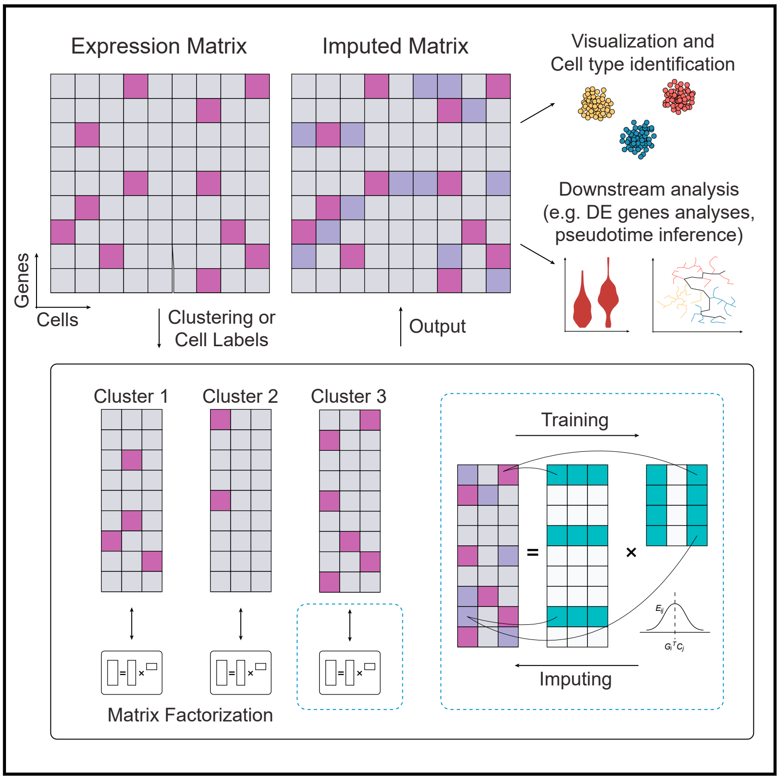
Cell Reports Methods

Article

A Bayesian factorization method to recover single-cell RNA sequencing data

Graphical abstract



Highlights

- Bfimpute recovers dropout events of scRNA-seq data by Bayesian factorization
- Gene expression is imputed through reconstruction of latent gene and cell matrices
- Bfimpute incorporates gene-gene or cell-cell relationships via the latent matrices
- Improved accuracy over a range of existing methods

Authors

Zi-Hang Wen, Jeremy L. Langsam,
Lu Zhang, Wenjun Shen, Xin Zhou

Correspondence

wjshen@stu.edu.cn (W.S.),
maizie.zhou@vanderbilt.edu (X.Z.)

In brief

Wen et al. develop a Bayesian matrix factorization method called Bfimpute for scRNA-seq imputation. Bfimpute reconstructs two latent gene and cell matrices to impute the gene expression matrix within each cell group, with or without the aid of cell type labels or bulk data.

Wen et al., 2022, Cell Reports Methods *2*, 100133
January 24, 2022 © 2021 The Author(s).
https://doi.org/10.1016/j.crmeth.2021.100133

Cell Reports Methods

CellPress
OPEN ACCESS

Article

A Bayesian factorization method to recover single-cell RNA sequencing data

Zi-Hang Wen,[1,2] Jeremy L. Langsam,[2] Lu Zhang,[3] Wenjun Shen,[4,*] and Xin Zhou[2,5,6,7,*]

[1]School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan 430074, China
[2]Department of Biomedical Engineering, Vanderbilt University, Nashville, TN 37235, USA
[3]Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong
[4]Department of Bioinformatics, Shantou University Medical College, Shantou 515041, China
[5]Department of Computer Science, Vanderbilt University, Nashville, TN 37235, USA
[6]Data Science Institute, Nashville, TN 37212, USA
[7]Lead contact
*Correspondence: wjshen@stu.edu.cn (W.S.), maizie.zhou@vanderbilt.edu (X.Z.)

https://doi.org/10.1016/j.crmeth.2021.100133

MOTIVATION Single-cell RNA sequencing (scRNA-seq) is a high-resolution RNA profiling technology that estimates distribution of expression levels across cells. However, missing values that arise due to technical limitations, also known as dropout events, complicate scRNA-seq analysis and limit its utility. Existing imputation methods have a limited ability to reveal cell-cell relationships, complicating cell clustering and trajectory analysis. We reasoned that a more efficient way to recover dropout events would be to incorporate available gene-specific or cell-specific information. Here, we introduce a matrix factorization method to recover dropout events within each cell type and better differentiate cell relationships. We achieve that by decomposing the count matrix into the product of gene-specific and cell-specific feature matrices. Additional gene- or cell-related information available can then be incorporated into the model by Bayesian inference.

SUMMARY

Single-cell RNA sequencing (scRNA-seq) offers opportunities to study gene expression of tens of thousands of single cells simultaneously, to investigate cell-to-cell variation, and to reconstruct cell-type-specific gene regulatory networks. Recovering dropout events in a sparse gene expression matrix for scRNA-seq data is a long-standing matrix completion problem. In this article, we introduce Bfimpute, a Bayesian factorization imputation algorithm that reconstructs two latent gene and cell matrices to impute the final gene expression matrix within each cell group, with or without the aid of cell type labels or bulk data. Bfimpute achieves better accuracy than ten other publicly notable scRNA-seq imputation methods on simulated and real scRNA-seq data, as measured by several different evaluation metrics. Bfimpute can also flexibly integrate any gene- or cell-related information that users provide to increase performance.

INTRODUCTION

Single-cell RNA sequencing (scRNA-seq) has been widely used to study genome-wide transcriptomes in single-cell resolution. The cellular resolution made possible by scRNA-seq data distinguishes it from bulk RNA-seq and makes it advantageous in investigating cell-to-cell variation (Tang et al., 2009). Today, different commercial platforms are available to perform scRNA-seq, including Fluidigm C1, Wafergen ICELL8, and 10x Genomics Chromium. Droplet-based methods via 10x Genomics Chromium can process tens of thousands of cells; microwell-based, microfluidic-based methods via Fluidigm C1 and Wafergen ICELL8 process fewer cells but with a higher sequencing depth. For all these platforms, missing values make up a large proportion of scRNA-seq data, ranging from 40% to 90% in the gene expression count matrix (Chu et al., 2016; Tang et al., 2017; Petropoulos et al., 2016; Zheng et al., 2017; Qiu, 2020; Scialdone et al., 2016; Vladoiu et al., 2019). This large percentage of missing events is defined as the so-called dropout phenomenon (Kharchenko et al., 2014). Gene dropout means a gene is observed at a moderate expression level in one cell but it is not detected in another cell of the same type. Analyses of scRNA-seq data, including dimensionality reduction, clustering, differential expression (DE), and pseudotime analysis have shown that effective imputations for dropout events improve downstream analyses and assist biological interpretations (Lönnstedt and Speed, 2002; Love et al., 2014; Gong et al., 2018).

To date, several notable imputation methods have been proposed: scImpute (Li and Li, 2018), DrImpute (Gong et al., 2018), MAGIC (Van Dijk et al., 2018), SAVER (Huang et al., 2018), VIPER (Chen and Zhou, 2018), PBLR (Zhang and Zhang, 2021), netNMF-sc (Elyanow et al., 2020), and SCRABBLE (Peng et al., 2019). scImpute first performs clustering to identify cell subpopulations and further identifies dropout events through a Gamma-Normal mixture model, finally imputing dropout events by a non-negative least-squares regression (Li and Li, 2018). DrImpute optimizes the step of identifying cell subpopulations to impute dropout events by averaging the imputation from multiple clustering results (Gong et al., 2018). MAGIC builds a Markov affinity-based graph for imputation relying on cell-to-cell interactions (Van Dijk et al., 2018). SAVER uses a Bayesian-based model by various prior probability, and alters all gene expression values (Huang et al., 2018). VIPER imputes dropout events relying on local neighborhood cells via non-negative sparse regression models (Chen and Zhou, 2018). PBLR first separates the expression matrix into low-rank matrices and then applies an efficient alternating direction method of multi-pliers algorithm for imputation (Zhang and Zhang, 2021). netNMF-sc implements a network-regularized non-negative matrix factorization and leverages gene-gene interaction to accomplish imputation (Elyanow et al., 2020). SCRABBLE has been recently introduced to impute dropout events by adopting bulk RNA-seq data (Peng et al., 2019). Even though a lot of efforts have been taken into analyzing and imputing real dropout events, imputation of dropout events is still a difficult problem because of the high dropout rate and complex cellular heterogeneities for different scRNA-seq datasets. A recent study performed a systematic benchmark comparison and evaluation of 18 state-of-the-art scRNA-seq imputation methods by dividing them into three categories: (1) model-based imputation methods, (2) smooth-based imputation methods, and (3) data reconstruction methods (Hou et al., 2020). The tools we mentioned above, such as scImpute, SAVER, and VIPER, are exemplars of model-based imputation methods, MAGIC and DrImpute are exemplars of smooth-based imputation methods, and PBLR represents the data reconstruction method by using a low-rank matrix-based approach. The benchmark study observed that most imputation methods were most effective for providing a point estimate of the activity of individual genes; however, they were less effective in recovering cell-to-cell relationships resulting in less improvement in cell clustering and trajectory analysis. Thus, it is important to design new imputation methods to further improve the analysis of cell-to-cell relationships or DE that takes into account cell variability.

Relying on low-rank matrix completion to impute missing values is a long-standing question and has been investigated in biological sciences, including gene expression prediction, miRNA disease, protein-protein interaction (Simm et al., 2017), etc. Even though similar mathematical models could be applied to different biological problems to solve the matrix completion problem in scRNA-seq (recovering the dropout events), it is crucial to take the features of scRNA-seq into consideration. Most of existing scRNA-seq imputation methods have shown that it is advantageous for imputation to borrow and leverage information from similar cells. In recent years, researchers also start to integrate additional gene- or cell-related information (e.g., gene-gene interactions for netNMF-sc, bulk data for SCRABBLE) to assist imputation, which is important in matrix completion problem.

In this study, we present Bfimpute, a powerful imputation tool for scRNA-seq data that recovers dropout events by factorizing the count matrix into the product of gene-specific and cell-specific feature matrices (Mnih and Salakhutdinov, 2008; Salakhutdinov and Mnih, 2008). Bfimpute uses full Bayesian inference to describe the latent information for genes and cells and carries out a Markov chain Monte Carlo scheme that is able to easily incorporate any gene- or cell-related information to train the model and perform the imputation (Simm et al., 2017) (Figure 1). Deviating from common matrix-factorization-based methods, Bfimpute performs clustering or uses given cell type labels to group cells to leverage the information from similar cells more accurately and also make different cell types more distinguishable. Bfimpute extracts the information from each cell based on its gene expression to construct a cell-type-specific latent matrix for each cell group. Cells within each cell group are more likely to have similar latent vectors than cells from different cell groups, therefore enhancing cell-to-cell relationships. We demonstrate that Bfimpute performs better than the eight other notable published imputation methods mentioned above (scImpute, SAVER, VIPER, DrImpute, MAGIC, PBLR, netNMF-sc, and SCRABBLE) and two other matrix-fatorization-based methods (mcImpute [Mongia et al., 2019], ALRA [Linderman et al., 2018]) in both simulated and real scRNA-seq datasets on improving clustering, data visualization, differential gene expression analysis, and recovering gene expression temporal dynamics (pseudotime inference analysis) (Cannoodt et al., 2016; Ji and Ji, 2016; Trapnell et al., 2014).

RESULTS

Bfimpute improves both visualization and cell type identification

Principal-component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) (Van der Maaten and Hinton, 2008; McCarthy et al., 2017) are two popular dimensionality reduction techniques often used to visualize high-dimensional scRNA-seq datasets. Since dropout values are unknown in real datasets, we first tested accuracy of all different imputation methods using a simulated dataset where the ground truth was known. We applied the Splatter (Zappia et al., 2017) package to generate simulated datasets, which captured many features observed in the scRNA-seq data, including zero-inflation, gene-wise dispersion, and differing sequencing depths between cells. To test the strength and robustness of different imputation methods, we simulated a wide range of datasets to include 5, 6, 7, and 8 different cell types (see STAR Methods). Bfimpute achieved the most compact and well-separated clusters on the simulation, followed by scImpute and DrImpute (Figure 2). For all different cell type simulations, we also evaluated the clustering performances by evaluation metrics, where Bfimpute achieved the best scores for adjusted Rand index,

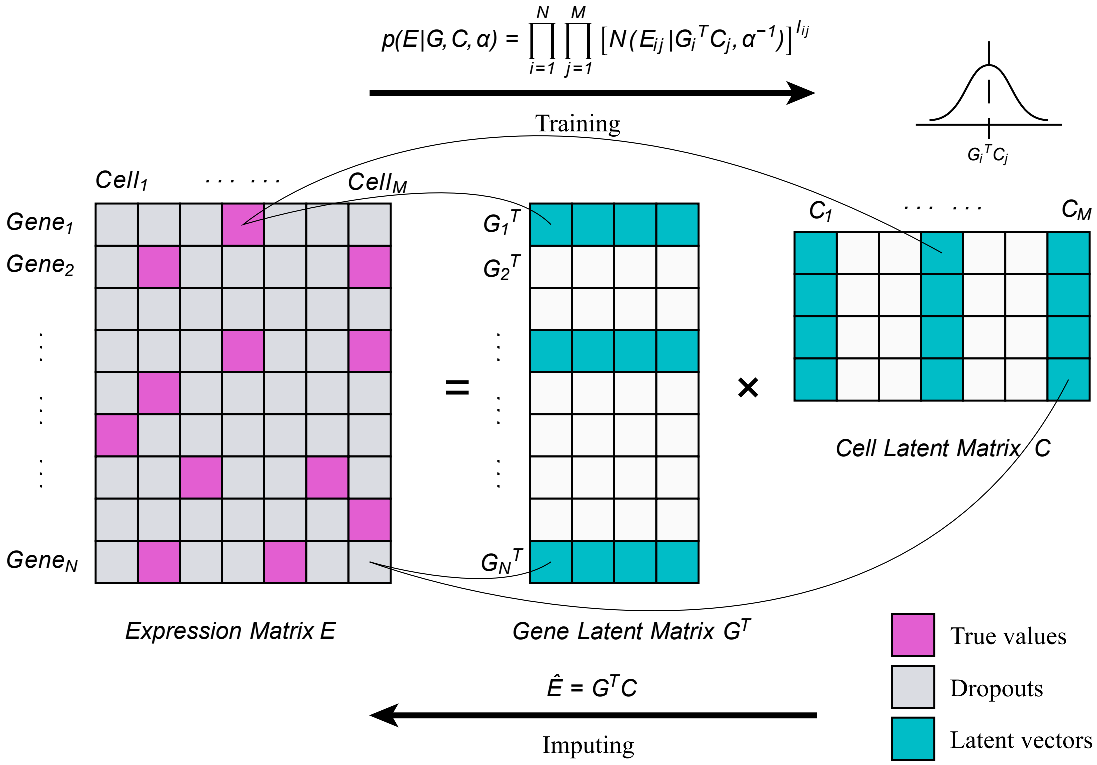


Figure 1. A brief illustration blueprinting the architecture of the Bfimpute method

In each group, Bfimpute borrows information from true values and factorizes the expression matrix into two latent matrices using MCMC. After training, Bfimpute imputes dropouts by performing product of the latent matrices. The details are shown in STAR Methods.

Jaccard index, normalized mutual information, and purity score compared with the raw data and five other imputation methods (see STAR Methods).

We further used two real datasets for this analysis. The first two principal components (PCs) from PCA were plotted to compare every dataset across seven different conditions: the raw dataset and six imputed ones through the Bfimpute, scImpute, SAVER, VIPER, DrImpute, and MAGIC methods. We first applied all imputation methods to a real scRNA-seq dataset from a human embryonic stem cell (ESC) differentiation study (Chu et al., 2016) to demonstrate the capacity of Bfimpute for improving the performance of data visualization. The dataset contains 1,018 single cells from 7 cell groups: neuronal progenitor cells (NPCs), definitive endoderm cell (DEC), endothelial cells (ECs), and trophoblast-like cells (TBs) are progenitors differentiated from H1 human ESCs. H9 human ESCs and human foreskin fibroblasts (HFFs) were used as controls cells. The raw dataset (i.e., without imputation) clearly identified the cluster of HFF cells; however, five other cell types were clustered very closely. After imputation by Bfimpute, the homogeneous subpopulations of H1 and H9 human ESCs were observed to substantially overlap and were well separated from the rest of the progenitors. The DECs, ECs, HFFs, NPCs, and TBs were also compactly clustered and well separated on the PCA plot (Figure 3A). Compared with the raw dataset, SAVER, VIPER, and DrImpute had no significant improvement for cell group identification. scImpute was the second best and generated similar compact cell groups as Bfimpute. We then compared clustering results of the spectral clustering algorithms (John et al., 2020) on the first several PCs to demonstrate the capability of Bfimpute to improve clustering accuracy in cell type identifications. For the true labels, we had seven cell types for this dataset, and we evaluated the clustering results by four different metrics: adjusted Rand index, Jaccard index, normalized mutual information (nmi), and purity (see STAR Methods). All four metrics suggested that Bfimpute achieved the best clustering accuracy compared with the raw data and the other five imputation methods (Figure 3B). We also show the comparison of the visualization performance through t-SNE. t-SNE on the raw dataset can better identify the seven cell types comparing to PCA. Bfimpute, DrImpute, and SAVER can further separate different cell groups and improve the visualization; however, the other four imputation methods demonstrated worse t-SNE results than raw data (Figure S1A).

To illustrate the recovering of dropouts in individual cells by imputation, we calculated the Pearson correlation from $\log_{10}$-transformed read counts between every pair of cells in the same type and from different cell types. This result indicated that imputation did recover the zero counts in every cell and the Pearson correlation increased from 0.70 to 0.87 for Bfimpute, 0.85 for scImpute, 0.72 for SAVER, 0.73 for VIPER, 0.78 for DrImpute, and 0.97 for MAGIC (Figure 3C, blue bars). One scatter plot of correlations between two randomly selected stem cells of the same cell type was demonstrated in Figure S1B. As we

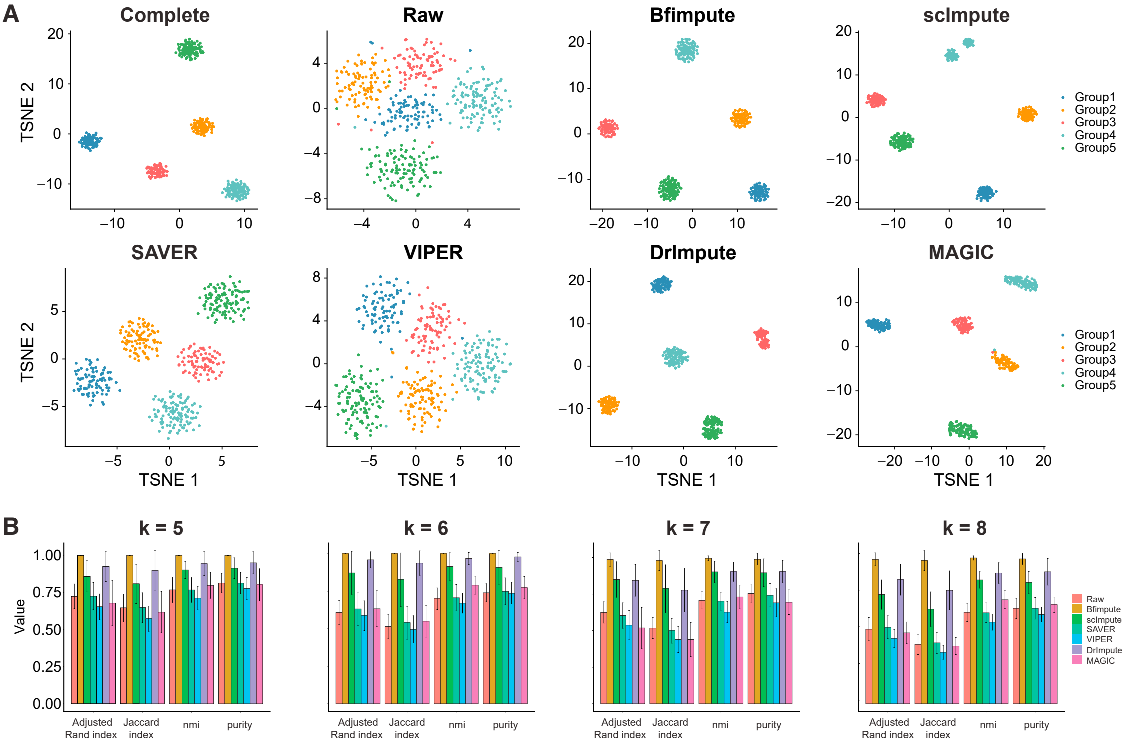


Figure 2. Bfimpute recovers dropout values and improves cell type identification in the simulated data

(A) The scatter plots show the t-SNE results calculated from the complete data, the raw data, and the imputed data by Bfimpute, scImpute, SAVER, VIPER, DrImpute, and MAGIC.

(B) k represents the number of cell clusters in simulated data. The adjusted Rand index, Jaccard index, nmi, and purity scores of clustering results are based on the raw and imputed data. Clustering is performed by the Spectrum package on the first two PCs as the PCA plot. Data are represented as mean ± STD among 10 different seeds.

expected, imputation methods usually increased the Pearson correlation between any two cells in the same cell type. Imputation should not increase the correlation between cells in different cell types by disregarding the biological variation between them. Among all imputation methods, MAGIC achieved the highest correlation in the same cell type, but the correlation between different cell types was also the highest (Figure 3C, red bars). Bfimpute demonstrated the best balance by maximizing the difference between correlation for the same over different cell types, which proves its ability to improve the analysis of cell-to-cell relationships.

Since Bfimpute performs clustering prior to imputation if the cell type information is unavailable, it is necessary to test whether the final clustering results are improved following imputation compared with the initial clustering. The scatter plots of the first two PCs from PCA results in the final clustering were more consistent with the true labels than the initial clustering results (Figures S1C–S1F). Especially, group 1 and group 2 in Figure S1F are separated well and show better consistency with DEC and EC in Figure S1E compared with corresponding groups in Figures S1D and C. In addition, we evaluated the initial and final clustering results relative to the true cell labels by the four types of evaluation metrics. The results confirmed that imputation with Bfimpute improved the performance of clustering (Figure S1G).

In addition to imputation, the latent gene matrix for each cell type generated by Bfimpute can facilitate the understanding of gene-gene relationships and cell functions. We launched a simple attempt to analyze gene-gene interaction networks and perform gene ontology (GO) enrichment analysis based on the gene latent matrix of NPCs from the Chu dataset with the assistance of WGCNA (Langfelder and Horvath, 2008) and clusterProfiler (Yu et al., 2012) (Figure S2; see details in STAR Methods). We identified one cluster (module) of highly correlated genes that were most involved in synaptic membrane, cation channel complex, and several other cellular components, which confirmed the functionality of NPCs.

We further investigated Bfimpute's performance of visualization and cell type identification on a zebrafish (Tang et al., 2017) scRNA-seq dataset. This dataset contains 246 single cells from 6 cell groups, and hematopoietic stem and progenitor cells (HSPCs) and HSPCs/thrombocytes among them come from one defined cell type with expected heterogeneity. After the quality control step, this Tang dataset was still sparse with zeros composing over 87.5% of the total counts. The comparison of visualization performance via PCA on the raw and six imputed datasets is shown in Figure S3. The raw dataset only coarsely identified the cluster for neutrophil cells, whereas cells from other cell types were mixed and spread wildly. After imputation by Bfimpute, four distinct immune cell subpopulations can be

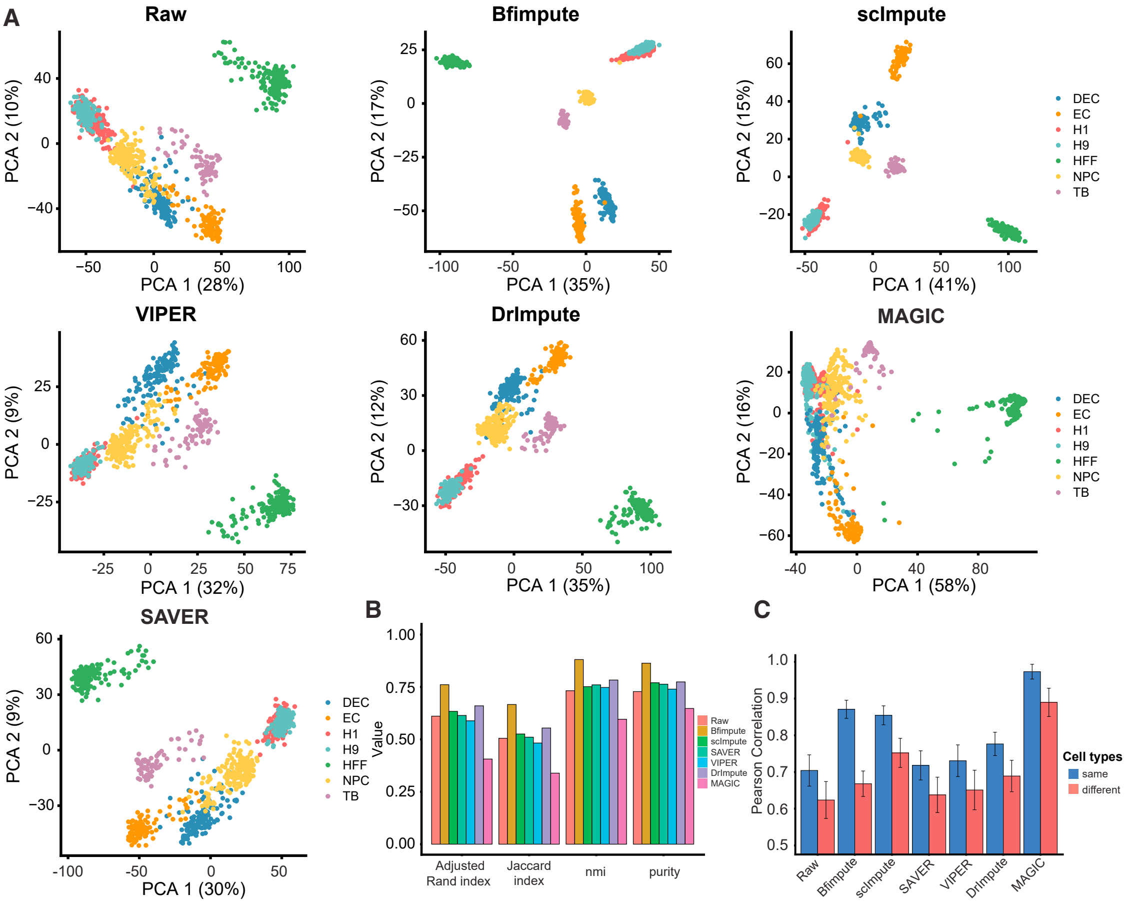


Figure 3. Bfimpute improves PCA visualization and cell type identification

(A) The first two PCs calculated from the raw data, and the imputed data by Bfimpute, scImpute, VIPER, DrImpute, MAGIC, and SAVER.

(B) The adjusted Rand index, Jaccard index, nmi, and purity scores of clustering results based on the raw and imputed data.

(C) Average Pearson correlations between any two cells from same type and different type. Data are represented as mean ± SD.

identified for neutrophils, T cells, natural killer (NK) cells, and B cells, where the cluster members were much more compact compared with those of the raw dataset. Neutrophils, and T, NK, and B cells, were distantly positioned on the PCA plot. HSPCs and HSPCs/thrombocytes were from one defined cell type with expected heterogeneity; so, after Bfimpute's imputation, they were still spatially closer than other cells (Figure S3A). The raw data and the imputed data by other five imputation methods did not correctly identify the four immune cell subpopulations. Clustering accuracy results from the four metrics for Bfimpute were better than the other five imputation methods, and Bfimpute achieved a better correlation for the same cell type without losing variation between different cells types (Figures S3B and S3C).

Bfimpute improves DE and pseudotime analysis

DE analysis is widely used in bulk RNA-seq data. Performing DE analysis for scRNA-seq data to reveal the stochastic nature of gene expression in single cells is challenging since scRNA-seq data suffer from high dropout events. However, it has been proven that good imputation methods could lead to a better agreement between scRNA-seq and bulk RNA-seq data of the same biological condition on genes known to have little cell-to-cell heterogeneity. We utilized the Chu dataset with both bulk and scRNA-seq data available on human ESCs and DECs to compare Bfimpute with the raw dataset and the other five imputation methods for DE analysis (Wang et al., 2011, 2012). This dataset contained 6 samples of bulk RNA-seq (4 in H1 ESCs and 2 in DEC) and 350 samples of scRNA-seq (212 in H1 ESCs and 138 in DEC). The percentages of zero entries were 8.8% in bulk data and 44.9% in scRNA-seq data, respectively. We first performed DE analysis in the bulk data and identified the top 200 DE genes by DESeq2 (Love et al., 2014). We then plotted the expression profiles of these scRNA-seq data for 7 conditions: raw dataset, Bfimpute, scImpute, SAVER, VIPER, DrImpute, and MAGIC. We found the expression profiles of these top 200 genes after Bfimpute's imputation demonstrated better concordance with those in bulk data (Figure 4A). To further evaluate whether imputation improves DE analysis in scRNA-seq data, we first used DESeq2 to identify DE genes for raw scRNA-seq dataset and scRNA-seq datasets after six different imputations. We then generated different lists of DE genes for the bulk data by applying

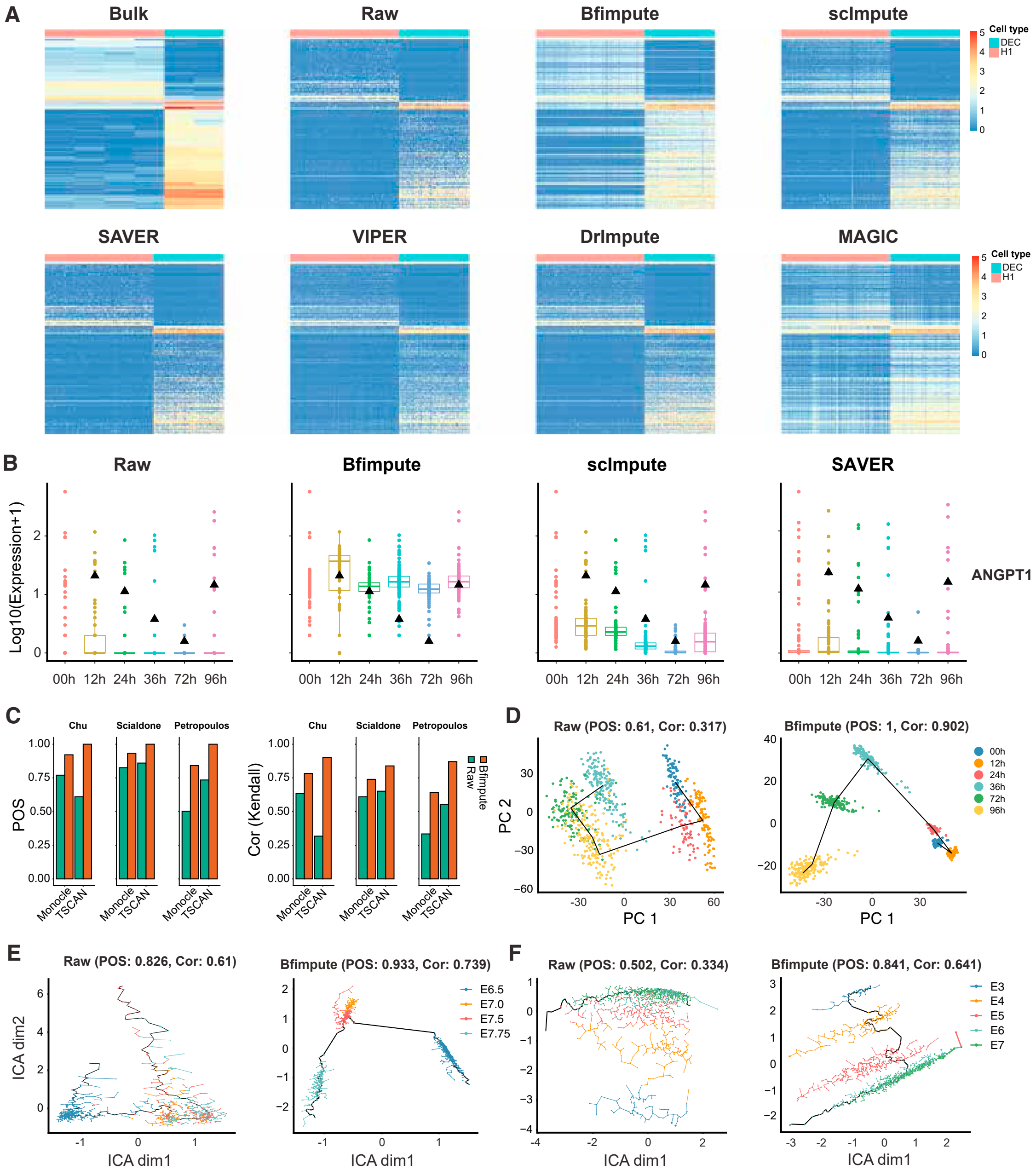


Figure 4. Bfimpute improves DE and pseudotime analysis

(A) The expression profiles of the top 200 DE genes detected in the bulk data by DESeq2 for 7 conditions: raw dataset, Bfimpute, scImpute, SAVER, VIPER, DrImpute, and MAGIC.

(B) Time course expression patterns of the example gene ANGPT1 that is annotated with the GO term “endoderm development.” The small black triangles marks the average bulk data for each time point.

(C) The barplots of POS and Kendall’s rank correlation score between the true time labels and pseudotime ordering inferred by Monocle and TSCAN on the Chu, Scialdone, and Petropoulos datasets.

(D) Visualization of lineage reconstruction of the Chu dataset by TSCAN. The lines show the edges of the minimum spanning tree (MST) of each cluster of cells.

(E and F) Visualization of lineage reconstruction by Monocle of the Scialdone and Petropoulos datasets, respectively. The lines connecting every point indicate the edges of the MST constructed by Monocle. The solid black line represents the main diameter path of the MST and denotes the backbone of the pseudotime ordering.

Table 1. The barplots of POS and Kendall's rank correlation score between true time labels and pseudotime ordering inferred by Monocle and TSCAN on the Chu, Scialdone, and Petropoulos datasets without or with imputation

		Monocle				TSCAN			
Datasets	Scores	Raw	Bfimpute	scImpute	DrImpute	Raw	Bfimpute	scImpute	DrImpute
Chu et al. (2016)	POS	0.770	0.921	0.907	0.883	0.610	1.000	0.994	0.989
	Kendall	0.633	0.783	0.752	0.721	0.317	0.902	0.890	0.880
Scialdone et al. (2016)	POS	0.826	0.933	0.879	0.829	0.860	1.000	0.910	0.917
	Kendall	0.610	0.739	0.670	0.611	0.651	0.839	0.708	0.717
Petropoulos et al. (2016)	POS	0.502	0.841	0.959	0.602	0.734	1.000	0.992	0.805
	Kendall	0.334	0.641	0.808	0.422	0.554	0.871	0.852	0.625

different thresholds for false-discovery rates of genes. Finally, for every threshold, we compared the DE genes for the bulk data and scRNA-seq data of those seven different conditions and calculated the AUC values for each condition. The AUC values suggested that all imputation methods improved DE analysis. Bfimpute generated DE genes most consistent with the bulk data (AUC values raw, 0.568; Bfimpute, 0.670; scImpute, 0.665; SAVER, 0.624; VIPER, 0.639; DrImpute, 0.657; and MAGIC, 0.668).

If bulk data for the same biological condition were provided, they could be used as a gold standard to compare the average gene expression level with the scRNA-seq data, even though the scRNA-seq data presented more cell-to-cell variation. We expected that average gene expression level in the scRNA-seq data was highly correlated with bulk RNA-seq data. To investigate this, we plotted correlations between gene expression in single-cell and bulk data and found that all imputation methods did improve the correlation between bulk and scRNA-seq data, and that Bfimpute, MAGIC, scImpute had the best improvement (Figure S1H). We further selected several genes (e.g., ANGPT1,GDF3, BMP4, EPB41L5) of DECs from different time points to plot their average gene expression levels in both bulk and scRNA-seq data. These genes were annotated with the GO term ‘‘endoderm development,’’ and they were likely to be affected by dropout events (Gong et al., 2018; Blake et al., 2017). Imputed read counts for these genes by Bfimpute showed higher gene expression correlation and better consistency with the bulk data (Figures 4B and S4A–S4D).

In addition to the DE analysis, we also used three datasets across developmental stages: time course scRNA-seq data (Chu et al., 2016) from the same study; stages of mouse mesodermal development (Scialdone et al., 2016); and stages of human preimplantation development (Petropoulos et al., 2016). The results showed that Bfimpute improved gene expression temporal dynamics through pseudotime inference analysis. The Chu dataset comprises a total of 758 single cells captured and profiled by scRNA-seq at 0, 12, 24, 36, 72, and 96 h of differentiation. The Scialdone dataset contains 1,205 cells from four stages at embryonic day 6.5 (E6.5), E7.0, E7.5, and E7.75. The Petropoulos dataset includes 1,529 cells from 5 stages, from developmental day E3 to E7. We first applied Bfimpute to the raw scRNA-seq data with true cell type labels, and then examined how the time course expression patterns changed in the imputed data using TSCAN (Ji and Ji, 2016) and Monocle (Trapnell et al., 2014; Qiu et al., 2017), which were designed to infer pseudotime from the biological process. We then calculated pseudo-temporal ordering score (POS) and Kendall's rank correlation score to evaluate the consistency between the true time labels and pseudotime ordering inferred by TSCAN and Monocle.

Both POS and Kendall's rank correlation score increased after imputation by Bfimpute compared with the raw data and other imputation methods (Figure 4C; Table 1). Figure 4D depicts the PCA plot of human ESCs from the Chu dataset and the time course in 2D space constructed using PCA without (left panel) or with (right panel) imputation by Bfimpute. The trajectory in this subplot was constructed using TSCAN. Without imputation, this trajectory started from 0 h and ended at 36 h, while, with imputation, the pseudotime trajectory fitted the true time labels perfectly from 0 h (blue) to 96 h (yellow). Both POS and Kendall's rank correlation score increased significantly (POS increased from 0.61 to 1, and Kendall's rank correlation increased from 0.317 to 0.902). In Figures 4E and 4F, depicting the Scialdone and Petropoulos datasets, respectively, scRNA-seq results were plotted in 2D space using independent component analysis and pseudotime trajectories were constructed by Monocle. The results demonstrated that POS and Kendall's rank correlation scores in both datasets increased further. Besides, in Figure 4E, the backbone pseudotime trajectory from E6.5 (blue) to E7.75 (cyan) constructed from data imputed by Bfimpute clearly showed the time course, and with minimal overlap between stages, unlike the raw data. In Figure 4F, prior to imputation, E3 and E4 stages were not reached at all by the backbone trajectory. In contrast, after imputation using Bfimpute, the backbone pseudotime trajectory started from E3, traversed through E4, E5, and terminated at E6 and E7.

In summary, all these results demonstrated that imputation using Bfimpute was able to enhance downstream analysis. Especially, Bfimpute significantly improved the performance of DE analysis using DESeq2 and pseudotime inference using Monocle and TSCAN. In the next section, we discuss imputation with the aid of cell type labels in more detail.

Bfimpute improves performance with the aid of additional experimental information

Imputation methods, including Bfimpute, scImpute, DrImpute, and PBLR, all first identify similar cells based on clustering, and imputation is then performed by leveraging the expression values from similar cells. Being able to first identify the appropriate cell groups enhanced the ability of imputing the dropout events. A substantial number of scRNA-seq studies have

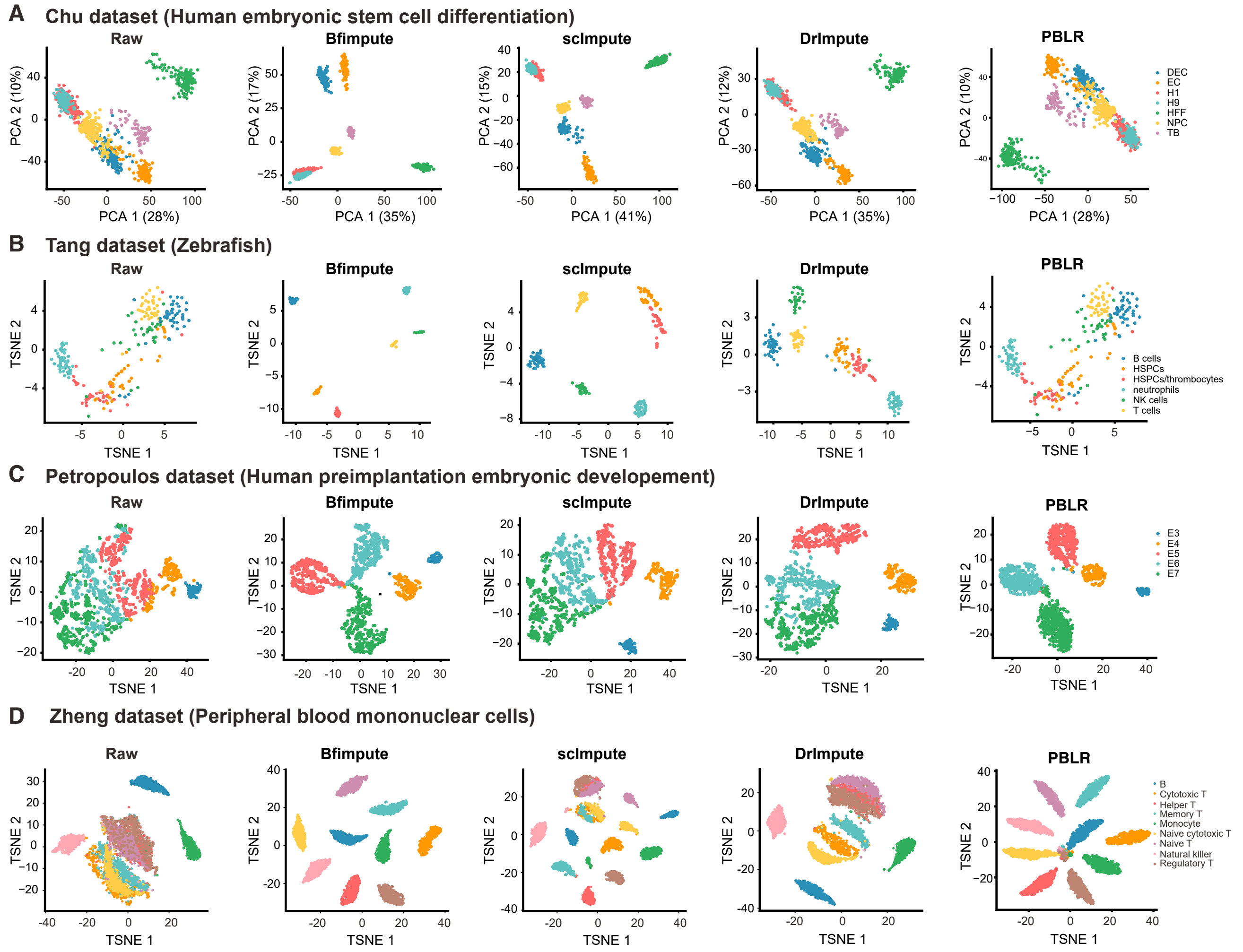


Figure 5. Bfimpute with labels improves PCA and t-SNE visualizations and cell type identification

(A) The first two PCs calculated from the raw data, and the imputed data by Bfimpute, scImpute, DrImpute, and PBLR for the Chu dataset (human embryonic stem cell differentiation study).

(B) The t-SNE results from the raw data, and the imputed data by Bfimpute, scImpute, DrImpute, and PBLR for the Tang dataset (zebrafish data).

(C) The t-SNE results from the raw data, and the imputed data by Bfimpute, scImpute, DrImpute, and PBLR for the Petropoulos dataset (human preimplantation embryonic development data).

(D) The t-SNE results from the raw data, and the imputed data by Bfimpute, scImpute, DrImpute, and PBLR for Zheng dataset (PBMCs).

identified cell types from experimental design or marker genes. We applied Bfimpute, scImpute, DrImpute, and PBLR to the raw scRNA-seq data with true cell type labels on five datasets in total.

We again investigated the PCA and t-SNE visualizations for identification of cell subpopulations. Our results showed that Bfimpute outperformed the other three methods and clearly differentiated almost every cell group in different datasets. For the Chu dataset, Bfimpute further correctly identified three outlier cells into correct groups compared with the previous imputation without cell labels (see Figure 5A versus Figure 3A: one EC [orange point], one DEC [blue point], and one NPC [yellow point] cell were brought back to the corresponding EC, DEC, and NPC cell groups, respectively). H9 cells were also further apart from H1 cells in the vertical dimension. For the Tang dataset, even the most mixed B, NK, and T cells (blue, green, and yellow colors) from the raw dataset were separated from each other after Bfimpute's imputation, and HSPCs and HSPCs/thrombocytes cells were spatially close, but split into two cell groups (Figures 5B and S5A). For the Petropoulos dataset, whose cells are distinguished by development stages other than the cell type, the five different stages were clearly distinguished from each other after Bfimpute's imputation (Figures 5C and S5B).

We also applied four imputation methods to a large 10x dataset generated by the high-throughput droplet-based system. To generate this dataset, we randomly selected 500 cells from 9 immune cell types, so that it contained a total of 4,500 peripheral

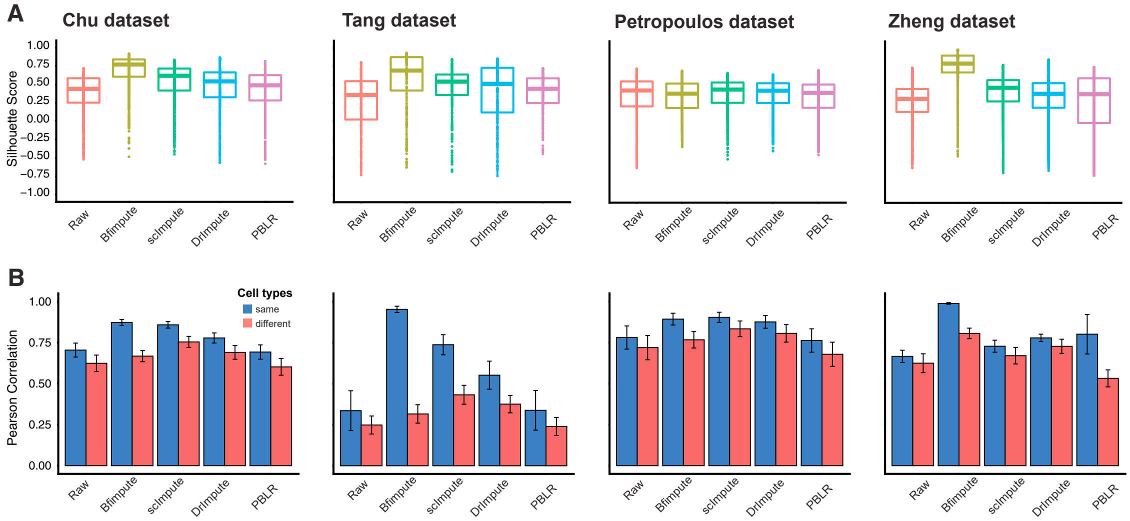


Figure 6. Bfimpute with labels further improves silhouette scores, cell-cell relationships, and clustering accuracy

(A) The boxplots of silhouette scores among all cells on the Chu dataset (human embryonic stem cell differentiation study), Tang dataset (zebrafish data), Petropoulos dataset (human preimplantation embryonic development data), and Zheng dataset (PBMCs).

(B) Average Pearson correlations between any two cells from same type and different type on the above-mentioned four datasets. Data are represented as mean ± SD.

blood mononuclear cells (Li and Li, 2018; Zheng et al., 2017). In the raw data, 98.3% read counts are exactly zeros. Our PCA and t-SNE results indicated that Bfimpute's imputation identified nine immune cell types from raw data (Figure 5D).

For all datasets mentioned above, we further estimated the silhouette score to evaluate the quality of the spectral clustering results after imputation based on how well its data points are clustered (Table S1; Figure 6A). Bfimpute achieved the best score in all but one set. We also calculated the Pearson correlation coefficient between every pair of cells from the same cell type and from different cell types (Figure 6B). Bfimpute was confirmed to be the best in all imputation methods.

To compare Bfimpute to several related methods using matrix factorization, we additionally ran ALRA and mcImpute on these four datasets. Some of them achieved above-average results but none of them performed better than Bfimpute (Figures S5C–S5J).

Due to the importance of scalability for imputation in big datasets, we further applied Bfimpute on an extremely large 10x-based dataset containing cells from multiple lineages (Vladoiu et al., 2019). The Vladoiu dataset contains 62,893 mouse cerebellum cells collected at 9 time points (E10, E12, E14, E16, E18, and postnatal day 0 [P0], P5, P7, and P14) with 91.9% dropout rate. As shown in Figure S6A, Bfimpute significantly improved t-SNE visualization. For example, cells at the P5 stage were better separated from cells at the P7 stage, and cells at the E14 and E16 stages were revealed. We also applied scImpute, DrImpute, SAVER, and PBLR with cell labels, and MAGIC and VIPER, on this dataset. After imputation, we used TSCAN to infer pseudotime. Bfimpute showed the best performance on both POS and Kendall's rank correlation score (Figure S6B). We further evaluated clustering performance using four types of evaluation metrics and all of them suggested that Bfimpute achieved the best clustering accuracy (Figure S6C). The computation time cost was ordered in the following ascending order: MAGIC, PBLR, Bfimpute, scImpute, and SAVER; DrImpute and VIPER failed to complete (Figure S6D).

SCRABBLE is an approach introduced recently integrating bulk data as a constraint to impute dropout events in scRNA-seq data. netNMF-sc is another tool to leverage gene-gene interactions for imputation. Since Bfimpute can easily adopt bulk data as additional information into the gene latent matrix, we also tested if bulk data can further improve performance. In the scRNA-seq dataset of human ESCs with bulk data, we did not observe significant differences with or without bulk data as additional information (Figures S1I and S1J versus Figure 5A). The reason could be that similar gene level information has less effect than similar cell level information for the imputation of dropout events. We also found that the performances of SCRABBLE and netNMF-sc after integrating accurate estimate of the gene expression distributions and gene-gene interaction information with bulk data, were not better than Bfimpute (Figures S1I and S1J).

DISCUSSION

scRNA-seq has become an indispensable tool in recent years, as it has made it possible to study genome-wide transcriptomes in single-cell resolution. Unfortunately, the large proportion of dropout events in scRNA-seq data limits its efficacy. Recently, there has been a debate about whether scRNA-seq datasets with unique molecular identifiers are zero-inflated. Some researchers have modeled droplet-based scRNA-seq data reasonably well without assuming that zero values are artificial (Svensson, 2020; Silverman et al., 2020). However, a recent study designing scRNA-seq imputation models without zero-inflation suggests that imputation is necessary and able to facilitate downstream analysis (Tang et al., 2020), because missing values are unavoidable at present due to sequencing technical limitations. The main point we can glean from the discussion regarding the types of zeros in this debate is the importance of discriminating the difference between zero values due to technical and biological factors (Silverman et al., 2020). In this study, we introduce Bfimpute, which applies a mixture model to identify zeros due to technical factors within each cell group after clustering or using cell type labels, and performs Bayesian factorization to recover dropout events in scRNA-seq data.

In this study, we thoroughly tested Bfimpute on both simulated and real datasets by comparison with ten other tools. By utilizing a Bayesian matrix factorization method to recover zeros due to technical factors in each cell cluster, we have shown that Bfimpute is most effective in recovering cell-to-cell relationships that further improve downstream analyses, including identification of cell subpopulations, differential expressed genes, and trajectory analysis. Bfimpute can be easily incorporated into the existing downstream pipelines and analyses.

Bfimpute is an example of low-rank matrix-based imputation methods. Several scRNA-seq imputation methods based on matrix factorization, such as mcImpute and ALRA, have been proposed recently. However, almost all of these methods are using majorization-minimization algorithms to find a single point estimate of the parameters, which are prone to overfitting. Bfimpute uses a fully Bayesian probabilistic matrix factorization by substituting hyperparameters with hyperpriors and performing Gibbs sampling for the approximate inference. The advantage of this Bayesian model is that it provides a predictive distribution instead of just a single number during recovering each dropout event to avoid overfitting, and the confidence in the prediction can be quantified and considered into the model. The use of a full Bayesian model proved to be a considerable advantage for Bfimpute to outperform other imputation methods. For the time complexity of the full Bayesian-based model, the most time-consuming aspect of training Bfimpute is the inversion of the $D \times D$ matrices for latent and feature vectors, which are $O(D^3)$ operations. In this case, Bfimpute is not as significantly affected by the size of the count matrix as the other methods. We have used $D = 32$ (by default) and also tested $D = 16$ for all the experiments and there were no significant differences for smaller D in downstream analyses.

Bfimpute imputes two latent cell and gene matrices for each cell group through a Gibbs sampling process, and reaches a stationary state to generate the final cell-gene expression matrix, in which the dropout events will be recovered. Another advantage of Bfimpute is being able to integrate any gene- or cell-related information of scRNA-seq data into these two latent gene and cell matrices to impute missing values. Information from both similar cells or/and bulk data can be easily integrated into our model. Even though some other methods have a similar functionality in this respect, which allows them to impute dropout events with the aid of number of cell types or cell labels, they fail to achieve as good performance as Bfimpute for most of scRNA-seq data that we tested. Any resource provided by the users from the cell level and gene level could be used as additional information to improve dropout events imputation in scRNA-seq data in the future.

Limitations of the study

As more imputation methods become available, a systematic evaluation is necessary. Two recent benchmarking studies demonstrated that the performance of imputation methods varied across evaluation criteria, experimental protocols, datasets, and downstream analyses (Hou et al., 2020; Zhang and Zhang, 2020). There is no consensus about which imputation method is the best. Although we benchmarked several simulated and real datasets and Bfimpute outperformed ten other notable imputation methods, it is premature to conclude that Bfimpute will achieve better performance across all datasets and improve downstream analysis. Imputation methods such as Bfimpute may also increase false positives for downstream analysis. Depending on the application, users may constraint parameters to maximize true positives or minimize false positives.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - ○ Lead contact
  - ○ Materials availability
  - ○ Data and code availability
- METHOD DETAILS
  - ○ Cell clustering and dropout detection
  - ○ Probabilistic model for scRNA-seq expression matrix imputation
  - ○ Gibbs sampler to impute dropout events
  - ○ Generation of simulated data
  - ○ Quality control for real datasets
  - ○ Evaluation metrics of clustering results
  - ○ Measurement of pseudotime ordering
  - ○ Gene-gene interaction and gene ontology enrichment analysis on gene latent matrix
- QUANTIFICATION AND STATISTICAL ANALYSIS

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.crmeth.2021.100133.

ACKNOWLEDGMENTS

This work was supported by Vanderbilt University Development Funds (FF_300033). L.Z. is partially supported by Research Grant Council Early Career Scheme (HKBU 22201419).

AUTHOR CONTRIBUTIONS

X.Z. conceived and led this work. W.S. provided guidance and criticism on this work. Z.-H.W. and X.Z. designed the model. Z.-H.W. implemented the Bfimpute software and performed data analysis. Z.-H.W. and X.Z. wrote the manuscript. W.S., J.L.L., and L.Z. reviewed and edited the manuscript. All authors proofread and approved the final manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: June 7, 2021
Revised: October 11, 2021
Accepted: November 24, 2021
Published: December 20, 2021

REFERENCES

Blake, J.A., Eppig, J.T., Kadin, J.A., Richardson, J.E., Smith, C.L., and Bult, C.J.; the Mouse Genome Database Group (2017). Mouse genome database (MGD)-2017: community knowledge resource for the laboratory mouse. Nucleic Acids Res. *45*, D723–D729.

Cannoodt, R., Saelens, W., and Saeys, Y. (2016). Computational methods for trajectory inference from single-cell transcriptomics. Eur. J. Immunol *46*, 2496–2506.

Chen, M., and Zhou, X. (2018). VIPER: variability-preserving imputation for accurate gene expression recovery in single-cell RNA sequencing studies. Genome Biol. *19*, 196.

Chu, L.-F., Leng, N., Zhang, J., Hou, Z., Mamott, D., Vereide, D.T., Choi, J., Kendziorski, C., Stewart, R., and Thomson, J.A. (2016). Single-cell RNA-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. Genome Biol. *17*, 173.

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.

Gong, W., Kwak, I.Y., Pota, P., Koyano-Nakagawa, N., and Garry, D.J. (2018). DrImpute: imputing dropout events in single cell RNA sequencing data. BMC Bioinformatics *19*, 220.

Hou, W., Ji, Z., Ji, H., and Hicks, S.C. (2020). A systematic evaluation of single-cell RNA-sequencing imputation methods. Genome Biol. *21*, 218.

Huang, M., Wang, J., Torre, E., Dueck, H., Shaffer, S., Bonasio, R., Murray, J.I., Raj, A., Li, M., and Zhang, N.R. (2018). SAVER: gene expression recovery for single-cell RNA sequencing. Nat. Methods *15*, 539–542.

Jaccard, P. (1912). The distribution of the flora in the alpine zone. 1. New Phytol *11*, 37–50.

Ji, Z., and Ji, H. (2016). TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. Nucleic Acids Res. *44*, e117.

John, C.R., Watson, D., Barnes, M.R., Pitzalis, C., and Lewis, M.J. (2020). Spectrum: fast density-aware spectral clustering for single and multi-omic data. Bioinformatics *36*, 1159–1166.

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004). Kernlab – an S4 package for kernel methods in R. J. Stat. Softw *11*, 1–20.

Kharchenko, P.V., Silberstein, L., and Scadden, D.T. (2014). Bayesian approach to single-cell differential expression analysis. Nat. Methods *11*, 740–742.

Langfelder, P., and Horvath, S. (2008). WGCNA: an R package for weighted correlation network analysis. BMC Bioinformatics *9*, 559.

Li, W.V., and Li, J.J. (2018). An accurate and robust imputation method scImpute for single-cell RNA-seq data. Nat. Commun *9*, 997.

Linderman, G.C., Zhao, J., and Kluger, Y. (2018). Zero-preserving imputation of scRNA-seq data using low-rank approximation. bioRxiv. 10.1101/397588.

Lönnstedt, I., and Speed, T. (2002). Replicated microarray data. Stat. Sin *12*, 31–46.

Love, M.I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol. *15*, 550.

McCarthy, D.J., Campbell, K.R., Lun, A.T., and Wills, Q.F. (2017). Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. Bioinformatics *33*, 1179–1186.

Mnih, A., and Salakhutdinov, R.R. (2008). Probabilistic matrix factorization. In Advances in Neural Information Processing Systems, pp. 1257–1264.

Mongia, A., Sengupta, D., and Majumdar, A. (2019). Mcimpute: matrix completion based imputation for single cell RNA-seq data. Front. Genet. *10*, 9.

Morey, L.C., and Agresti, A. (1984). The measurement of classification agreement: an adjustment to the rand statistic for chance agreement. Educ. Psychol. Meas *44*, 33–37.

Ng, A.Y., Jordan, M.I., and Weiss, Y. (2001). On spectral clustering: analysis and an algorithm. Adv. Neural Inf. Process. Syst. *2*, 849–856.

Peng, T., Zhu, Q., Yin, P., and Tan, K. (2019). SCRABBLE: single-cell RNA-seq imputation constrained by bulk RNA-seq data. Genome Biol. *20*, 88.

Petropoulos, S., Edsgärd, D., Reinius, B., Deng, Q., Panula, S.P., Codeluppi, S., Plaza Reyes, A., Linnarsson, S., Sandberg, R., and Lanner, F. (2016). Single-cell RNA-seq reveals lineage and X chromosome dynamics in human preimplantation embryos. Cell *165*, 1012–1026.

Qiu, P. (2020). Embracing the dropouts in single-cell RNA-seq analysis. Nat. Commun *11*, 1169.

Qiu, X., Mao, Q., Tang, Y., Wang, L., Chawla, R., Pliner, H.A., and Trapnell, C. (2017). Reversed graph embedding resolves complex single-cell trajectories. Nat. Methods *14*, 979–982.

Rand, W.M. (1971). Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. *66*, 846–850.

Zheng, G.X.Y., Terry, J.M., Belgrader, P., Ryvkin, P., Bent, Z.W., Wilson, R., Ziraldo, S.B., Wheeler, T.D., McDermott, G.P., Zhu, J., et al. (2017). Massively parallel digital transcriptional profiling of single cells. Nat. Commun *8*, 14049.

Salakhutdinov, R., and Mnih, A. (2008). Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In Proceedings of the 25th International Conference on Machine Learning, pp. 880–887.

Scialdone, A., Tanaka, Y., Jawaid, W., Moignard, V., Wilson, N.K., Macaulay, I.C., Marioni, J.C., and Göttgens, B. (2016). Resolving early mesoderm diversification through single-cell expression profiling. Nature *535*, 289–293.

Silverman, J.D., Roche, K., Mukherjee, S., and David, L.A. (2020). Naught all zeros in sequence count data are the same. Comput. Struct. Biotechnol. J. *18*, 2789–2798.

Simm, J., Arany, A., Zakeri, P., Haber, T., Wegner, J.K., Chupakhin, V., Ceulemans, H., and Moreau, Y. (2017). Macau: scalable Bayesian factorization with high-dimensional side information using MCMC. In 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP) (IEEE), pp. 1–6.

Strehl, A., and Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. *3*, 583–617.

Svensson, V. (2020). Droplet scRNA-seq is not zero-inflated. Nat. Biotechnol *38*, 147–150.

Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B.B., Siddiqui, A., et al. (2009). mRNA-Seq whole-transcriptome analysis of a single cell. Nat. Methods *6*, 377–382.

Tang, Q., Iyer, S., Lobbardi, R., Moore, J.C., Chen, H., Lareau, C., Hebert, C., Shaw, M.L., Neftel, C., Suva, M.L., et al. (2017). Dissecting hematopoietic and renal cell heterogeneity in adult zebrafish at single-cell resolution using RNA sequencing. J. Exp. Med *214*, 2875–2887.

Tang, W., Bertaux, F., Thomas, P., Stefanelli, C., Saint, M., Marguerat, S., and Shahrezaei, V. (2020). bayNorm: Bayesian gene expression recovery, imputation and normalization for single-cell RNA-sequencing data. Bioinformatics *36*, 1174–1181.

Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N.J., Livak, K.J., Mikkelsen, T.S., and Rinn, J.L. (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nat. Biotechnol *32*, 381–386.

Van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-SNE. J. Machine Learn. Res. *9*, 2579–2605.

van Dijk, D., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A.J., Burdziak, C., Moon, K.R., Chaffer, C.L., Pattabiraman, D., et al. (2018). Recovering gene interactions from single-cell data using data diffusion. Cell *174*, 716–729.e27.

Vladoiu, M.C., El-Hamamy, I., Donovan, L.K., Farooq, H., Holgado, B.L., Sundaravadanam, Y., Ramaswamy, V., Hendrikse, L.D., Kumar, S., Mack, S.C., et al. (2019). Childhood cerebellar tumours mirror conserved fetal transcriptional programs. Nature *572*, 67–73.

Wagner, S., and Wagner, D. (2007). Comparing Clusterings: An Overview. Technical Report 2006-04, Faculty of Informatics

Wang, P., Rodriguez, R.T., Wang, J., Ghodasara, A., and Kim, S.K. (2011). Targeting SOX17 in human embryonic stem cells creates unique strategies for isolating and analyzing developing endoderm. Cell Stem Cell *8*, 335–346.

Wang, P., McKnight, K.D., Wong, D.J., Rodriguez, R.T., Sugiyama, T., Gu, X., Ghodasara, A., Qu, K., Chang, H.Y., and Kim, S.K. (2012). A molecular signature for purified definitive endoderm guides differentiation and isolation of endoderm from mouse and human embryonic stem cells. Stem Cells Dev. *21*, 2273–2287.

Yu, G., Wang, L.-G., Han, Y., and He, Q.-Y. (2012). clusterProfiler: an R package for comparing biological themes among gene clusters. OMICS *16*, 284–287.

Zappia, L., Phipson, B., and Oshlack, A. (2017). Splatter: simulation of single-cell RNA sequencing data. Genome Biol. *18*, 174.

Zhang, L., and Zhang, S. (2020). Comparison of computational methods for imputing single-cell RNA-sequencing data. IEEE/ACM Trans. Comput. Biol. Bioinformatics *17*, 376–389.

Zhang, L., and Zhang, S. (2021). Imputing single-cell RNA-seq data by considering cell heterogeneity and prior expression of dropouts. J. Mol. Cell Biol. *13*, 29–40.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Human embryonic stem cell differentiation	Chu et al. (2016)	GSE75748
Human embryonic stem cell differentiation (time course data)	Chu et al. (2016)	GSE75748
Zebrafish	Tang et al. (2017)	GSE100911
Stages of mouse mesodermal development	Scialdone et al. (2016)	GSE74994
Stages of human preimplantation development	Petropoulos et al. (2016)	ArrayExpress: E-MTAB-3929
Peripheral blood mononuclear cells	Zheng et al. (2017)	https://support.10xgenomics.com/single-cell-gene-expression/datasets
Mouse cerebellum cells	Vladoiu et al. (2019)	GSE118068
Software and algorithms		
Splatter	Zappia et al. (2017)	https://bioconductor.org/packages/splatter/
scater	McCarthy et al. (2017)	https://bioconductor.org/packages/scater/
kernlab	Karatzoglou et al. (2004)	https://CRAN.R-project.org/package=kernlab
Spectrum	John et al. (2020)	https://CRAN.R-project.org/package=Spectrum
DESeq2	Love et al. (2014)	https://bioconductor.org/packages/DESeq2/
Monocle	Trapnell et al. (2014)	https://github.com/cole-trapnell-lab/monocle-release
TSCAN	Ji and Ji (2016)	https://github.com/zji90/TSCAN
WGCNA	Langfelder and Horvath (2008)	https://CRAN.R-project.org/package=WGCNA
clusterProfiler	Yu et al. (2012)	https://bioconductor.org/packages/clusterProfiler/
TSCAN	Ji and Ji (2016)	https://github.com/zji90/TSCAN
scDatasets	Gong et al. (2018)	https://github.com/gongx030/scDatasets
scImpute	Li and Li (2018)	https://github.com/Vivianstats/scImpute
DrImpute	Gong et al. (2018)	https://github.com/gongx030/DrImpute
MAGIC	Van Dijk et al. (2018)	https://github.com/KrishnaswamyLab/MAGIC
SAVER	Huang et al. (2018)	https://github.com/mohuangx/SAVER
VIPER	Chen and Zhou (2018)	https://github.com/ChenMengjie/VIPER
PBLR	Zhang and Zhang (2021)	https://github.com/amsszlh/PBLR
netNMF-sc	Elyanow et al. (2020)	https://github.com/raphael-group/netNMF-sc
SCRABBLE	Peng et al. (2019)	https://github.com/software-github/SCRABBLE
Bfimpute	Zenodo	https://doi.org/10.5281/zenodo.5676122

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and code should be directed to and will be fulfilled by the lead contact, Prof. Xin Zhou (maizie.zhou@vanderbilt.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- All data reported in this paper will be shared by the lead contact upon request.
- All original code has been deposited at Zenodo and is publicly available as of the date of publication. DOIs are listed in the key resources table.

- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

METHOD DETAILS

Cell clustering and dropout detection

Bfimpute first provides an optional normalization step to smooth the gene expression values (counts per million, followed by logarithm base 10 with bias 1.01). Bfimpute then performs a local imputation within each cell group. We adopt the same approach as scImpute (Li and Li, 2018) to detect cell clusters, which applies spectral clustering methods on the result of Principal Component Analysis (PCA) to reduce the impact of dropout events. We integrate the 'Spectrum' function of the Spectrum R package (John et al., 2020) and the 'specc' function of the kernlab R package (Ng et al., 2001) for spectral clustering. Bfimpute also adopts the Gamma-Normal mixture distribution model to determine dropout events (Li and Li, 2018).

Probabilistic model for scRNA-seq expression matrix imputation

After above-mentioned steps, we then adapted a multi-variate priors model from Bayesian Probabilistic Matrix Factorization (BPMF) (Salakhutdinov and Mnih, 2008) to recover dropouts for scRNA-seq datasets. Since every cell group is mathematically equivalent, we arbitrarily choose one to demonstrate local imputation in Bfimpute. Suppose we have N genes and M cells in one cell group, and the expression matrix is $E \in \mathbb{R}^{N \times M}$. Each entity E_{ij} represents the expression level of gene i in cell j. Bfimpute factorizes E into $G \in \mathbb{R}^{D \times N}$ and $C \in \mathbb{R}^{D \times M}$ which are defined as gene and cell latent matrix, respectively, where D is the dimension of the latent factor. Column vector G_i and C_j represent the gene-specific and cell-specific latent vector, respectively. The imputed matrix to recover E will be given as $\widehat{E} = G^T C$.

We introduce the Gaussian noise model for the gene expression profile E with precision α, which was firstly proposed by Probabilistic Matrix Factorization (PMF) (Mnih and Salakhutdinov, 2008):

$$p(E|G, C, \alpha) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left[\mathcal{N}\left(E_{ij} | G_i^T C_j, \alpha^{-1} \right) \right]^{I_{ij}} \quad \text{(Equation 1)}$$

where I_{ij} is the indicator function that is 0 if the E_{ij} is a dropout and equal to 1 otherwise.

To get use of gene or cell related information such as bulk data or other data user provided, we add entity features $S^G \in \mathbb{R}^{F_G \times N}$ and $S^C \in \mathbb{R}^{F_C \times M}$ as gene and cell feature matrix, respectively, where F_G and F_C are the dimentionalities of these additional features. The Gaussian model for the prior distributions over genes and cells latent vectors adapted from Macau (Simm et al., 2017) will be given by:

$$\begin{aligned} p\left(G_i \middle| S_i^G, \mu_G, \Lambda_G, \beta_G\right) &= \mathcal{N}\left(G_i | \mu_G + {\beta_G}^T S_i^G, {\Lambda_G}^{-1}\right) \\ p\left(C_j \middle| S_j^C, \mu_C, \Lambda_C, \beta_C\right) &= \mathcal{N}\left(C_j | \mu_C + {\beta_C}^T S_j^C, {\Lambda_C}^{-1}\right) \end{aligned} \quad \text{(Equation 2)}$$

where $\{\mu_G, \mu_C\}$ and $\{\Lambda_G, \Lambda_C\}$ are the means and precisions, and $\beta_G \in \mathbb{R}^{F_G \times D}$ and $\beta_C \in \mathbb{R}^{F_C \times D}$ are the weight matrices for the entity features. Weight initialization by a zero mean normal distribution is used and they will be updated iteratively by the Bayesian inference steps (details described later). Also, direct imputation of single cell RNA-seq data could be applied by initiating zeros into feature vectors S^G and S^C(where $F_G = F_C = 1$) if no additional information is given.

The characteristics of each specific gene are represented in the D dimension of the gene latent matrix. If two genes are sharing similar functions, they will have similar latent vectors and thus, similar dot products with the cell latent matrix, which lead to similar expressions. The gene feature matrix is expected to represent additional information relative to the expression matrix. For instance, it could be a binary matrix showing the substructure information of the candidate genes or a bulk RNA-seq matrix which has the same genes with the scRNA-seq count matrix. The cell-specific matrices have similar biological meaning to gene-specific matrices but for cells.

To perform Bayesian inference, we introduce the priors referring to BPMF (Salakhutdinov and Mnih, 2008) for $\{\mu_G, \Lambda_G\}$ and $\{\mu_C, \Lambda_C\}$.

$$\begin{aligned} p(\mu_G, \Lambda_G | \mu_0, \beta_0, \nu_0, W_0) &= \mathcal{N}\left(\mu_G \middle| \mu_0, (\beta_0 \Lambda_G)^{-1}\right) \mathcal{W}(\Lambda_G | W_0, \nu_0) \\ p(\mu_C, \Lambda_C | \mu_0, \beta_0, \nu_0, W_0) &= \mathcal{N}\left(\mu_C \middle| \mu_0, (\beta_0 \Lambda_C)^{-1}\right) \mathcal{W}(\Lambda_C | W_0, \nu_0) \end{aligned} \quad \text{(Equation 3)}$$

where $\mathcal{W}$ is the Wishart Distribution with ν_0 as the degrees of freedom and W_0 as the scale matrix.

We also set a zero mean normal distribution as β_G and β_C's priors and a gamma distribution as the problem dependent α_G and α_C's hyperpriors (Simm et al., 2017):

$$\begin{aligned} p\left(\beta_G | \Lambda_G, \alpha_G\right) &= \mathcal{N}\left(vec(\beta_G) | 0, {\Lambda_G}^{-1} \otimes (\alpha_G \mathbf{I})^{-1}\right) \\ p\left(\beta_C | \Lambda_C, \alpha_C\right) &= \mathcal{N}\left(vec(\beta_C) | 0, {\Lambda_C}^{-1} \otimes (\alpha_C \mathbf{I})^{-1}\right) \end{aligned} \quad \text{(Equation 4)}$$

$$\begin{aligned} p(\alpha_G|k,\theta) &= \mathcal{G}(\alpha_G|k/2, 2\theta/k) \\ p(\alpha_C|k,\theta) &= \mathcal{G}(\alpha_C|k/2, 2\theta/k) \end{aligned} \qquad \text{(Equation 5)}$$

where $vec(\beta_X)$ is the vectorization of β_X, $\otimes$ represents the Kronecker product and α_X is the precision $(X \in \{G, C\})$. $k/2$ and $2\theta/k$ are shape and scale, respectively. k and θ are hyperparameters which are set to 1.

Gibbs sampler to impute dropout events

We use Markov Chain Monte Carlo (MCMC) algorithm to train Bfimpute, which is a sampling based approach to tackle the Bayesian inference problem. Bfimpute constructs a Markov Chain from a random initial value and after running the chain for $\tilde{K}$ steps, it will eventually converge to its stationary distribution. Bfimpute then uses the average of $(K - \tilde{K})$ stationary stages to approximate the real distribution of E and gain the estimated values $\widehat{E}_{ij}$ for dropouts:

$$p\left(\widehat{E}_{ij}|E,G,C\right) \approx \frac{1}{K-\tilde{K}} \sum_{k=\tilde{K}+1}^{K} p\left(\widehat{E}_{ij}|G_i^{(k)}, C_i^{(k)}, \alpha\right) \qquad \text{(Equation 6)}$$

More specifically, Bfimpute chooses Gibbs sampler to achieve Bayesian matrix factorization. In every cycle, we sample the conditional distribution from the posterior distribution in Bayes' theorem. Since the probabilistic models of genes and cells are symmetric, the conditional distributions over genes and the conditional distribution over cells have the same form. In particular, based on Equations (1) and (2), the conditional probability for G_i is:

$$p\left(G_i\middle|E,C,\alpha,S_i^G,\mu_G,\Lambda_G,\beta_G\right) = \mathcal{N}\left(G_i\middle|\mu_i^{(G)'},\Lambda_i^{(G)'}\right) \propto \prod_{j=1}^{M}\left[\mathcal{N}\left(E_{ij}|G_i{}^T C_j,\alpha^{-1}\right)\right]^{I_{ij}} \times p\left(G_i\middle|S_i^G,\mu_G,\Lambda_G,\beta_G\right) \qquad \text{(Equation 7)}$$

where

$$\begin{cases} \Lambda_i^{(G)'} = \Lambda_G + \alpha \sum_j \left(S_j S_j{}^T\right)^{I_{ij}} \\ \mu_i^{(G)'} = \left(\left[\Lambda_i^{(G)'}\right]^{-1}\right)\left[\Lambda_G\left(\mu_G + \beta_G{}^T x_i^{(G)}\right) + \alpha \sum_j (E_{ij}C_j)^{I_{ij}}\right] \end{cases}$$

According to Equations (2) and (3), we can derive the conditional probability for μ_G and Λ_G:

$$\begin{aligned} p\left(\mu_G,\Lambda_G\middle|G,S^G,\beta_G,\alpha_G,\mu_0,\beta_0,\nu_0,W_0\right) &= \mathcal{N}\left(\mu_G\middle|\mu_0{}',(\beta_0{}'\Lambda_G)^{-1}\right)\mathcal{W}(\Lambda_G|W_0{}',\nu_0{}') \\ &\propto p\left(G_i\middle|S_i^G,\mu_G,\Lambda_G,\beta_G\right) \times p(\mu_G,\Lambda_G|\mu_0,\beta_0,\nu_0,W_0\Big) \end{aligned} \qquad \text{(Equation 8)}$$

where

$$\begin{cases} \mu_0{}' = \dfrac{\beta_0\mu_0 + N\overline{G}}{\beta_0 + N} \\ \beta_0{}' = \beta_0 + N \\ \nu_0{}' = \nu_0 + N + F_G \\ W_0{}' = \left[W_0{}^{-1} + N\overline{H} + \beta_0\mu_0\mu_0{}^T - \beta_0{}'\mu_0{}'\mu_0{}'^T + \alpha_G\beta_G{}^T\beta_G\right]^{-1} \\ \overline{G} = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N}\left(G_i - \beta_G{}^T S_i^G\right) \\ \overline{H} = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N}\left(G_i - \beta_G{}^T S_i^G\right)\left(G_i - \beta_G{}^T S_i^G\right)^T \end{cases}$$

Considering Equations (4) and (5), we get the conditional probability for α_G:

$$p(\alpha_G|\beta_G,\Lambda_G,k,\theta) = \mathcal{G}(\alpha_G|k'/2, 2\theta'/k') \propto p(\beta_G|\Lambda_G,\alpha_G) \times p(\alpha_G|k,\theta) \qquad \text{(Equation 9)}$$

where

$$\begin{cases} k' = \dfrac{(F_G D + \theta)k}{\theta + \theta \cdot tr\left(\beta_G{}^T \beta_G \Lambda_G\right)} \\ \theta' = F_G D + \theta \end{cases}$$

From Equations (2) and (4), we are able to know the conditional probability for β_G:

$$p\left(\beta_G \middle| \Lambda_G, \alpha_G, G, S^G, \mu_G\right) = \mathcal{N}\left(\mu_{\beta_G}, \Lambda_{\beta_G}\right) \\ \propto p\left(\beta_G | \Lambda_G, \alpha_G) \times \prod_i p\left(G_i | S_i^G, \mu_G, \Lambda_G, \beta_G\right)\right) \qquad \text{(Equation 10)}$$

Because the size of the precision matrix $\Lambda\ \{\beta\ G\}$ is too large to compute, we consider to do this part in an alternative way (Simm et al., 2017) by calculating:

$$\tilde{\beta}_G = \left(S^{G^T} S^G + \alpha_G \mathbf{I}\right)^{-1} \left(S^{G^T} (\tilde{G} + E_1) + \sqrt{\alpha_G} E_2\right) \qquad \text{(Equation 11)}$$

where $\tilde{G} = (G - \mu_G)^T$, and each row of $E_1 \in \mathbb{R}^{N \times D}$ and $E_2 \in \mathbb{R}^{F_G \times D}$ is sampled from $\mathcal{N}(0, \Lambda_G^{-1})$. The Gibbs sampling framework of Bfimpute is shown below:

Gibbs sampling for Bfimpute

1. Initialize $\{\mathbf{G^0}, \mathbf{C^0}, \beta_G^{(0)}, \beta_C^{(0)}, \alpha_G^{(0)}, \alpha_C^{(0)}\}$
2. For k = 1, 2, …, K
    a. Sample the means $\{\mu_G, \mu_C\}$ and precisions $\{\Lambda_G, \Lambda_G\}$ of gene and cell latent matrices:

    $$\mu_G{}^{(k)}, \Lambda_G{}^{(k)} \sim p\left(\mu_G, \Lambda_G \middle| G^{(k-1)}, S^G, \beta_G{}^{(k-1)}, \alpha_G{}^{(k-1)}\right)$$

    $$\mu_C{}^{(k)}, \Lambda_C{}^{(k)} \sim p\left(\mu_C, \Lambda_C \middle| C^{(k-1)}, S^C, \beta_C{}^{(k-1)}, \alpha_C{}^{(k-1)}\right)$$

    b. Sample gene and cell latent matrices {G, C}:
    - For each i = 1, …, N sample gene latent vectors in parallel:

    $$G_i{}^{(k)} \sim p\left(G_i \middle| E, C^{(k-1)}, S_i^G, \mu_G{}^{(k)}, \Lambda_G{}^{(k)}, \beta_G{}^{(k-1)}\right)$$

    - For each i = 1, …, M sample cell latent vectors in parallel:

    $$C_i{}^{(k)} \sim p\left(C_i \middle| E, G^{(k-1)}, S_i^C, \mu_C{}^{(k)}, \Lambda_C{}^{(k)}, \beta_C{}^{(k-1)}\right)$$

    c. Sample the precisions $\{\alpha_G, \alpha_C\}$ of weight matrices:

    $$\alpha_G{}^{(k)} \sim p\left(\alpha_G | \beta_G{}^{(k-1)}, \Lambda_G{}^{(k)}\right)$$

    $$\alpha_C{}^{(k)} \sim p\left(\alpha_C | \beta_C{}^{(k-1)}, \Lambda_C{}^{(k)}\right)$$

    d. Sample weight matrices $\{\beta_G, \beta_C\}$:

    $$\beta_G{}^{(k)} = \left(S^{G^T} S^G + \alpha_G{}^{(k)} \mathbf{I}\right)^{-1} \left(S^{G^T} \left(\tilde{G}^{(k)} + E_1\right) + \sqrt{\alpha_G{}^{(k)}} E_2\right)$$

    $$\beta_C{}^{(k)} = \left(S^{C^T} S^C + \alpha_C{}^{(k)} \mathbf{I}\right)^{-1} \left(S^{C^T} \left(\tilde{C}^{(k)} + E_1\right) + \sqrt{\alpha_C{}^{(k)}} E_2\right)$$

Generation of simulated data

We first simulated a single cell RNA-seq count matrix with 20,000 genes and 500 cells evenly split into 5 groups using the scater (McCarthy et al., 2017) package and Splatter (Zappia et al., 2017) package. The parameter which controls the probability that a gene will be selected as DE was set to 0.08 while the location and scale factor were set to 0.3 and 0.5, respectively. We used 'experiment' to add the global dropout for every cell. In order to show the universal applicability of Bfimpute, we further generated 6, 7, 8 groups of cells with 600, 700, 800 as total cell numbers and 10 runs for each data with different seeds using the same parameters mentioned above.

Quality control for real datasets

We did quality control (QC) (function from R package scDatasets) for all real datasets to ensure fairness for all methods before imputation except for Zheng dataset (PBMCs) and the Vladoiu dataset. As they are based on 10x Genomics platform with an extremely high dropout rate, the QC step for them may remove and lose nearly 80% genes.

Evaluation metrics of clustering results

We used four evaluation methods: adjusted Rand index (Morey and Agresti, 1984), Jaccard index (Jaccard, 1912), normalized mutual information (nmi) (Strehl and Ghosh, 2002), and purity score, to analyse the agreement between true cluster labels and the spectral clustering (John et al., 2020) results on the first several Principle Components (PCs) of imputed matrix. Most of these four measurements vary from 0 to 1, with 1 indicating perfect match between them, except the adjusted Rand index which could yield negative values when agreement is less than expected by chance. The adjusted Rand index is an adjusted version of Rand's statistic (Rand, 1971) which is the probability that a randomly selected pair is classified in agreement. The Jaccard index is similar to Rand Index, but disregards the pairs of elements that are in different clusters for both clusterings (Wagner and Wagner, 2007). The normalized mutual information combines multiple clusterings into a single one without accessing the original features or algorithms that determine these clusterings. The purity score shows the rate of the total number of cells that are classified correctly.

We also used silhouette scores to evaluate the quality of spectral clustering (John et al., 2020) results on the first several Principle Components (PCs) of the imputed matrix based on how well its data points are clustered. Each data point is assigned to a silhouette measure to represent how close a data point is to its own cluster in comparison to other clusters. The silhouette varies from −1 to +1, with a high value indicating the data point is well matched to its own cluster and poorly matched to neighboring clusters.

Measurement of pseudotime ordering

We used two measurements: Pseudo-temporal Ordering Score (POS) and Kendalls rank correlation score, to evaluate the consistency between the pseudotime ordering and the time labels. POS is a quantitative measure of the reliability of numerous possible pseudotime course proposed by TSCAN (Ji and Ji, 2016). Kendall's rank correlation score is a traditional statistical measurement for ordinal association between two measured quantities.

Gene-gene interaction and gene ontology enrichment analysis on gene latent matrix

Bfimpute is able to generate cell and gene latent matrices for each cell type after imputation which allow us to further investigate the gene-gene relationships and cell functions. To first analyze gene-gene interactions network, we used the 'blockwiseModules' function from the WGCNA package (Langfelder and Horvath, 2008) to construct the weighted gene co-expression network from the gene latent matrix of one cell type and obtain a number of gene clusters (modules) of highly correlated genes. We then performed Gene Ontology (GO) enrichment analysis using 'enrichGO' from the clusterProfiler package (Yu et al., 2012) and detected 'Cellular Component' related GO items in each gene module.

QUANTIFICATION AND STATISTICAL ANALYSIS

Statistical details and software used for various types of data analyses in this work are cited in the appropriate sections in the STAR Methods. The agreements between true cluster labels and spectral clustering results from scRNA-seq data without or with imputation were calculated using adjusted Rand index, Jaccard index, nmi, purity score, and silhouette score. The agreements between true time label and pseudotime trajectory built from scRNA-seq data without or with imputation were calculated using POS and Kendalls rank correlation score.