}W)$ as follows:

$$
\begin{aligned}
& \min_{W\geq 0, W^TW=I} \|A-WW^T\|_F^2 \\
= & \min_{W\geq 0, W^TW=I} \mathrm{tr}[(A-WW^T)(A-WW^T)] \\
= & \min_{W\geq 0, W^TW=I} \mathrm{tr}(I) - 2\mathrm{tr}(W^TAW) + \mathrm{tr}(A^TA) \\
= & \min_{W\geq 0, W^TW=I} 2(\mathrm{tr}(I) - \mathrm{tr}(W^TAW)) + const \\
= & \min_{W\geq 0, W^TW=I} 2(\mathrm{tr}(W^TW) - \mathrm{tr}(W^TAW)) + const \\
= & \min_{W\geq 0, W^TW=I} 2\mathrm{tr}(W^T(I-A)W) + const \\
= & \min_{W\geq 0, W^TW=I} 2\mathrm{tr}(W^T(I-D^{-1/2}\tilde{A}D^{-1/2})W) + const \\
= & \min_{W\geq 0, W^TW=I} 2\mathrm{tr}(W^T\tilde{L}W) + const
\end{aligned}
\tag{2}
$$

In Eq. (2), "const" means a constant number, and in penultimate line, we replace A by

$$A = D^{-1/2}\tilde{A}D^{-1/2}, D = diag(\tilde{A}).$$

Moreover, Ding et.al [5] proved that the orthogonality of W can still be retained if the constraint $W^TW = I$ is removed.

In other words, if the cell similarity matrix is normalized, and Euclidean distance is utilized to measure similarities in the latent space, then $\min\limits_{W\geq 0} \|A-WW^T\|_F^2$ is equivalent to the graph regularization term. However, in this work, we chose MST-based cell similarity measurement, we directly used $\min\limits_{W\geq 0} \|A-WW^T\|_F^2$ to preserve the local structure of cell similarity network.

2 Blocklization Procedures

In the distributed SGD, original data matrix was firstly divided into blocks. Let K be the prescribed number of splits, we first divided X_0 into K^2 blocks of various sizes, and then divided A, M, W, H and L into blocks accordingly. Specifically, let $m_d = \lfloor \frac{m}{K} \rfloor, m_r = m(\text{mod } K), n_d = \lfloor \frac{n}{K} \rfloor, n_r = n(\text{mod } K)$, so that $m = m_d \cdot K + m_r, n = n_d \cdot K + n_r$. X_0 and M would be divided into K^2 blocks, $(K-1)^2$ among them are of size $m_d \times n_d$, $K-1$ among them are of size $m_r \times n_d$, $K-1$ among them are of size $m_d \times n_r$, and the rest one is of size $m_r \times n_r$. A (or L) would be divided into K^2 blocks, $(K-1)^2$ among them are of size $m_d \times m_d$ (or $n_d \times n_d$), $K-1$ among them would be of size $m_d \times m_r$ (or $n_d \times n_r$), $K-1$ among them would be of $m_r \times m_d$ (or $n_r \times n_d$), and the rest would be $m_r \times m_r$ (or $n_r \times n_r$ respectively). W (or H) would be divided into K blocks, $K-1$ among them are of size $m_d \times p$ (or $p \times n_d$), and the rest one is of size $m_r \times p$ (or $p \times n_r$ respectively). For example, as illustrated in Figure 1 (d) (pipeline figure) (d), if $K = 5$ and A is a 10×10 matrix, then we would have 25 blocks with size 2×2.

3 Interchangeability

To ensure the independence of each process, interchangeability [10] of the index quadruple set U^t should be maintained so that the optimization of (W^i, W^j, H^r, H^s) won't affect another pairs.

Definition of Interchangeability [10]
$\mathcal{U}_1$,$\mathcal{U}_2$ are interchangeable sets concerning a loss function $\mathcal{L}$ if any two instances $u_1 \in \mathcal{U}_1$ and $u_2 \in \mathcal{U}_2$ are interchangeable, where u_1, u_2 are interchangeable if

$$\begin{aligned} \nabla \mathcal{L}_{u_1}(\theta) = \nabla \mathcal{L}_{u_1}(\theta - \epsilon \nabla \mathcal{L}_{u_2}(\theta)), \\ \nabla \mathcal{L}_{u_2}(\theta) = \nabla \mathcal{L}_{u_2}(\theta - \epsilon \nabla \mathcal{L}_{u_1}(\theta)), \end{aligned} \tag{3}$$

According to [10] , the interchangeability can be maintained while quadruples (i_1, j_1, r_1, s_1) and (i_2, j_2, r_2, s_2) do not coincide, to be precise, we need $i_1 \neq i_2, j_1 \neq j_2, r_1 \neq r_2, s_1 \neq s_2, i_1 \neq j_2, i_2 \neq j_1, r_1 \neq s_2$ and $r_2 \neq s_1$. For example, $(1, 2, 1, 2)$ and $(3, 4, 5, 6)$ do not coincide, then we can update (W^1, W^2, H^1, H^2) and (W^3, W^4, H^5, H^6) in parallel. By interchangeability, $|U^t|$, the cardinality of the index quadruple set would not exceed the number of blocks, which we prescribed as the number of parallelized processes.

4 Algorithm 1

```
Algorithm 1 BANMF-S
Input: $X_0 \in \mathcal{R}^{m\times n}, M \in \{0,1\}^{m\times n}, p \ll m, p \ll n$, $K \in \mathbb{N}$ , $\gamma_1, \gamma_2, \alpha_1, \alpha_2 > 0$, $\eta_t > 0$
Register $K$ processes
Calculate high-order gene similarity matrix $S^g$ and the Laplacian $L$
Calculate cell similarity matrix $A$
Initialize $W$ and $H$ according to k-means
Partition $L$, $A$, $M$, $X_0$ and the corresponding $W$ and $H$ into blocks
repeat
    Randomly generate a set of interchangeable quadruples of indices $U^t = \{(i_1^t, j_1^t, r_1^t, s_1^t), (i_2^t, j_2^t, r_2^t, s_2^t), \cdots\}$
    For $(i,j,r,s) \in U^t$, $W_{t+1}^i = W_t^i - \eta_t \nabla_{W_i} \tilde{O}, W_{t+1}^j = W_t^j - \eta_t \nabla_{W_j} \tilde{O}, H_{t+1}^r = H_t^r - \eta_t \nabla_{H_i} \tilde{O}, H_{t+1}^s = H_t^s - \eta_t \nabla_{H_s} \tilde{O}$
    $F_{t+1} \leftarrow ||X_0 - M \circ (W_{t+1}H_{t+1})||_F^2 + \gamma_1 \|A - W_{t+1}W_{t+1}^T\|_F^2 + \gamma_2 Tr(H_{t+1}LH_{t+1}^T) + \alpha_1 \|W_{t+1}\|_F^2 + \alpha_2 \|H_{t+1}\|_F^2$
until $|F_{t+1} - F_t| < \epsilon$
Output: $W, H$
```

5 Supplementary Information for Data

5.1 Dataset Information

Details regarding the datasets are given in Table 1. The accession number, source download Uniform Resource Locators (URLs), original data sizes, filtered data sizes and number of clusters (time stamps) are provided in Table 1.

Short Name	Accession	Species	# Genes (Raw)	# Cells (Raw)	# Genes (Filtered)	# Cells (Filtered)	# Clusters (Time Stamps)
Petropoulos	E-MTAB-3929	human	21749	1529	16202	1529	5
Scialdone	`https://gastrulation.stemcells.cam.ac.uk`	mouse	41388	1205	16941	1205	4
Pollen	`https://github.com/gongx030/scDatasets`	human	21471	299	14194	299	11
Deng	`https://github.com/gongx030/scDatasets`	mouse	18884	286	14230	286	10
Baron-Hm	GSE84133-GSM2230760	human	20125	1303	9189	1303	14
Baron-Ms	GSE84133-GSM2230761	mouse	14878	822	7339	822	13
PBMC	`https://www.10xgenomics.com`	human	19867	42504	9189	1974	5

Table 1: Dataset Description

Remark: pbmc10k dataset was obtained from `https://www.10xgenomics.com`, and was further devided into *cell 1k, 3k, 5k, 7k, 10k* and *gene 1k, 3k, 5k, 7k, 10k* datasets. The bulk immune cell RNA-seq data was obtained from GSE74246.

5.2 Methods Summary

We choose seven methods for comparison and provide the platform information, package version and source in Table 2.

Methods	Platform	Version	Source
ALRA	R	0.0.0.9000	`https://github.com/KlugerLab/ALRA`
bayNorm	R	1.5.14	`https://github.com/WT215/bayNorm`
DrImpute	R	1.2	`https://github.com/gongx030/DrImpute`
MAGIC	Python	3.0.0	`https://magic.readthedocs.io/en/stable/`
SAVER	R	1.1.3	`https://github.com/mohuangx/SAVER`
scImpute	R	0.0.9	`https://github.com/Vivianstats/scImpute`
scRMD	R	0.99.0	`https://github.com/XiDsLab/scRMD`
monocle2	R	2.26.0	`http://cole-trapnell-lab.github.io/monocle-release/`

Table 2: Methods Description

5.3 Supplementary Tables for Simulation Study

Table 3 provides matrix density ratio for the downsampled matrix and the original PBMC dataset. Table 4 records the downsampling rates for *Simulation 2*. Table 5 gives the RMSE results for the simulation study.

	downrate 30	downrate 35	downrate 40	downrate 45	downrate 50	downrate 55	downrate 60
Matrix Density Ratio	1.31	1.22	1.12	1.03	0.94	0.84	0.75

Table 3: Matrix Density Ratio for the Downsampled Matrix and Original PBMC

	B cell	CD4+ T cell	CD8+ T cell	Monocyte	NK cell
$\boldsymbol{p}_i$	0.45	0.35	0.45	0.45	0.35

Table 4: Down-sampling Rates for *Simulation 2*

method	downrate 30	downrate 35	downrate 40	downrate 45	downrate 50	downrate 55	downrate 60	B cell	CD4+ T cell	CD8+ T cell	Monocyte	NK cell
BANMF-S	1.3646	1.3553	1.3553	1.3223	**1.3469**	**1.3616**	**1.4072**	**1.1385**	1.2651	**1.1974**	**1.0603**	1.2508
ALRA	0.9758	1.1765	1.1213	1.2454	1.4202	1.3805	1.5288	1.1614	1.0096	1.2137	1.0940	1.0942
bayNorm	2.4843	2.6724	2.8844	2.9594	2.9593	2.9592	2.9598	2.2686	1.9292	2.2258	2.1552	1.9213
DrImpute	1.6919	1.7383	1.7911	1.8501	1.9118	1.9814	2.0574	1.5979	1.6476	1.6583	1.4350	1.7497
MAGIC	1.7172	1.7677	1.8224	1.8830	1.9479	2.0209	2.0980	1.6151	1.6494	1.6661	1.4255	1.7671
SAVER	2.4912	2.5329	2.5722	2.6116	2.6483	2.6846	2.7197	2.1897	2.2294	2.2205	1.9874	2.3425
scImpute	1.9196	1.9269	1.9231	1.9297	1.9444	1.9382	1.9668	1.6802	1.8312	1.7479	1.4903	1.9433
scRMD	1.7069	1.7229	1.7456	1.7743	1.8101	1.8532	1.9070	1.5037	1.6201	1.5745	1.3939	1.6955

Table 5: RMSE

6 ARI and NMI Results for Clustering

Table 6 and 7 provides ARI and NMI results for clustering study.

dataname	ALRA	bayNorm	DrImpute	MAGIC	noimp	SAVER	scImpute	scRMD	BANMF-S	BANMF-S-latent
Baron_Hm	0.4317	0.3093	**0.5464**	0.4390	0.4329	0.4537	0.4242	0.3976	0.4473	0.5339
Baron_Ms	0.3776	0.3208	0.4643	0.3747	0.3904	0.3959	0.3941	0.3900	**0.4774**	**0.5806**
Deng	0.4806	0.3851	0.4559	0.4837	0.4015	0.4147	0.4339	0.3895	**0.4908**	0.4773
PBMC	0.6380	0.4418	0.6372	0.8014	0.4315	0.4671	0.4657	0.4308	**0.8896**	**0.8991**
Petropoulos	0.2332	0.2959	0.3374	0.3822	0.3039	0.3459	0.3459	0.2894	**0.4942**	**0.4965**
Pollen	0.6089	0.5351	0.6195	0.6483	0.6679	0.6253	0.6164	0.7252	0.6318	**0.7473**
Scialdone	0.5765	0.6012	0.6115	0.5801	0.5686	0.5768	0.5768	0.5713	0.5725	0.5688

Table 6: ARI

dataname	ALRA	bayNorm	DrImpute	MAGIC	noimp	SAVER	scImpute	scRMD	BANMF-S	BANMF-S-latent
Baron_Hm	0.5965	0.4580	**0.6390**	0.6023	0.5693	0.6021	0.5649	0.5466	0.5885	0.6180
Baron_Ms	0.5643	0.4683	0.6195	0.5475	0.5594	0.5699	0.5563	0.5627	0.6145	**0.6406**
Deng	0.7008	0.5748	0.6729	0.6619	0.5747	0.6131	0.6472	0.5796	0.6678	**0.6956**
PBMC	0.7156	0.4019	0.7147	0.7259	0.5401	0.5799	0.5733	0.5388	**0.8141**	**0.8236**
Petropoulos	0.3656	0.4279	0.4815	0.5180	0.4366	0.4843	0.4840	0.4284	**0.5727**	**0.5734**
Pollen	0.8031	0.7643	0.8031	0.8130	0.8234	0.8031	0.7969	0.8478	0.8031	**0.8669**
Scialdone	0.4990	0.5123	0.5179	0.5004	0.4884	0.4934	0.4986	0.4766	0.4807	0.4749

Table 7: NMI

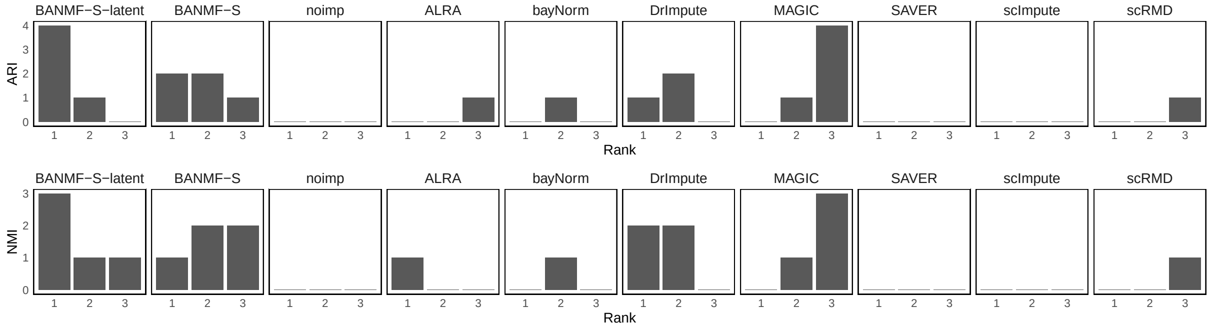

Figure 1: Clustering Results Summary

7 Experiments

7.1 Datasets

We adopted eight published real scRNA-seq datasets to validate BANMF-S's performance, they were named as Baron (including Baron_Hm and Baron_Ms), PBMC, Deng, Pollen, Petropoulous, Scialdone and pbmc10k. These datasets would be used to evaluate whether the imputed expression profiles could enhance the performance of downstream tasks. Specially, Petropoulos and Scialdone have time stamps, they were used for pseudotime trajectory inference; the other five datasets have cell type labels, they were used for clustering. Petropoulous and Scialdone have also been used for testing clustering performance, where we used the time stamps as reference labels. Detailed information on the eight data sets are listed as follow, and they were processed in accordance with Section Data Preprocessing.

- Baron [1]: We used Baron_Hm (Baron_Ms) for the human (mouse) cell dataset, which are obtained by inDrop-seq, a droplet-based sequence technique. The filtered Baron_Hm (Baron_Ms) dataset contains 1303 (822) cells and 9189 (7339) genes, annotated as 14 (13) cell types by known markers through hierarchical clustering in the source paper. We regarded the given cell labels as gold standard.

- Deng [4]: The Deng dataset contains 10 cell types sampled from mouse preimplantation embryos. The filtered matrix contained 14230 genes and 286 cells.

- PBMC: The raw PBMC (peripheral blood mononuclear cells) dataset was downloaded from 10x Genomics website, consisting of 42504 human peripheral blood mononuclear cells from five cell types (B cell, CD4+ T cell, CD8+ T cell, NK cell and Monocyte). The cell labels were annotated through FACS technique based on surface markers. The filtered matrix includes 1974 cells and 9189 genes.

- Pollen [12]: The raw Pollen dataset involves 299 cells from human cerebral cortex, which can be further classified into 11 groups at the cell line level, where the K562, HL60, CRL-2339 cell lines come from blood cell tissue, the Kera, BJ and CRL-2338 cell lines are from dermal cell tissue, the NPC (Neural Progenitor Cells), GW16, GW21 and GW21+3 cell lines are nerve cells, and iPS cells are human-induced stem cells. We used the given cell line labels as cluster labels in clustering analysis and 14194 genes were remained after the filtering.

- Petropoulos[11]: The Petropoulos dataset characterizes the transcriptomic profiles of human preimplantation development, which includes 1529 cells from five (E3, E4, E5, E6 and E7). We used the time stamps as reference labels in clustering analysis.
- Scialdone [14]: The Scialdone dataset records single cell transcriptomics from mouse mesodermal development covering timecourse samples from early gastrulation at embryonic day E6.5 to primitive red blood cells generation at E7.75 (HF), also including E7 (PS) and NP (E7.5).
- pbmc10k: The pbmc10k dataset was obtained from 10x Genomics website that contains 11571 cells and 13570 genes after filtering and was used to assess computational efficiency.

We curated another 12 simulated data sets in two ways, namely, *simulation 1* and *simulation 2*. In *simulation 1*, we retained high quality genes and cells from PBMC dataset, and then randomly removed non-zero reads following a Binomial distribution with varying dropout rates, and obtained 7 datasets. In *simulation 2*, we generated 5 single cell data sets by Multinomial distribution using gene proportions of bulk immune cell profilings and library lengths in the PBMC single cell data set. Details of simulation procedures are provided in Section Results.

7.2 Data Preprocessing

For those real datasets, we first removed ERCC spike-ins and mitochondrial genes if necessary. Then performed Quality Control (QC) by removing genes that are recorded in less than 5% of total cells and cells whose capture rates are less than 1%. Since the original PBMC dataset is highly sparse with 2.95% non-zero entries, we applied a lenient QC policy there. Within each cell types of PBMC, genes with less than 0.1% expressions were removed while cell expressed in more than 5% genes were retained. The accession number, source Uniform Resource Locators (URLs), original data sizes, filtered data sizes and number of clusters (time stamps) are provided in Supplementary Table 1.

After quality control, all data were normalized to 10^4 counts per cell, followed by adding one pseudocount, and then the sum was log-2 transformed. That is, if the raw count matrix is denoted by $\tilde{X}$, then the normalized matrix $\tilde{X}_{norm}$ is computed by

$$[\tilde{X}_{norm}]_{ij} = \frac{\tilde{X}_{ij}}{\sum_j \tilde{X}_{ij}} \times 10^4$$

and then the log-transformed matrix X_0 is given by

$$[X_0]_{ij} = \log_2([\tilde{X}_{norm}]_{ij} + 1).$$

While we used X_0 as the input expression matrix for BANMF-S, , various processed expression matrices were used for the other seven state-of-the-art algorithms as the algorithms required. For example, if any method requires original counts , then take $\tilde{X}$ as an input. Since evaluations were compared in log-transformed matrices, post-processing were added in accordance with pre-processing after imputation if outputs were normalized data or count data.

7.3 State-of-the-art Algorithms

To validate the effectiveness of BANMF-S, we chose seven state-of-the-art algorithms for comparison, they are SAVER [7], MAGIC [16], scImpute [8], DrImpute [6], bayNorm [15], scRMD [3] and ALRA[9]. SAVER employs a penalized regression method to estimate gene-gene correlations and a Poisson-Gamma model is built for imputation. MAGIC utilizes the powered Markov transition matrix, created by normalizing cell-to-cell affinity matrix, to compute the "overall" similarity across cells and then "smooth" across the whole sample. After identifying similar cells from different clustering methods, within each cell sub-populations, DrImpute uses the averaged expression to estimate missing values while scImpute builds a Gamma-Normal mixture model to make further inference. ALRA adopts matrix factorization for recovery and designs an adaptive thresholding strategy for non-negative constraint.

7.4 Evaluation

In the simulation studies, we employed the Rooted Mean Squared Error ($RMSE$), the average squared difference between the imputed values and the actual values, to check whether the imputation method is able to handle the technical noises,

$$RMSE = \sqrt{\frac{\sum\limits_{(i,j)\in\mathcal{D}} ([X_{imp}]_{ij} - [X_0]_{ij})^2}{|\mathcal{D}|}} \tag{4}$$

Here X_{imp} denotes the imputed expression matrix, $\mathcal{D}$ refers to the collection of dropout positions in X_0, and $|\mathcal{D}|$ represents the number of dropouts. Moreover, we used the cell-wise correlation between the imputed scRNA-seq profile and the reference profile to evaluate an imputation method's capability to recover sample-level biological expression, which is given by,

$$\text{Cell-wise Correlation} = [corr(X_{imp}[i,:], X_0[j,:])], \\ i,j \in \{1,2,\cdots,m\}, i,j \in \mathcal{C}$$

Here $X_{imp}[i,:]$ (or $X_0[j,:]$) means $i-$th row of X_{imp} (or X_0) respectively. Moreover, cell i and cell j belonged to a same cluster $\mathcal{C}$.

In the real data studies, the accuracy of cell type clustering was evaluated from two metrics, namely Adjust Rand index (ARI) and Normalized Mutual Information (NMI). ARI is a popular matching coefficient in evaluating classification accuracy. Let $A = \{A_1, \cdots, A_r\}$ and $B = \{B_1, \cdots, B_s\}$ be two clusterings of a collection of N cells. For $r \in \{1, \cdots, R\}$ and $s \in \{1, \cdots, S\}$, let n_{rs} be the number of cells which are assigned to the label A_r and the label B_s.

If $a_r = \sum_s n_{rs}$ and $b_s = \sum_r n_{rs}$ are respectively the number of cells in the labels A_r and B_s, then the ARI is calculated by

$$ARI(A,B) = \frac{\sum_{rs}\binom{n_{rs}}{2} - \left[\sum_r \binom{a_r}{2}\sum_s \binom{b_s}{2}\right] \bigg/ \binom{N}{2}}{\frac{1}{2}\left[\sum_r \binom{a_r}{2} + \sum_s \binom{b_s}{2}\right] - \left[\sum_r \binom{a_r}{2}\sum_s \binom{b_s}{2}\right] \bigg/ \binom{N}{2}}$$

On the other hand, NMI is a measure that quantifies the similarity between two sets of data by assessing their mutual information,

$$NMI(A,B) = \frac{2 \times I(A,B)}{H(A) + H(B)}$$

where $I(A,B)$ stands for the mutual information between the partition A and B and $H(\cdot)$ represents the entropy. The ranges of these two metrics are from 0 to 1, the larger the better. We use the function `adjustedRandIndex` in `R` package `mclust` (version 6.0.0) and function `NMI` in `R` package `aricode` (version 1.0.2) to compute these two metrics.

Monocle2 [13] is a computational method that infers lineage relationships of individual cells by constructing a principal tree, which represents the progression trajectory of the given samples. Subsequently, cells are ordered along the learned graph, and their relative geodesic distances to the initial cell state are computed as the corresponding "pseudotime" of the cellular transition along the dynamic progress. The accuracy of trajectory inference was assessed through the correlation between the given time stamp $\mathtt{t}_1$ and the computed pseudotime orders $\mathtt{t}_2$, i.e., $|corr(\mathtt{t}_1, \mathtt{t}_2)|$. In this paper, we considered Pearson correlation and Kendall's correlation.

8 BANMF-S is an Efficient Algorithm

The blocklization strategy improves the computational efficiency in two ways. On the one hand, it enables BANMF-S to solve the traditional NMF problem by SGD in parallelization, saving wallclock time for large-scale datasets. On the other hand, it allows BANMF-S to improve computational memory cost by circumventing direct large-scale matrix computations, and therefore, avoids the storage of numerous large-scale intermediate matrices. As is shown in the memory plots in Figures 4 (b-c), the slopes for the matrix-based methods, scRMD and ALRA, are larger than BANMF-S. To explain this, scRMD utilized ADMM to solve the robust matrix decomposition problem in Eq (5),

$$\begin{aligned} &\min_{L,S} \frac{1}{2}\|Y - L + S\|_F^2 + \lambda\|L\|_* + \tau\|S\|_1 \\ &\text{subject to } P_\Omega(S) \geq 0, P_{\Omega^c}(S) = 0, \text{ and } L \geq 0, \end{aligned} \tag{5}$$

where $Y \in \mathbb{R}^{m\times n}$ refers to the observed matrix, $\Omega \in \mathbb{R}^{m\times n}$ represents the projection matrix, and $L, S \in \mathbb{R}^{m\times n}$ are low-rank and sparsity restrictions (see [3] for details). Apart from tracing five large-scale matrices, Y, Ω, L, S and the recovered expression matrix `exprs`, scRMD introduces latent varible $Z \in \mathbb{R}^{m\times n}$ and the Lagrange multiplier $\Lambda \in \mathbb{R}^{m\times n}$ in the ADMM algorithm and uses two extra $m-$by$-n$ matrices `Z_hat` and `L_old` as intermediate variables, which is computationally expensive. In real studies, manual garbage collection plays an important role in improving memory performance, by which unused variables needs deallocation as soon as possible. Diving into the source codes, we found that scRMD failed to manually remove $m-$by$-n$ initialization variables such as `initL`, `initS` and `initLambda`, which would be consequently tracked as part of the peak memory usage of the entire job by the Slurm workload manager. Similar to scRMD, ALRA needs to store several $m-$by$-n$ matrices, the observed matrix, the mask matrix, the $K-$SVD factorized low-rank matrix, and five other intermediate matrices, named by `A_norm_rank_k_mins`, `A_norm_rank_k_cor`, `A_norm_rank_k_temp`, `A_norm_rank_k_cor_sc` and `lt0`. With those intermediate matrix variables, scRMD and ALRA may be resource-acceptable for small-scale datasets, but resource-intensive, even detrimental when confronted with large-scale datasets. Back to BANMF-S, our method first restores $X_0, M \in \mathbb{R}^{m\times n}$, $A \in \mathbb{R}^{m\times m}$ and $L \in \mathbb{R}^{n\times n}$ in the global environment. In our core computational module, rather than the direct manipulation of the $m-$by$-n$ matrix, we tackled matrices of $\mathcal{O}(m_d n_d)$. At each iteration, we sampled block quadruples to K registered pipes (processes in the context of parallelization), where each pipe contained variables of $\{A^{ij} \in \mathbb{R}^{m_d\times m_d}, L^{rs} \in \mathbb{R}^{n_d\times n_d}, W^i, W^j \in \mathbb{R}^{m_d\times p}, H^r, H^s \in \mathbb{R}^{p\times n_d}, X_0^{ir}, X_0^{is}, X_0^{jr}, X_0^{js}, M^{ir}, M^{is}, M^{jr}, M^{js} \in \mathbb{R}^{m_d\times n_d}\}$ and the derivatives $\{\nabla_{W^i}\tilde{O}, \nabla_{W^j}\tilde{O} \in \mathbb{R}^{m_d\times p}, \nabla_{H^r}\tilde{O}, \nabla_{H^s}\tilde{O} \in \mathbb{R}^{K\times n_d}\}$. To sum up all processes, the maximum memory requirement of our computational module can be regarded as $K \cdot m_d n_d + K \cdot m_d^2 + K \cdot n_d^2$, which demonstrates considerable improvements in terms of memory compared to the whole scale.

9 UMAP Plots of Real Dataset Results

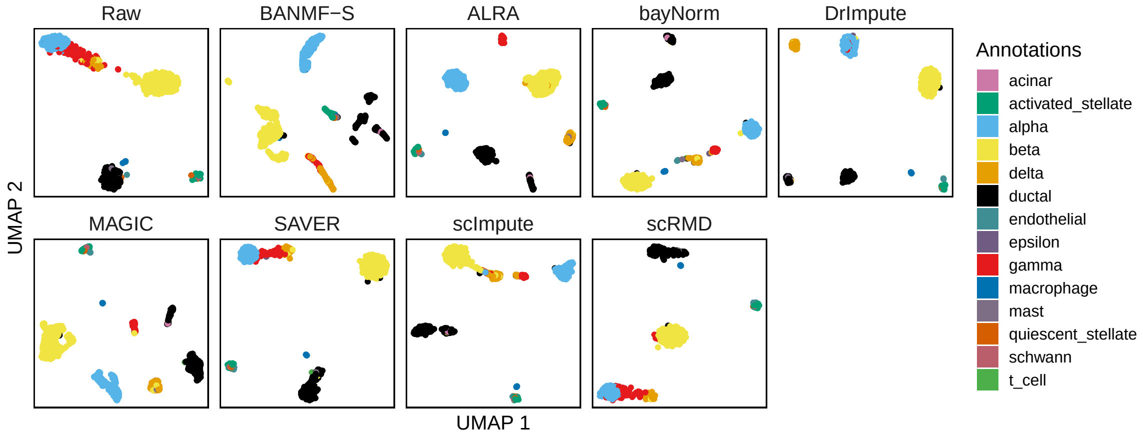

Figure 2: UMAP results for Baron_Hm

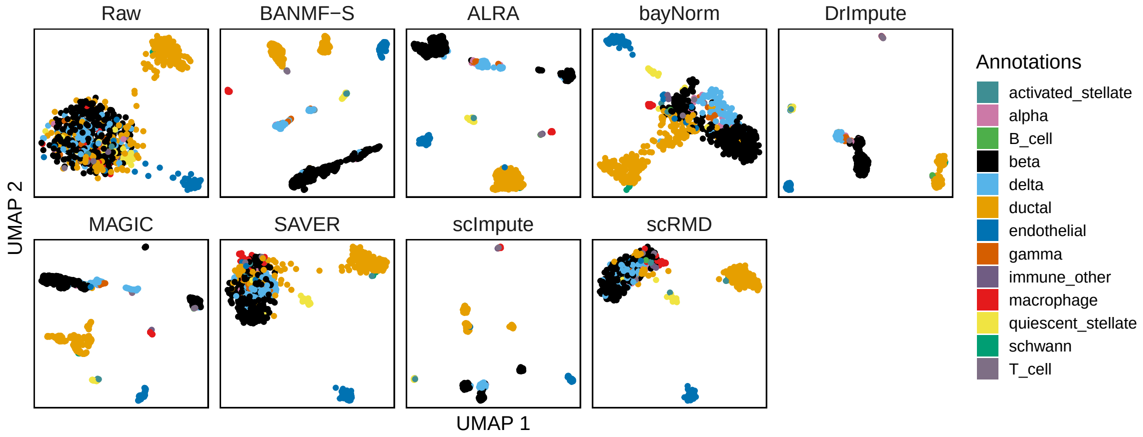

Figure 3: UMAP results for Baron_Ms

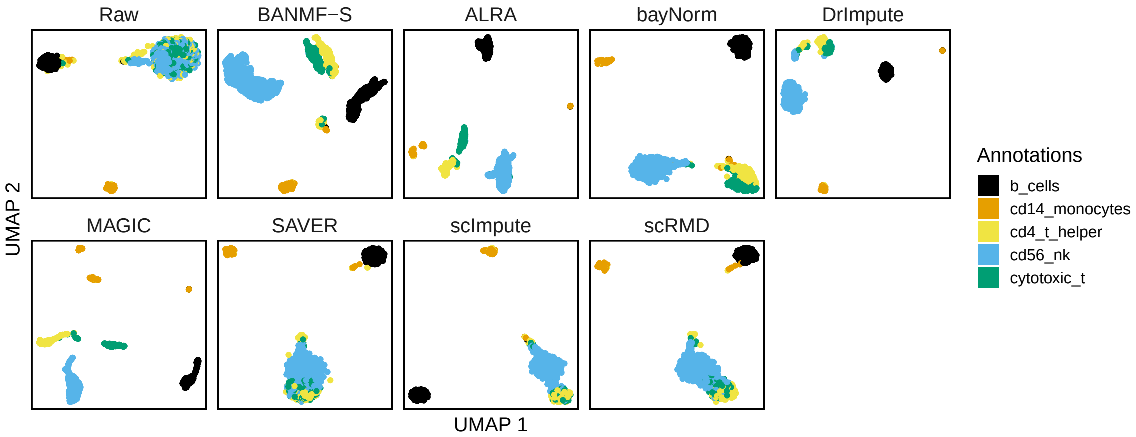

Figure 4: UMAP results for PBMC

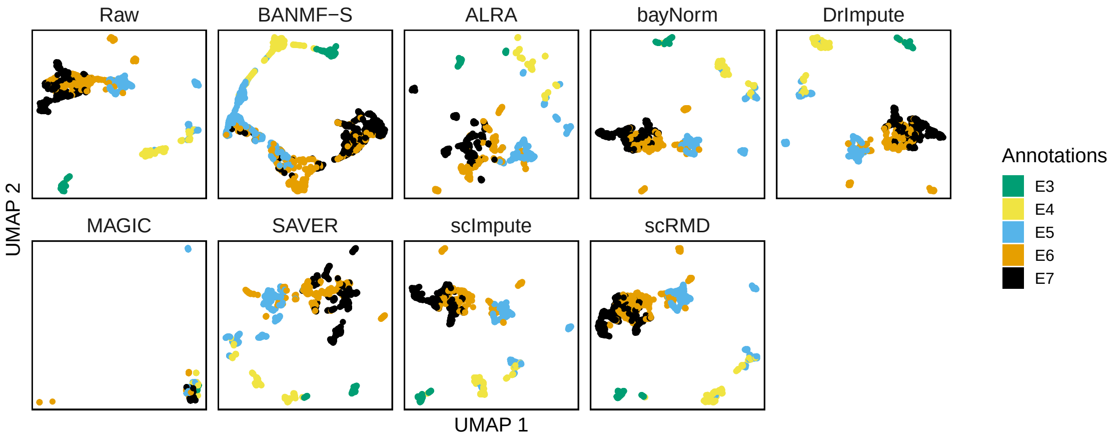

Figure 5: UMAP results for Petropoulos

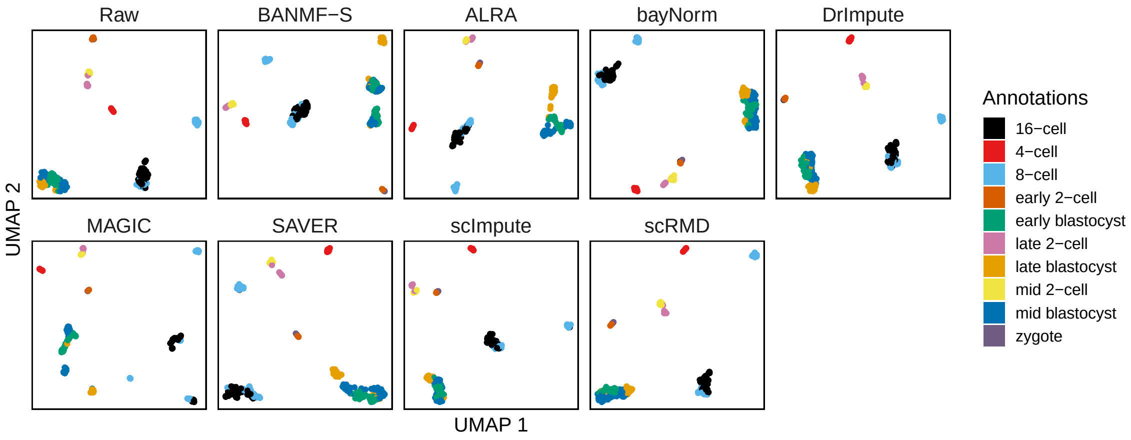

Figure 6: UMAP results for Deng

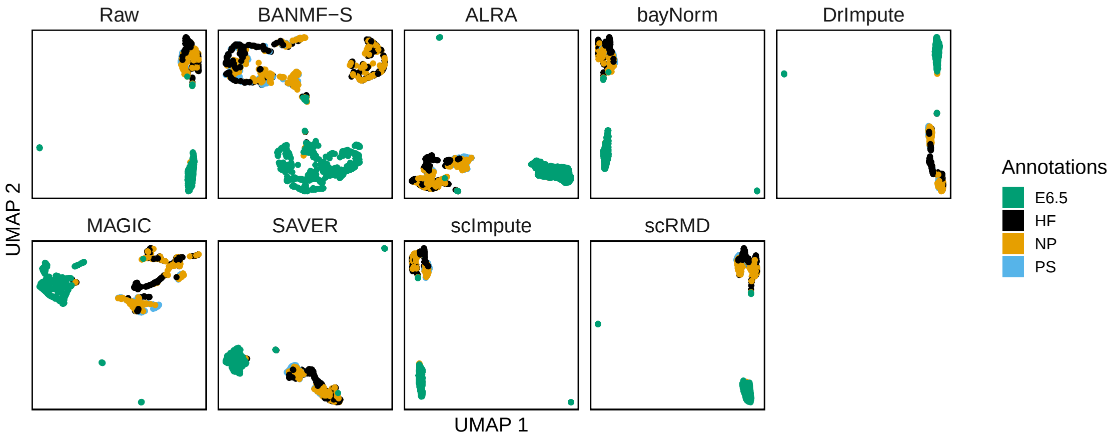

Figure 7: UMAP results for Scialdone

10 Imputation Recovery Rate for Simulated Datasets

method	Original sparsity	Imputed sparsity	Recovery rate	Dataset
ALRA	0.1541	0.9462	0.9847	Downrate 30
BANMF-S	0.1541	1.0000	1.0000	Downrate 30
bayNorm	0.1541	0.1791	0.2465	Downrate 30
DrImpute	0.1541	0.9960	0.9992	Downrate 30
MAGIC	0.1541	1.0000	1.0000	Downrate 30
SAVER	0.1541	1.0000	1.0000	Downrate 30
scImpute	0.1541	0.5450	0.7766	Downrate 30
scRMD	0.1541	0.2682	0.5353	Downrate 30
ALRA	0.1432	0.8670	0.9527	Downrate 35
BANMF-S	0.1432	1.0000	1.0000	Downrate 35
bayNorm	0.1432	0.1551	0.1438	Downrate 35
DrImpute	0.1432	0.9978	0.9998	Downrate 35
MAGIC	0.1432	1.0000	1.0000	Downrate 35
SAVER	0.1432	1.0000	1.0000	Downrate 35
scImpute	0.1432	0.5312	0.7668	Downrate 35
scRMD	0.1432	0.2578	0.5249	Downrate 35
ALRA	0.1322	0.9340	0.9796	Downrate 40
BANMF-S	0.1322	1.0000	1.0000	Downrate 40
bayNorm	0.1322	0.1373	0.0576	Downrate 40
DrImpute	0.1322	0.9980	0.9998	Downrate 40
MAGIC	0.1322	1.0000	1.0000	Downrate 40
SAVER	0.1322	1.0000	1.0000	Downrate 40
scImpute	0.1322	0.5193	0.7639	Downrate 40
scRMD	0.1322	0.2450	0.5096	Downrate 40
ALRA	0.1212	0.9076	0.9689	Downrate 45
BANMF-S	0.1212	1.0000	1.0000	Downrate 45
bayNorm	0.1212	0.1212	0.0000	Downrate 45
DrImpute	0.1212	0.9980	0.9998	Downrate 45
MAGIC	0.1212	1.0000	1.0000	Downrate 45
SAVER	0.1212	1.0000	1.0000	Downrate 45
scImpute	0.1212	0.5049	0.7528	Downrate 45
scRMD	0.1212	0.2304	0.4901	Downrate 45
ALRA	0.1102	0.8605	0.9464	Downrate 50
BANMF-S	0.1102	1.0000	1.0000	Downrate 50
bayNorm	0.1102	0.1102	0.0000	Downrate 50
DrImpute	0.1102	0.9977	0.9997	Downrate 50
MAGIC	0.1102	1.0000	1.0000	Downrate 50
SAVER	0.1102	1.0000	1.0000	Downrate 50
scImpute	0.1102	0.4853	0.7370	Downrate 50
scRMD	0.1102	0.2139	0.4668	Downrate 50
ALRA	0.0989	0.9469	0.9833	Downrate 55
BANMF-S	0.0989	1.0000	1.0000	Downrate 55
bayNorm	0.0989	0.0989	0.0000	Downrate 55
DrImpute	0.0989	0.9975	0.9997	Downrate 55
MAGIC	0.0989	1.0000	1.0000	Downrate 55
SAVER	0.0989	1.0000	1.0000	Downrate 55
scImpute	0.0989	0.4796	0.7367	Downrate 55
scRMD	0.0989	0.1965	0.4410	Downrate 55
ALRA	0.0879	0.9452	0.9812	Downrate 60
BANMF-S	0.0879	1.0000	1.0000	Downrate 60
bayNorm	0.0879	0.0879	0.0000	Downrate 60
DrImpute	0.0879	0.9973	0.9996	Downrate 60
MAGIC	0.0879	1.0000	1.0000	Downrate 60
SAVER	0.0879	1.0000	1.0000	Downrate 60
scImpute	0.0879	0.4521	0.7107	Downrate 60
scRMD	0.0879	0.1764	0.4072	Downrate 60

Table 8: Recovery rate for Simulation 1

method	Original sparsity	Imputed sparsity	Recovery rate	Dataset
ALRA	0.1193	0.7137	0.9573	B cell
BANMF-S	0.1193	1.0000	1.0000	B cell
bayNorm	0.1193	0.1439	0.1994	B cell
DrImpute	0.1193	0.9688	0.9996	B cell
MAGIC	0.1193	0.9990	1.0000	B cell
SAVER	0.1193	1.0000	1.0000	B cell
scImpute	0.1193	0.4822	0.8557	B cell
scRMD	0.1193	0.2596	0.6094	B cell
ALRA	0.1314	0.8129	0.9763	CD4+ T cell
BANMF-S	0.1314	0.9999	1.0000	CD4+ T cell
bayNorm	0.1314	0.2154	0.4741	CD4+ T cell
DrImpute	0.1314	0.9815	0.9995	CD4+ T cell
MAGIC	0.1314	0.9999	1.0000	CD4+ T cell
SAVER	0.1314	0.9999	1.0000	CD4+ T cell
scImpute	0.1314	0.4618	0.7871	CD4+ T cell
scRMD	0.1314	0.2558	0.5858	CD4+ T cell
ALRA	0.1176	0.6462	0.9174	CD8+ T cell
BANMF-S	0.1176	1.0000	1.0000	CD8+ T cell
bayNorm	0.1176	0.1467	0.2239	CD8+ T cell
DrImpute	0.1176	0.9881	0.9996	CD8+ T cell
MAGIC	0.1176	0.9996	1.0000	CD8+ T cell
SAVER	0.1176	0.9996	1.0000	CD8+ T cell
scImpute	0.1176	0.4767	0.8242	CD8+ T cell
scRMD	0.1176	0.2519	0.5797	CD8+ T cell
ALRA	0.1446	0.7977	0.9633	Monocyte
BANMF-S	0.1446	1.0000	1.0000	Monocyte
bayNorm	0.1446	0.1676	0.1656	Monocyte
DrImpute	0.1446	0.9773	0.9985	Monocyte
MAGIC	0.1446	0.9999	1.0000	Monocyte
SAVER	0.1446	0.9999	1.0000	Monocyte
scImpute	0.1446	0.5707	0.8865	Monocyte
scRMD	0.1446	0.2826	0.5781	Monocyte
ALRA	0.1193	0.7503	0.9690	NK cell
BANMF-S	0.1193	1.0000	1.0000	NK cell
bayNorm	0.1193	0.2212	0.5376	NK cell
DrImpute	0.1193	0.9825	0.9999	NK cell
MAGIC	0.1193	0.9952	1.0000	NK cell
SAVER	0.1193	1.0000	1.0000	NK cell
scImpute	0.1193	0.4203	0.8082	NK cell
scRMD	0.1193	0.2415	0.5976	NK cell

Table 9: Recovery rate for Simulation 2

11 Trajectory Visualizations

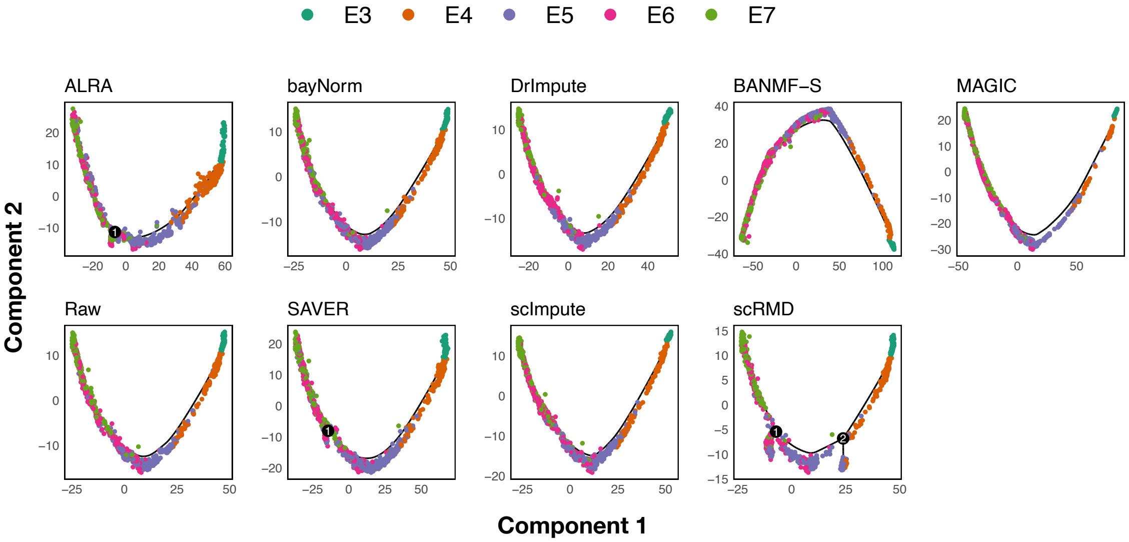

Figure 8: Petropoulos

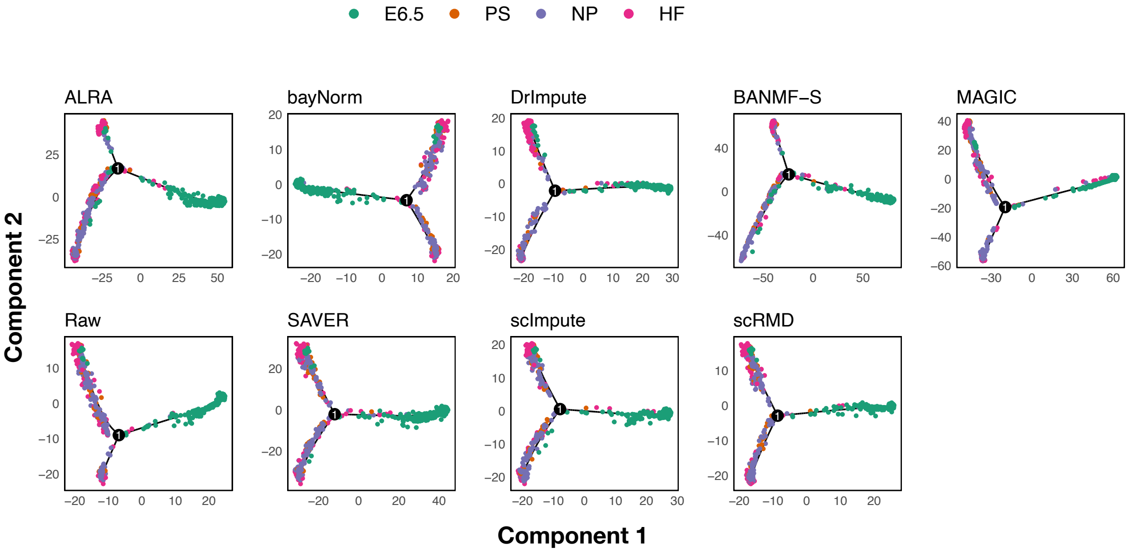

Figure 9: Scialdone

12 Parameters

To evaluate the impact of hyper-parameters on the imputation results, we perform sensitivity analyses on the simulated datasets and assess the imputed matrices under RMSE measurement. To be specific, fixing $\alpha_1 = \alpha_2 = 0.1$, we conduct BANMF-S on *Simulation 1* and *Simulation 2* with the following suites of gammas,

- $\gamma_1 = 0$, $\gamma_2 = 1$
- $\gamma_1 = 1$, $\gamma_2 = 0$
- $\gamma_1 = 0$, $\gamma_2 = 0$
- $\gamma_1 = \gamma_2 = 0.1, 0.5, 1, 5, 10, 50, 100, 500, 1000$

We also study the impact of sparsity penalties by fixing $\gamma_1 = \gamma_2 = 0.1$ and conducting BANMF-S over the following suites of alphas,

- $\alpha_1 = \alpha_2 = 0.1, 0.5, 1, 5, 10, 50, 100, 500, 1000$

Results of sensitivity analyses with respect to $\gamma_1(\alpha_1)$ and $\gamma_2(\alpha_2)$ are demonstrated in the upper(lower) panel of Figure 10 and Figure 11. RMSE exhibits like a U-shape curve as γ_1 and γ_2 jointly increase, which indicates that penalties improve the accuracy of recovered matrices. As α_1 and α_2 increase, RMSE remains relatively stable.

The sensitivity analyses demonstrate that (i) graphical penalties improves the accuracy of recovered matrices; (ii) as long as γ_1, γ_2 is relatively small, i.e., $\gamma_1, \gamma_2 \leq 5$, there is no significant differences under RMSE measurement. We suggest choosing parameters according to the following criterion,

- For matrix of size greater than 10^4, we recommend using $\gamma_1, \gamma_2 \in \{1, 2, 3, 4, 5\}$ and $\alpha_1, \alpha_2 \in \{1, 2, 3, 4, 5\}$.
- For matrix of size less than 10^4, we recommend using $\gamma_1, \gamma_2 \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and $\alpha_1, \alpha_2 \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

Table 10 provides the recorded parameters in this paper.

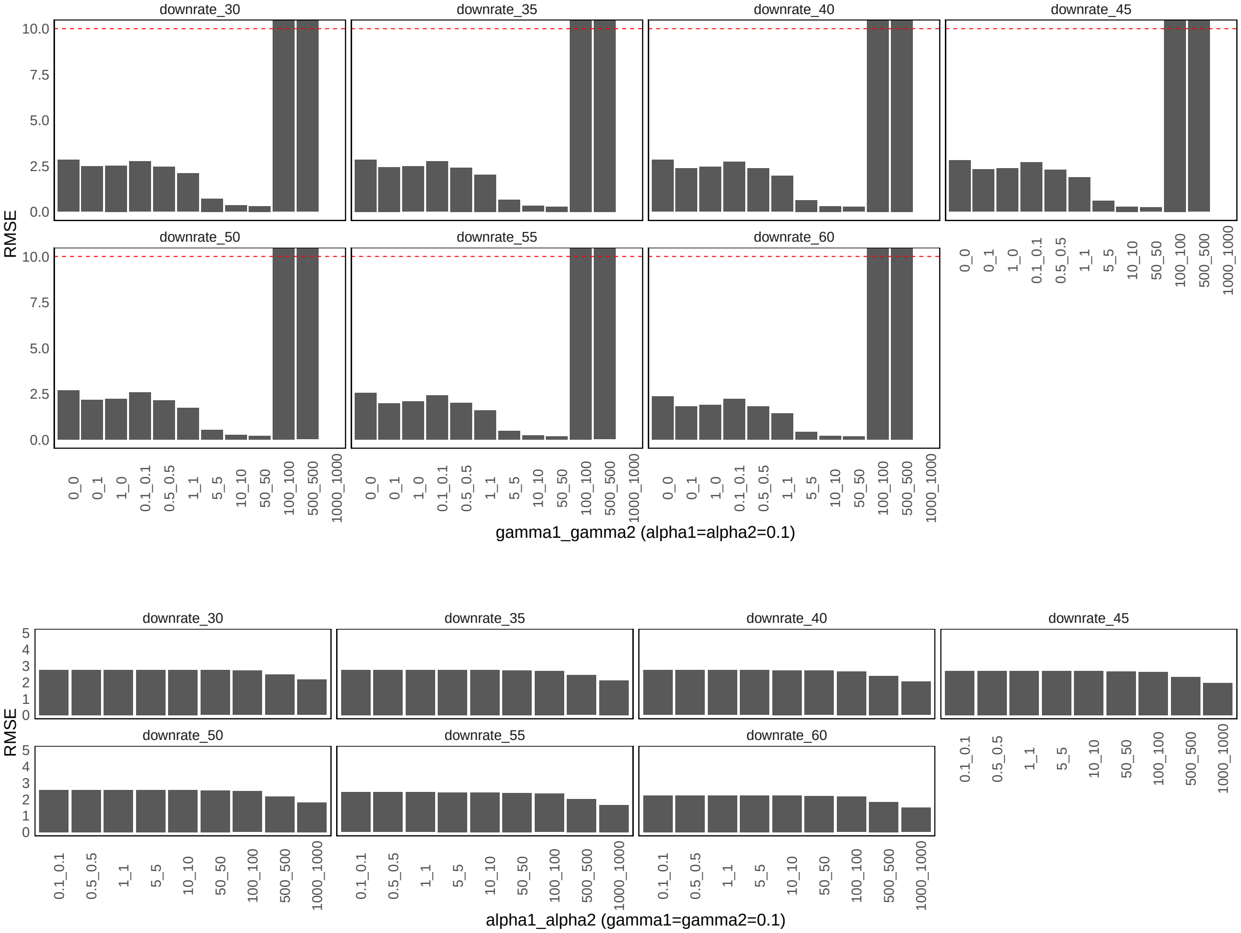

Figure 10: Sensitivity Analysis for *Simulation 1*

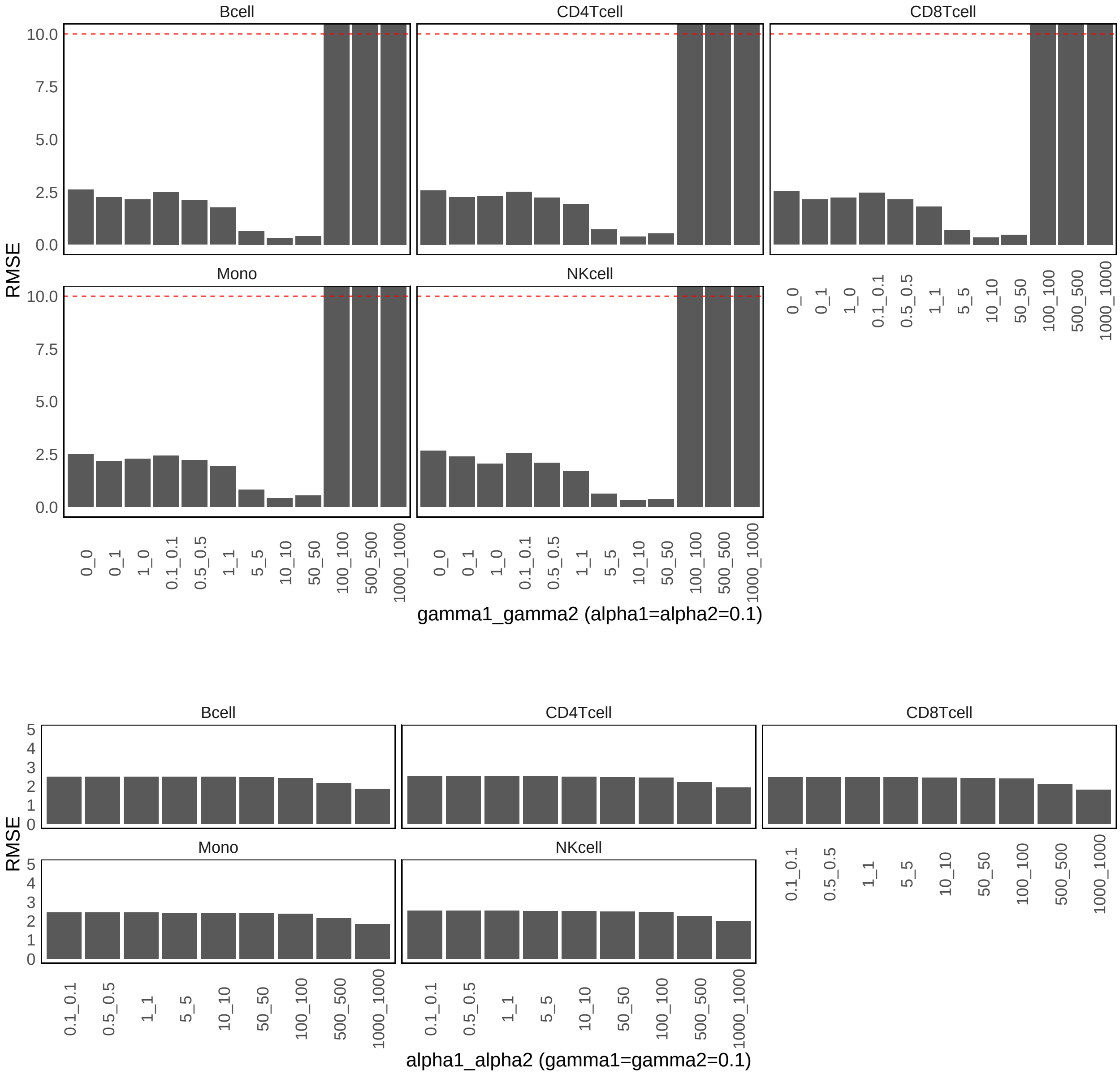

Figure 11: Sensitivity Analysis for *Simulation 2*

Dataname	α_1	α_2	γ_1	γ_2	K
Simulation 1	0.1	0.1	0.1	0.1	15
Simulation 2	0.1	0.1	0.1	0.1	15
PBMC	5	1	5	1	5
Baron_ms	5	1	5	1	15
Baron_hm	5	1	5	1	15
Pollen	0.1	0.1	0.1	0.1	15
Deng	0.5	0.1	0.5	0.1	15
Petropoulos	0.5	0.1	0.5	0.1	15
Scialdone	0.5	0.1	0.5	0.1	15
cell 1k-10k	5	1	5	1	15
gene 1k-10k	5	1	5	1	15

Table 10: Parameters used for each dataset

References

[1] Maayan Baron, Adrian Veres, Samuel L Wolock, Aubrey L Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K Wagner, Shai S Shen-Orr, Allon M Klein, et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell systems*, 3(4):346–360, 2016.

[2] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1548–1560, 2010.

[3] Chong Chen, Changjing Wu, Linjie Wu, Xiaochen Wang, Minghua Deng, and Ruibin Xi. scrmd: imputation for single cell rna-seq data via robust matrix decomposition. *Bioinformatics*, 36(10):3156–3161, 2020.

[4] Qiaolin Deng, Daniel Ramsköld, Björn Reinius, and Rickard Sandberg. Single-cell rnaseq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science*, 343(6167):193–196, 2014.

[5] Chris Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 606–610. SIAM, 2005.

[6] Wuming Gong, Il-Youp Kwak, Pruthvi Pota, Naoko Koyano-Nakagawa, and Daniel J Garry. Drimpute: imputing dropout events in single cell rna sequencing data. *BMC bioinformatics*, 19(1):1–10, 2018.

[7] Mo Huang, Jingshu Wang, Eduardo Torre, Hannah Dueck, Sydney Shaffer, Roberto Bonasio, John I Murray, Arjun Raj, Mingyao Li, and Nancy R Zhang. Saver: gene

expression recovery for single-cell rna sequencing. *Nature methods*, 15(7):539–542, 2018.

[8] Wei Vivian Li and Jingyi Jessica Li. An accurate and robust imputation method scimpute for single-cell rna-seq data. *Nature communications*, 9(1):1–9, 2018.

[9] George C Linderman, Jun Zhao, Manolis Roulis, Piotr Bielecki, Richard A Flavell, Boaz Nadler, and Yuval Kluger. Zero-preserving imputation of single-cell rna-seq data. *Nature communications*, 13(1):192, 2022.

[10] Ninghao Liu, Xiao Huang, and Xia Hu. Accelerated local anomaly detection via resolving attributed networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2337–2343, 2017.

[11] Sophie Petropoulos, Daniel Edsgärd, Björn Reinius, Qiaolin Deng, Sarita Pauliina Panula, Simone Codeluppi, Alvaro Plaza Reyes, Sten Linnarsson, Rickard Sandberg, and Fredrik Lanner. Single-cell rna-seq reveals lineage and x chromosome dynamics in human preimplantation embryos. *Cell*, 165(4):1012–1026, 2016.

[12] Alex A Pollen, Tomasz J Nowakowski, Joe Shuga, Xiaohui Wang, Anne A Leyrat, Jan H Lui, Nianzhen Li, Lukasz Szpankowski, Brian Fowler, Peilin Chen, et al. Low-coverage single-cell mrna sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nature biotechnology*, 32(10):1053–1058, 2014.

[13] Xiaojie Qiu, Qi Mao, Ying Tang, Li Wang, Raghav Chawla, Hannah A Pliner, and Cole Trapnell. Reversed graph embedding resolves complex single-cell trajectories. *Nature methods*, 14(10):979–982, 2017.

[14] Antonio Scialdone, Yosuke Tanaka, Wajid Jawaid, Victoria Moignard, Nicola K Wilson, Iain C Macaulay, John C Marioni, and Berthold Göttgens. Resolving early mesoderm diversification through single-cell expression profiling. *Nature*, 535(7611):289–293, 2016.

[15] Wenhao Tang, François Bertaux, Philipp Thomas, Claire Stefanelli, Malika Saint, Samuel Marguerat, and Vahid Shahrezaei. baynorm: Bayesian gene expression recovery, imputation and normalization for single-cell rna-sequencing data. *Bioinformatics*, 36(4):1174–1181, 2020.

[16] David Van Dijk, Roshan Sharma, Juozas Nainys, Kristina Yim, Pooja Kathail, Ambrose J Carr, Cassandra Burdziak, Kevin R Moon, Christine L Chaffer, Diwakar Pattabiraman, et al. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729, 2018.