Accuracy, Robustness and Scalability of Dimensionality Reduction Methods for Single Cell RNAseq Analysis

Shiquan Sun[1, 2], Jiaqiang Zhu[2], Ying Ma[2] and Xiang Zhou[2, 3, #]

1. School of Computer Science, Northwestern Polytechnical University, Xi’an, Shaanxi 710129, P.R. China

2. Department of Biostatistics, University of Michigan, Ann Arbor, MI 48109, USA

3. Center for Statistical Genetics, University of Michigan, Ann Arbor, MI 48109, USA

#: correspondence to XZ (xzhousph@umich.edu)

ABSTRACT

Dimensionality reduction (DR) is an indispensable analytic component for many areas of single cell RNA sequencing (scRNAseq) data analysis. Proper DR can allow for effective noise removal and facilitate many downstream analyses that include cell clustering and lineage reconstruction. Unfortunately, despite the critical importance of DR in scRNAseq analysis and the vast number of DR methods developed for scRNAseq studies, however, few comprehensive comparison studies have been performed thus far to evaluate the effectiveness of different DR methods in scRNAseq. Here, we aim to fill this critical knowledge gap by providing a comprehensive comparative evaluation of a variety of commonly used DR methods for scRNAseq studies. Specifically, we compared 11 different DR methods on 28 publicly available scRNAseq data sets that cover a range of sequencing techniques and sample sizes. We evaluate the performance of different DR methods both for cell clustering and for lineage reconstruction in terms of their accuracy and robustness. We evaluate the computational scalability of different DR methods by recording their computational cost. Based on the comprehensive evaluation results, we provide important guidelines for choosing DR methods in scRNAseq data analysis. We also provide all analysis scripted used in the present study at www.xzlab.org/reproduce.html. Together, we hope that our results will serve as an important practical reference for practitioners to choose DR methods in the field of scRNAseq analysis.

INTRODUCTION

Single-cell RNA sequencing (scRNAseq) is a rapidly growing and widely applying technology [1-3]. By measuring gene expression at single cell level, scRNAseq provides an unprecedented opportunity to investigate the cellular heterogeneity of complex tissues [4-8]. However, despite the popularity of scRNAseq, analyzing scRNAseq data remains a challenging task. Specifically, due to the low capture efficiency and low sequencing depth per cell in scRNAseq data, gene expression measurements obtained from scRNAseq are noisy: collected scRNAseq gene measurements are often in the form of low expression counts paired with an excessive number of zeros known as dropouts [9]. Subsequently, dimensionality reduction (DR) methods that transform the original high-dimensional noisy expression matrix into a low-dimensional subspace with enriched signals become an important data processing step for scRNAseq analysis [10]. Proper DR can allow for effective noise removal, facilitate data visualization, and enable efficient and effective downstream analysis of scRNAseq [11].

DR is indispensable for many types of scRNAseq analysis. Because of the importance of DR in scRNAseq analysis, many DR methods have been developed and routinely used in many scRNAseq software tools that include, but not limited to, cell clustering tools [12, 13] and lineage reconstruction tools [14]. Indeed, most commonly used scRNAseq clustering methods rely on DR as the first analytic step [15]. For example, Seurat applies clustering algorithms directly on a low dimensional space inferred from principal component analysis (PCA) [16]. CIDR improves clustering by improving PCA through imputation [17]. SC3 combines different ways of PCA for consensus clustering [18]. Besides PCA, other DR techniques are also commonly used for cell clustering. For example, nonnegative matrix factorization (NMF) is used in SOUP [19]. Diffusion map is used in destiny [20]. Multidimensional scaling (MSD) is used in ascend [21]. Variational inference autoencoder is used in scVI [22]. In addition to cell clustering, most cell lineage reconstruction and developmental trajectory inference algorithms also rely on DR [14]. For example, TSCAN builds cell lineages using minimum spanning tree based on a low dimensional PCA space [23]. Waterfall performs *k*-means clustering in the PCA space to eventually produce linear trajectories [24]. SLICER uses locally linear embedding (LLE) to project the set of cells into a lower dimension space for reconstructing complex cellular trajectories [25]. Monocle employs independent components analysis (ICA) for DR before building the trajectory [26]. Wishbone combines PCA and diffusion maps to allow for bifurcation trajectories [27].

Besides the generic DR methods mentioned in the above paragraph, many DR methods have also been developed recently that are specifically targeted for modeling scRNAseq data. These scRNAseq specific DR methods can account for either the count nature of scRNAseq data and/or the dropout events commonly encountered in scRNAseq studies. For example, ZIFA relies on a zero-inflation normal model to model dropout events [28]. pCMF models both dropout events and the mean-variance dependence resulting from the count nature of scRNAseq data [29]. ZINB-WaVE incorporates additional gene-level and sample-level covariates for more accurate DR [30]. Finally, several deep learning-based DR methods have recently been developed to enable scalable and effective computation in large-scale scRNAseq data; for example, data that are collected by 10X Genomics techniques [31] and/or from large consortium studies such as Human Cell Atlas (HCA) [32, 33]. Common deep learning-based DR methods for scRNAseq include Dhaka [34], scScope [35], VASC [36], and DCA [37], to name a few.

With all these different DR methods for scRNAseq data analysis, one naturally wonders which DR method one would prefer for different types of scRNAseq analysis. Unfortunately, despite the popularity of scRNAseq technique, the critical importance of DR in scRNAseq analysis, and the vast number of DR methods developed for scRNAseq studies, few comprehensive comparison studies have been performed to evaluate the effectiveness of different DR methods for practical applications. Here, we aim to fill this critical knowledge gap by providing a comprehensive comparative evaluation of a variety of commonly used DR methods for scRNAseq studies. Specifically, we compared 11 different DR methods on 28 publicly available scRNAseq data sets that cover a range of sequencing techniques and sample sizes. We evaluate the performance of different DR methods both for cell clustering and for lineage reconstruction in terms of their accuracy and robustness using different metrics. We also evaluate the computational scalability of different DR methods by recording their computational time. Together, we hope our results can serve as an important guideline for practitioners to choose DR methods in the field of scRNAseq analysis.

RESULTS

Our evaluated the performance of 11 DR methods (Table 1; Figure S1) on 28 publicly available scRNAseq data sets (Table S1). Details of these data sets are provided in Methods and Materials. Briefly, these data sets cover a wide variety of sequencing techniques that include Smart-Seq2 (7 data sets), 10X genomics (6 data sets), Fluidigm C1 (4 data sets), Smart-Seq (5 data sets), inDrop (1 data set), SMARTer (3 data sets) and others (2 data sets). In addition, these data cover a range of sample sizes from a couple hundred cells to a few thousand cells. In each data set, we evaluated the effectiveness of different DR methods for one of the two important tasks: cell clustering and lineage inference. In addition, we recorded computation time of different DR methods. An overview of the comparison workflow is shown in Figure 1. All data and analysis scripts for reproducing the results in the paper is available at www.xzlab.org/reproduce.html.

Cell clustering

We first evaluated the performance of different DR methods for cell clustering applications. To do so, we obtained 14 publicly available scRNAseq data sets and simulated two additional scRNAseq data sets using *Splatter* package (Table S1). Each of the 14 real scRNAseq data sets contains known cell clustering information while each of the two simulated data sets contains 4 or 8 known cell types. For each DR method and each data set, we applied DR to extract a fixed number of low-dimensional components (e.g. these are the principal components in the case of PCA). We varied the number of low-dimensional components to examine their influence on downstream analysis. In particular, for a data with less than or equal to 300 cells, we varied the number of low dimensional components to be either 2, 6, 14, or 20. For a data with more than 300 cells, we varied the number of low dimensional components to be either 0.5%, 1%, 2%, or 3% of the total number of cells. We then applied either the hierarchical clustering algorithm or the *k*-means clustering algorithm to obtain cluster labels. We used both normalized mutual information (NMI) and adjusted rand index (ARI) values for comparing the true cell labels and inferred cell labels obtained by clustering methods based on the low dimensional components.

The evaluation results on DR methods based on clustering analysis using the *k*-means clustering algorithm are summarized in Figure 2 (for NMI criterion) and Figure S2 (for ARI criterion). Because the results based on either of the two criteria are similar, we will mainly explain the results based on the NMI criteria in Figure 2. For easy visualization, we also display the results averaged across data sets in

Figure S3. A few patterns are noticeable. First, as one would expect, clustering accuracy depends on the number of low-dimensional components that are used for clustering. Specifically, accuracy is relatively low when the number of low-dimensional components is very small (e.g. 2 or 0.5%) and generally increases with the number of included components. In addition, accuracy usually saturates once a sufficient number of components is included, though the saturation number of components can vary across data sets and across methods. For example, the average NMI across all data sets and across all methods are 0.59, 0.67, 0.68 and 0.69 for increasingly large number of components, respectively. Second, when conditional on using a low number of components, scRNAseq specific DR method ZINB-WaVE and scRNAseq non-specific DR method ICA often outperform the other methods. For example, with the lowest number of components, the average NMI across all data sets for ICA and ZINB-WaVE is 0.77 and 0.76, respectively. The performance of ICA and ZINB-WaVE is followed by Diffusion Map (0.71), ZIFA (0.69), PCA (0.68), FA (0.68), NMF (0.59) and DCA (0.57). While the remaining three methods, Poisson NMF (0.42), pCMF (0.41) and scScope (0.26), do not fare well with a low number of components. The good performance of ZINB-WaVE with a low number of components presumably is because its direct modeling of dropout events and raw counts of scRNAseq data. The good performance of ICA with a low number of components presumably is because it extracts low-dimensional components using a non-linear transformation. Third, with increasing number of low-dimensional components, generic methods such as FA, ICA and PCA are often comparable with scRNAseq specific methods such as ZINB-WaVE, and in many cases can slightly outperform ZINB-WaVE. For example, with the highest number of low-dimensional components, the average NMI across all data sets for FA, ICA, PCA, ZINB-WaVE and Diffusion Map are 0.85, 0.84, 0.83, 0.83 and 0.80, respectively. Their performance is followed by ZIFA (0.79), NMF (0.73), and DCA (0.69). The same three methods, pCMF (0.55), Poisson NMF (0.31), and scScope (0.31), again do not fare well with a large number of low-dimensional components. The comparable results of generic DR methods with scRNAseq specific DR methods with a high number of low-dimensional components are also consistent some of the previous observations; for example, the original ZINB-WaVE paper observed that PCA generally can yield comparable results with scRNAseq specific DR methods in real data [30].

Besides the *k*-means clustering algorithm, we also used the hierarchical clustering algorithm to evaluate the performance of different DR methods (Figures S4-S6). In this comparison, we had to exclude one DR method, scScope, as hierarchical clustering does not work on the extracted low-dimensional components from scScope. Consistent with the *k*-means clustering results, we found that the clustering accuracy measured by hierarchical clustering is relatively low when the number of low-dimensional components is very small (e.g. 2 or 0.5%), but generally increases with the number of included components. In addition, consistent with the *k*-means clustering results, we found that generic DR methods often yield results comparable to or better than scRNAseq specific DR methods. In particular, two generic DR methods, FA and NMF, outperform various other DR methods across a range of settings. For example, when the number of low-dimensional components is moderate (6 or 1%), both FA and NMF achieve an average NMI value of 0.80 across data sets (Figures S6). Their performance is followed by PCA (0.72), Poisson NMF (0.71), ZINB-WaVE (0.71), Diffusion Map (0.70), ICA (0.69), ZIFA (0.68), pCMF (0.65), and DCA (0.63). We note, however, that the clustering results obtained by hierarchical clustering are often slightly worse than that obtained by *k*-means clustering across settings (e.g. Figure S3 vs Figure S6), consistent with the fact that many scRNAseq clustering methods use *k*-means as a key ingredient [18, 24].

While some DR methods (e.g. Poisson NMF, ZINB-WaVE, pCMF and DCA) directly model count data, many DR methods (e.g. DR methods, PCA, ICA, FA, NMF, and Diffusion Map) require normalized data. The performance of DR methods that use normalized data may depend on how data are normalized. Therefore, we investigated how different normalization approaches impact on the performance of the aforementioned five DR methods that use normalized data. We examined two alternative data transformation approaches, log2 CPM (count per million) and z-score, in addition to the log2 count we used in the previous results (transformation details are provided in Methods and Materials). The evaluation results are summarized in Figures S7-S10 and are generally insensitive to the transformation approach deployed. For example, with the *k*-means clustering algorithm, when the number of low-dimensional components is small (1%), PCA achieves an NMI value of 0.82, 0.82 and 0.81, for log2 count transformation, log2 CPM transformation, and z-score transformation, respectively (Figures S3A, S8A, and S10A). Similar results hold for the hierarchical clustering algorithm (Figures S3B, S8B, and S10B). Therefore, different data transformation approaches do not appear to substantially influence the performance of DR methods.

Finally, we also investigated the stability and robustness of different DR methods. To do so, we randomly split the *Kumar* data into two subsets with an equal number of cells for each cell type in the two subsets. We applied each DR method to the two subsets and measured the clustering performance in each subset separately. We repeated the procedure 10 times to capture the potential stochasticity during the data split. We visualize the clustering performance of different DR methods in the two subsets separately. Such visualization allows us to check the effectiveness of DR methods with respective to reduced sample size in the subset, as well as the stability/variability of DR methods across different split replicates (Figure S11). The results show that four of the DR methods, PCA, ICA, FA, and ZINB-WaVE, often achieve both accurate clustering performance and highly stable and consistent results across the subsets. The accurate and stable performance of both ICA and ZINB-WaVE is notable even with a relatively small number of low-dimensional components. For example, with very small number of low-dimensional components, both ICA and ZINB-WaVE achieve an average NMI value of 0.98 across the two subsets, with virtually no performance variability across data splits (Figure S11).

Overall, the results suggest that, in terms of downstream clustering analysis accuracy and stability, ICA is preferable across a range of data sets examined here. In addition, scRNAseq specific DR methods such as ZINB-WaVE is also preferable if one is interested in extracting a small number of low-dimensional components, while generic methods such as PCA or FA are also preferred when one is interested in extracting a large number of low-dimensional components.

Trajectory inference

We next evaluated the performance of different DR methods for lineage inference applications (details in Methods and Materials). To do so, we obtained 14 publicly available scRNAseq data sets, each of which contains known lineage information (Table S2). The known lineage in all these data are linear, without bifurcation or multifurcation patterns. For each data set, we applied one DR method at a time to extract a fixed number of low-dimensional components. In the process, we varied the number of low-dimensional components from 2, 6, 14 to 20 to examine their influence for downstream analysis. With the extracted low-dimensional components, we first used the hierarchical clustering algorithm or the *k*-means clustering algorithm to obtain cell type labels, where the number of cell types in the clustering was set to be the known truth. Afterwards, we supplied the low-dimensional components and cell type labels to the software *Slingshot* [38] to infer the lineage. Following [38], we evaluated the performance of DR methods by Kendall correlation coefficient that compares the true lineage and inferred lineage obtained based on the low-dimensional components. In this comparison, we also excluded one DR method, scScope, which is not compatible with *Slingshot*. The lineage inference results for the remaining DR methods are summarized in Figures 3 and S12-17.

Different from the clustering results where accuracy generally increases with increasing number of included low-dimensional components, the lineage tracing results do not show a clear increasing pattern with respect to the number of low-dimensional components, especially when we used *k*-means clustering as the initial step (Figures 3 and Figure S13A). For example, the average Kendall correlation across all data sets and across all methods are 0.37, 0.37, 0.38 and 0.37 for increasingly large number of components, respectively. When we use hierarchical clustering algorithm as the initial step, the lineage tracing results in the case of a small number of low-dimensional components are slightly inferior as compared to the results obtained using a large number of low-dimensional components (Figures S12 and Figure S13B). However, we do note that the lineage tracing results obtained using *k*-means are better than that obtained using hierarchical clustering as the initial step. Therefore, *k*-means clustering algorithm is recommend as the initial step for lineage inference and a small number of low-dimensional components there is often sufficient for accurate results. When conducting lineage inference based on a low number of components with *k*-means, we found that four DR methods, PCA, FA, NMF and ZINB-WaVE, all perform well for lineage inference (Figure S13A). These methods also worked reasonably well for the previous cell clustering analysis. For example, with the lowest number of components, the average Kendall correlation across data sets for PCA, FA, NMF, and ZINB-WaVE, are 0.43, 0.42, 0.42, and 0.41, respectively. Their performance is followed by ICA (0.38), ZIFA (0.37), and Diffusion Map (0.35). While DCA (0.31), pCMF (0.29), and Poisson NMF (0.29) do not fare well. These four methods (PCA, FA, NMF and ZINB-WaVE), with the only exception of NMF, are also among the best performers for lineage inference across varying number of low-dimension components.

For methods that require normalized data, we further examined the influence of different data transformation approaches on their performance (Figures S14-S15). Like in the clustering comparison, we found that different transformations do not influence the performance results for most DR methods in lineage inference. For example, with the *k*-means clustering algorithm as the initial step, when the number of low-dimensional components is small, ICA achieves a Kendall correlation of 0.38, 0.36 and 0.38, for log2 count transformation, log2 CPM transformation, and z-score transformation, respectively (Figures S13A, S15A, and S17A). Similar results hold for the hierarchical clustering algorithm (Figures S13B, S15B, and S17B). However, some notable exceptions exist. For example, with log2 CPM transformation but not the other transformations, the performance of Diffusion Map increases with increasing number of included components when *k*-means clustering was used as the initial step: the average Kendal correlation across different low-dimensional components are 0.37, 0.42, 0.44, and 0.47, respectively (Figures S14 and S15A). As another example, with z-score transformation but not with the other transformations, FA achieves the highest performance among all DR methods across different number of low-dimensional components.

We also investigated the stability and robustness of different DR methods by data split on the *Hayashi* data. We applied each DR method to the two subsets and measured the lineage inference performance in the two subsets separately. We again visualize the clustering performance of different DR methods in the two subsets, separately. Such visualization allows us to check the effectiveness of DR methods with respective to reduced sample size in the subset, as well as the stability/variability of DR methods across different split replicates (Figure S18). The results show that three of the DR methods, FA, Diffusion Map, and ZINB-WaVE, often achieve both accurate performance and highly stable and consistent results across the subsets. The accurate and stable performance of these is notable even with a relatively small number of low-dimensional components. For example, with very small number of low-dimensional components, FA, Diffusion Map and ZINB-WaVE achieve Kendall correlation of 0.75, 0.77, and 0.77 averaged across the two subsets, respectively, and again with virtually no performance variability across data splits (Figure S18).

Overall, the results suggest that, in terms of downstream lineage inference accuracy and stability, the scRNAseq non-specific DR method FA is preferable cross a range of data sets examined here. In addition, scRNAseq specific DR method ZINB-WaVE and the scRNAseq non-specific DR method NMF are also preferable if one is interested in extracting a small number of low-dimensional components. scRNAseq specific DR method Diffusion Map may also be preferable if one is interested in extracting a large number of low-dimensional components.

Computation time

We recorded and compared computing time for different DR methods on simulated data sets. Here, we also examined how computation time for different DR methods varies with respect to the number of low-dimensional components extracted (Figure 4A) as well as with respect to the number of cells contained in the data (Figure 4B). Overall, the computational cost of three methods, ZINB-WaVE, ZIFA, and pCMF, is substantially heavier than the remaining methods. Their computation time increase substantially with both increasingly large number of low-dimensional components and increasingly large number of cells in the data. Specifically, when the sample size equals 500 and the desired number of low dimensional components equals 22, the computing time for ZINB-WaVE, ZIFA, and pCMF to analyze 10,000 genes are 2.15, 1.33, and 1.95 hours, respectively (Figure 4A). When the sample size increases to 10,000, the computing time for ZINB-WaVE, ZIFA, and pCMF increases to 12.49, 20.50, and 15.95 hours, respectively (Figure 4B). Similarly, when the number of low-dimensional components increases to 52, the computing time for ZINB-WaVE, ZIFA, and pCMF increases to 4.56, 4.27, and 4.62 hours, respectively. Besides these three methods, the computing cost of both ICA and Poisson NMF can also increase noticeably with increasingly large number of low-dimensional components. The computing cost of ICA, but to a lesser extent of Poisson NFM, also increases substantially with increasingly large number of cells. In contrast, PCA, FA, Diffusion Map, and the two deep learning-based methods (DCA, and scScope) are computationally efficient. In particular, the computation time for these five methods are stable and do not show substantial dependence on the sample size or the number of low-dimensional components. Certainly, we expect that the computation time of all DR methods will further increase as the sample size of the scRNAseq data sets increases in magnitude. Overall, in terms of computing time, PCA, FA, Diffusion Map, DCA, and scScope are preferable.

Practical guidelines

In summary, our comparison analysis shows that different DR methods can have different merits for different tasks. Subsequently, it is not straightforward to identify a single DR method that strives the best in all data sets and for all downstream analyses. Instead, we provide a relatively comprehensive practical guideline for choosing DR methods in scRNAseq analysis in Figure 5. Our guideline is based on the accuracy and effectiveness of DR method in terms of the downstream analysis, the robustness and stability of DR method in terms of replicability and consistency across data splits, as well as computational scalability for large scRNAseq data sets. Briefly, for cell clustering analysis, PCA, ICA, FA and ZINB-

WaVE are recommended for small data where computation is not a concern. In contrast, PCA, ICA, FA are recommended for large data where computation is a concern. For lineage inference analysis, FA, NMF and ZINB-WaVE are all recommended for small data. In contrast, a subset of these methods, FA is also recommended for large scRNAseq data. In addition, for very large scRNAseq data sets (e.g. >100,000 samples), DCA perhaps is the only feasible approach with reasonable performance for both downstream analyses. Finally, beside these general recommendations, we note that some methods have additional features that are desirable for practitioners. For example, ZINB-WaVE can include sample-level and gene-level covariates, thus allowing us to easily control for batch effects or size factors. We provide our detailed recommendations in Figure 5.

DISCUSSION

We have presented a comprehensive comparison of different dimensionality reduction methods for scRNAseq analysis based on two important downstream applications: cell clustering and trajectory inference. We hope the summary of these state-of-the-art DR methods, the detailed comparison results, and the recommendations and guidelines for choosing DR methods can assist researchers in the analysis of their own scRNAseq data.

We have primarily focused on evaluating DR methods based on two downstream applications. We did not, however, examine the performance of DR methods for data visualization purposes. Data visualization aims to project scRNAseq data into a two- or three-dimensional subspace for visualizing cell clustering results. For example, both tSNE [39] and UMAP [40] are commonly applied data visualization tools. Different from cell clustering, which often rely on a relatively large number of low-dimensional components, data visualization focuses on using only the top two or three low-dimensional components. Subsequently, DR methods for data visualization may not fare well for cell clustering, and DR methods for cell clustering may not fare well for visualization [41]. Besides data visualization, we note that DR methods are also used for many other analytic tasks in scRNAseq studies. For example, factor models for DR is an important modeling part for multiple scRNAseq data sets alignment [16], for integrative analysis of multiple omics data sets [42, 43], as well as for deconvoluting bulk RNAseq data using cell type specific gene expression measurements from scRNAseq [44, 45]. In addition, cell classification in scRNAseq also relies on a low-dimensional structure inferred from original scRNAseq through DR [46, 47]. Therefore, the comparative results obtained from the present study can provide important insights into these different scRNAseq analytic tasks. In addition, investigating the performance of DR methods in these different scRNAseq downstream analyses is an important future research direction.

We mostly focused on evaluating feature extraction methods for DR. Another important category of DR method is the feature selection method, which aims to select a subset of features/genes directly from the original feature space. The feature section methods rely on different criteria to select important genes and are also commonly used in the preprocessing step of scRNAseq data analysis [48]. For example, M3Drop relies on dropout events in scRNAseq data to identify informative genes [49]. Seurat uses gene expression variance to select highly variable genes [16]. Evaluating the benefits of different methods and criteria for selecting informative genes for different downstream tasks is another important future direction.

Due to the heavy computing cost of several DR methods, we unfortunately have to limit our comparisons to scRNAseq data sets with sample sizes less than 10,000 cells. With the advance of scRNAseq technologies and with the increase collaborations across scientific groups, new consortium projects such as the Human Cell Atlas (HCA) will generate scRNAseq data sets that contain millions of cells [32]. The large data at this scale poses critical computational and statistical challenges to many current DR methods. Many existing DR methods, in particular those that require the computation and memory storage of a covariance or distance matrix among cells, will no longer be applicable there. Therefore, new algorithmic innovations and new efficient computational approximations will likely be needed to scale many of the existing DR methods to millions of cells.

METHODS AND MATERIALS

scRNAseq data sets

We obtained a total of 28 scRNAseq data sets from public domains for benchmarking DR methods. All data sets were retrieved from the Gene Expression Omnibus (GEO) database (https://www.ncbi.nlm.nih.gov/geo/) or the 10X genomics website (https://support.10xgenomics.com/single-cell-gene-expression/datasets). These data sets cover a wide variety of sequencing techniques that include Smart-Seq2 (7 data sets), 10X genomics (6 data sets), Fluidigm C1 (4 data sets), Smart-Seq (5 data sets), SMARTer (3 data sets), inDrop (1 data set) and others (2 data sets). In addition, these data cover a range of sample sizes from a couple hundred cells to several thousand cells measured in either human (16 data sets) or mouse (12 data sets). In each data set, we evaluated the effectiveness of different DR methods for one of the two important downstream analysis tasks: cell clustering and lineage inference. In particular, 14 data sets were used for cell clustering evaluation while another 14 data sets were used for lineage inference evaluation. For cell clustering, 10 of the 14 data sets were obtained by mixing cells from different cell types either pre-determined by fluorescence activated cell sorting (FACS) or cultured on different conditions. Therefore, these 10 studies contain the *true* cell type labels for all cells. The remaining 4 data sets contain cell labels that were determined in the original study and we simply treated them as truth though we do acknowledge that such “true” clustering information may not be accurate. For lineage inference, 4 of the 14 data sets were obtained by mixing cells from different cell types pre-determined by FACS. These different cell types are at different developmental stages of a single linear lineage; thus these 4 studies contain the *true* lineage information for all cells. The remaining 10 data sets contain cells that were collected at multiple time points during the development process. For these data, we simply treated cells at these different time points as part of a single linear lineage, though we do acknowledge that different cells collected at the same time point may represent different developmental trajectories from an early time point if the cells at the early time are heterogeneous. In either case, the true lineage in all these 14 data sets are treated as linear, without any bifurcation or multifurcation patterns.

A detailed list of the selected scRNAseq datasets with corresponding data features is provided in Tables S1-S2. In each of the above 28 data sets, we removed genes that are expressed in less than five cells. For methods modeling normalized data, we transformed the raw counts data into continuous data with the *normalize* function implemented in *scater* (R package v1.12.0). We then applied log2 transformation on the normalized counts by adding one to avoid log transforming zero values. We simply term this normalization as log2 count transformation, though we do acknowledge that such transformation does take into account of cell size factor etc. through the *scater* software. In addition to log2 count transformation, we also explore the utility of two additional data transformation: log2 CPM transformation and z-score transformation. In the log2 CPM transformation, we first computed counts per million reads (CPM) and then performed log2 transformation on the resulted CPM value by adding a constant of one to avoid log transformation of zero quantities. In the z-score transformation, for each gene in turn, we standardized CPM values to achieve a mean of zero and variance of one across cells using *Seurat* package (v2.3).

Besides the above 28 real scRNAseq data sets, we also simulated 2 additional scRNAseq data sets for cell clustering evaluation. In the simulations, we used all 94 cells from one cell type (*v6.5 mouse 2i+LIF*) in the Kumar data as input. We simulated scRNAseq data with 500 cells and a known number of cell types, which were set to be either 4 or 8, using the *Splatter* package v1.2.0. All parameters used in the *Splatter* (e.g., mean rate, shape, dropout rate, etc.) were set to be approximately those estimated from the real data. In the case of 4 cell types, we set the group parameter in *Splatter* as 4. We set the percentage of cells in each group as 0.1, 0.15, 0.5 and 0.25, respectively. We set the proportion of the differentially expressed genes in each group as 0.02, 0.03, 0.05 and 0.1, respectively. In the case of 8 cell types, we set group/cell type parameter as 8. We set the percentage of cells in each group as 0.12, 0.08, 0.1, 0.05, 0.3, 0.1, 0.2 and 0.05, respectively. We set the proportion of the differentially expressed genes in each group as 0.03, 0.03, 0.03, 0.1, 0.05, 0.07, 0.08, and 0.1, respectively.

Compared dimensionality reduction methods

DR methods aim to transform an originally high-dimensional feature space into a low-dimensional representation with a much-reduced number of components. These components are in the form of a linear or non-linear combination of the original features (known as feature extraction DR methods) and in the extreme case are themselves a subset of the original features (known as feature selection DR methods). In the present study, we have collected and compiled a list of 11 popular and widely used DR methods in the field of scRNAseq analysis. These DR methods include factor analysis (FA; R package *psych*, v1.8.12), principal component analysis (PCA; R package *stats*, v3.6.0), independent component analysis (ICA; R package *ica*, v1.0.2), Diffusion Map (Diffusion Map; R package *destiny*, v2.14.0), nonnegative matrix factorization (NMF; R package NNLM, v1.0.0), Kullback-Leibler divergence-based NMF (Poisson NMF; R package NNLM, v1.0.0), zero-inflated factor analysis (ZIFA; Python package *ZIFA*), zero-inflated negative binomial based wanted variation extraction (ZINB-WaVE; R package *zinbwave*, v1.6.0), probabilistic count matrix factorization (pCMF; R package pCMF, v1.0.0), deep count autoencoder network (DCA; Python package *dca*), and a scalable deep-learning-based approach (scScope; Python package *scscope*). An overview of these 11 DR methods with their corresponding modeling characteristics is provided in Table 1.

Assess the performance of dimensionality reduction methods

We evaluated the performance of DR methods by evaluating how effective the low-dimensional components extracted from DR methods are for downstream analysis. We evaluated either of the two commonly applied downstream analysis, clustering analysis and lineage reconstruction analysis, in the 30 data sets described above. In the analysis, we varied the number of low-dimensional components extracted from these DR methods. Specifically, for cell clustering data sets, in a data with less than or equal to 300 cells, we varied the number of low dimensional components to be either 2, 6, 14, or 20. In a data with more than 300 cells, we varied the number of low dimensional components to be either 0.5%, 1%, 2%, or 3% of the total number of cells. For lineage inference data sets, we varied the number of low dimensional components to be either 2, 6, 14, or 20 for all data sets, since common lineage inference methods prefer a relatively small number of components.

For clustering analysis, after DR with these DR methods, we used two different clustering methods, the hierarchical clustering (R function *hclust*; stats v3.5.3) and *k*-means clustering (R function *kmeans*; stats v3.6.0), to perform clustering on the reduced feature space. The *k*-means clustering is a key ingredient of commonly applied scRNAseq clustering methods such as SC3 [18] and Waterfall [24] while the hierarchical clustering is a key ingredient of commonly applied scRNAseq clustering methods such as CIDR [17] and CHETAH [50]. In both these clustering methods, we set the number of clusters *k* to be the known number of cell types in the data. We compared the cell clusters inferred using the low dimensional components to the true cell cluster and evaluated clustering accuracy by two criteria: the adjusted rand index (ARI) [51] and the normalized mutual information (NMI) [52]. The ARI and NMI are defined as:

$$ARI(P,T) = \frac{\sum_{l,s}\binom{n_{ls}}{2} - \left(\sum_l \binom{n_l}{2}\sum_s\binom{n_s}{2}\right)/\binom{n}{2}}{\left(\sum_l\binom{n_l}{2}+\sum_s\binom{n_s}{2}\right)/\binom{n}{2} - \left(\sum_l\binom{n_l}{2}\sum_s\binom{n_s}{2}\right)/\binom{n}{2}} \text{ and } NMI(P,T) = \frac{2MI(P,T)}{H(P)+H(T)},$$

where $P = (p_1, p_1, \cdots, p_n)^T$ denotes the inferred cell-type cluster labels from clustering analysis while $T = (t_1, t_1, \cdots, t_n)^T$ denotes the known true cell-type labels for n samples in the data; l and s enumerate the clusters, with $l, s = 1, \cdots, k$; $n_l = \sum_l I(p_i = l)$ is the number of cells that belong to cluster *l* in the inferred cluster labeling, with $I(\cdot)$ being an indicator function; $n_s = \sum_s I(t_i = s)$ is the number of cells that belong to cluster *s* in the true cluster labeling; and $n_{ls} = \sum_{l,s} I(p_i = l) I(t_i = s)$ is the number of times where the *i*'th cell belongs to the cluster l in the inferred cluster labeling and belongs to the cluster s in the true cluster labeling; note that n_{ls} effectively measures the number of cells that are in common between P and T; $MI(P,T) = \sum_l \sum_s \frac{n_{ls}}{n} log\left(\frac{\frac{n_{ls}}{n}}{\frac{n_s n_l}{n}}\right)$ is the mutual information between two cluster labels; and $H(T) = \sum_s \frac{n_s}{n} log\left(\frac{n_s}{n}\right)$ is the entropy function for true cell-type labeling. We used the *compare* function in the *igraph* R package (v1.0.0) to compute both ARI and NMI criteria. For each data set, we repeated the above procedure five times and report the averaged results to avoid the influence of the stochasticity embedded in some DR methods and/or the clustering algorithm.

For trajectory inference, after DR with these DR methods, we used *Slingshot* [38] (R package, v1.2.0), which is the recommend lineage inference method based on a recent comparative study [14]. The *Slingshot* software takes two input data: the low-dimensional components extracted from DR methods and a vector of cluster labels predicted by clustering algorithms. For the later, we used either *k*-means or hierarchical clustering algorithm on the extracted low-dimensional components to obtain cluster labels. After obtaining the two types of input through the *slingshot* function, we used the *getLineages* function to fit a minimum spanning tree (MST) to identify lineage. The final output from *Slingshot* is an object of class *SlingshotDataSet* that contains the inferred lineage information. We follow the original *Slingshot* paper [38] to evaluate the accuracy of the inferred lineage using the Kendall rank correlation coefficient. To do so, for each data, we first ranked genes based on their position on the true lineage. We ordered all *m* genes based on this rank order and denoted the corresponding rank in ascending order for these genes as $\{x_1, \cdots, x_m\}$, where $x_i \leq x_{i+1}$. Note that the true lineage is linear without any bifurcation or multifurcation patterns, while the inferred lineage may contain multiple ending points in addition to the single starting point. Therefore, for each inferred lineage, we examined one trajectory at a time, where each trajectory consists of the starting point and one of the ending points. In each trajectory, we ranked genes in order based on their position in the trajectory. We denote the corresponding rank order in the inferred trajectory for all *m* genes as $\{y_1, \cdots, y_m\}$, where we set y_l as missing if *l*'th gene is not included in the inferred trajectory. For each pair of non-missing genes, we labeled the gene pair (*i*, *j*) as a concordant pair if their relative rank in the inferred lineage are consistent with their relative rank in the true lineage; that is, either $(x_i \geq x_j \,\&\, y_i \geq y_j)$ or $(x_i < x_j \,\&\, y_i < y_j)$. Otherwise, we labeled the gene pair (*i*, *j*) as discordant. We denoted C as the number of concordant pairs, D as the number of discordant pairs, and U as the total number of non-missing genes. The Kendell correlation coefficient is then computed as

$$\tau = \frac{C - D}{U(U-1)/2}.$$

Afterwards, we obtained the maximum absolute τ over all these trajectories as the final Kendall correlation score to evaluate the similarity between the inferred lineage and the true lineage. For each data set, we repeated the above procedure five times and report the averaged results to avoid the influence of the stochasticity embedded in some DR methods and/or the lineage inference algorithm.

Finally, we investigated the stability and robustness of different DR methods in both cell clustering or lineage inference applications through data splitting. Here, we focused on two representative scRNAseq data sets, the *Kumar* data set for cell clustering and the *Hayashi* data set for lineage inference. For each data, we randomly split the data into two subsets with an equal number of cells in each cell type in the two subsets. We repeated the split procedure 10 times to capture the potential stochasticity during the data split. In each split replicate, we applied different DR methods to analyze each subset separately. We used *k*-means clustering algorithm to infer the clustering labels in each subset. We used NMI to measure cell clustering accuracy and used Kendall correlation to measure lineage inference accuracy.

FUNDING

This study was supported by the National Institutes of Health (NIH) Grants R01HG009124 and R01GM126553, and the National Science Foundation (NSF) Grant DMS1712933. This project has also been made possible in part by grant number 2018-181314 from the Chan Zuckerberg Initiative DAF, an advised fund of Silicon Valley Community Foundation. S.S. was supported by NIH Grant R01HD088558 (PI Tung), Top International University Visiting Program for Outstanding Young Scholars of Northwestern Polytechnical University, the Fundamental Research Funds for the Central Universities (Grant No. 3102017OQD098), and the Natural Science Foundation of Shaanxi Province (Grant No. 2019JQ127). J.Z. was supported by NIH Grant U01HL137182 (PI Kang).

AVAILABILITY OF DATA AND MATERIALS

All source code and data sets used in our experiments have been deposited at www.xzlab.org/reproduce.html.

AUTHORS’ CONTRIBUTIONS

XZ conceived the idea and provided funding support. SS and XZ designed the experiments. SS adapted software, performed simulations and analyzed real data. JQ and YM collected data sets and helped interpreting results. SS and XZ wrote the manuscript with input from all other authors.

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

No ethnical approval was required for this study. All utilized public data sets were generated by other organizations that obtained ethical approval.

CONSENT FOR PUBLICATION

Not applicable.

COMPETING INTERESTS

The authors declare that they have no competing interests.

REFERENCE

1. Picelli S, Faridani OR, Bjorklund AK, Winberg G, Sagasser S, Sandberg R: **Full-length RNA-seq from single cells using Smart-seq2.** *Nature Protocols* 2014, **9:**171-181.
2. Xi Chen, Sarah A. Teichmann, Meyer KB: **From Tissues to Cell Types and Back: Single-Cell Gene Expression Analysis of Tissue Architecture.** *Annual Review of Biomedical Data Science* 2018, **1:**29-51.
3. Ziegenhain C, Vieth B, Parekh S, Reinius B, Guillaumet-Adkins A, Smets M, Leonhardt H, Heyn H, Hellmann I, Enard W: **Comparative Analysis of Single-Cell RNA Sequencing Methods.** *Molecular Cell* 2017, **65:**631-643.
4. Buettner F, Natarajan KN, Casale FP, Proserpio V, Scialdone A, Theis FJ, Teichmann SA, Marioni JC, Stegie O: **Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells.** *Nature Biotechnology* 2015, **33:**155-160.
5. McDavid A, Finak G, Gottardo R: **The contribution of cell cycle to heterogeneity in single-cell RNA-seq data.** *Nature Biotechnology* 2016, **34:**591-593.
6. Li HP, Courtois ET, Sengupta D, Tan YL, Chen KH, Goh JJL, Kong SL, Chua C, Hon LK, Tan WS, et al: **Reference component analysis of single-cell transcriptomes elucidates cellular heterogeneity in human colorectal tumors.** *Nature Genetics* 2017, **49:**708-718.
7. Patel AP, Tirosh I, Trombetta JJ, Shalek AK, Gillespie SM, Wakimoto H, Cahill DP, Nahed BV, Curry WT, Martuza RL, et al: **Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma.** *Science* 2014, **344:**1396-1401.
8. Rozenblatt-Rosen O, Stubbington MJT, Regev A, Teichmann SA: **The Human Cell Atlas: from vision to reality.** *Nature* 2017, **550:**451-453.
9. Stegle O, Teichmann SA, Marioni JC: **Computational and analytical challenges in single-cell transcriptomics.** *Nature Reviews Genetics* 2015, **16:**133-145.
10. Altman N, Krzywinski M: **The curse(s) of dimensionality.** *Nature Methods* 2018, **15:**399-400.
11. Tenenbaum JB, de Silva V, Langford JC: **A global geometric framework for nonlinear dimensionality reduction.** *Science* 2000, **290:**2319-2323.
12. Duo A, Robinson MD, Soneson C: **A systematic performance evaluation of clustering methods for single-cell RNA-seq data.** *F1000Res* 2018, **7:**1141.
13. Kiselev VY, Andrews TS, Hemberg M: **Challenges in unsupervised clustering of single-cell RNA-seq data.** *Nature Reviews Genetics* 2019, **20:**273–282.

14. Saelens W, Cannoodt R, Todorov H, Saeys Y: **A comparison of single-cell trajectory inference methods.** *Nat Biotechnology* 2019, **20:**547-554.
15. Zappia L, Phipson B, Oshlack A: **Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database.** *Plos Computational Biology* 2018, **14:**1006245.
16. Butler A, Hoffman P, Smibert P, Papalexi E, Satija R: **Integrating single-cell transcriptomic data across different conditions, technologies, and species.** *Nature Biotechnology* 2018, **36:**411-420.
17. Lin PJ, Troup M, Ho JWK: **CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data.** *Genome Biology* 2017, **18:**59.
18. Kiselev VY, Kirschner K, Schaub MT, Andrews T, Yiu A, Chandra T, Natarajan KN, Reik W, Barahona M, Green AR, Hemberg M: **SC3: consensus clustering of single-cell RNA-seq data.** *Nature Methods* 2017, **14:**483-486.
19. Zhu LX, Lei J, Klei L, Devlin B, Roeder K: **Semisoft clustering of single-cell data.** *Proceedings Of the National Academy Of Sciences Of the United States Of America* 2019, **116:**466-471.
20. Angerer P, Haghverdi L, Buttner M, Theis FJ, Marr C, Buettner F: **destiny: diffusion maps for large-scale single cell data in R.** *Bioinformatics* 2016, **32:**1241-1243.
21. Senabouth A, Lukowski SW, Hernandez JA, Andersen S, Mei X, Nguyen QH, Powell JE: **ascend: R package for analysis of single cell RNA-seq data.** *BioRxiv* 2017.
22. Way GP, Greene CS: **Bayesian deep learning for single-cell analysis.** *Nature Methods* 2018, **15:**1009-1010.
23. Ji ZC, Ji HK: **TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis.** *Nucleic Acids Research* 2016, **44:**e117.
24. Shin J, Berg DA, Zhu YH, Shin JY, Song J, Bonaguidi MA, Enikolopov G, Nauen DW, Christian KM, Ming GL, Song HJ: **Single-Cell RNA-Seq with Waterfall Reveals Molecular Cascades underlying Adult Neurogenesis.** *Cell Stem Cell* 2015, **17:**360-372.
25. Welch JD, Hartemink AJ, Prins JF: **SLICER: inferring branched, nonlinear cellular trajectories from single cell RNA-seq data.** *Genome Biology* 2016, **17:**106.
26. Trapnell C, Cacchiarelli D, Grimsby J, Pokharel P, Li SQ, Morse M, Lennon NJ, Livak KJ, Mikkelsen TS, Rinn JL: **The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells.** *Nature Biotechnology* 2014, **32** 381-386.
27. Setty M, Tadmor MD, Reich-Zeliger S, Ange O, Salame TM, Kathail P, Choi K, Bendall S, Friedman N, Pe'er D: **Wishbone identifies bifurcating**

developmental trajectories from single-cell data. *Nature Biotechnology* 2016, **34:**637-645.

28. Pierson E, Yau C: **ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis.** *Genome Biology* 2015, **16:**241.
29. Durif G, Modolo L, Mold JE, Lambert-Lacroix S, Picard F: **Probabilistic Count Matrix Factorization for Single Cell Expression Data Analysis.** *Bioinformatics* 2019, **10812:**btz177.
30. Risso D, Perraudeau F, Gribkova S, Dudoit S, Vert JP: **A general and flexible method for signal extraction from single-cell RNA-seq data.** *Nature Communications* 2018, **9:**284
31. Zheng GXY, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, Ziraldo SB, Wheeler TD, McDermott GP, Zhu JJ, et al: **Massively parallel digital transcriptional profiling of single cells.** *Nature Communications* 2017, **8:**14049
32. Regev A, Teichmann SA, Lander ES, Amt I, Benoist C, Birney E, Bodenmiller B, Campbell P, Carninci P, Clatworthy M, et al: **The Human Cell Atlas.** *Elife* 2017, **6**.
33. Adlung L, Amit I: **From the Human Cell Atlas to dynamic immune maps in human disease.** *Nature Reviews Immunology* 2018, **18:**597-598.
34. Rashid S, Shah S, Bar-Joseph Z, Pandya R: **Dhaka: Variational Autoencoder for Unmasking Tumor Heterogeneity from Single Cell Genomic Data.** *Bioinformatics* 2019**:**btz095.
35. Deng Y, Bao F, Dai QH, Wu LF, Altschuler SJ: **Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning.** *Nature Methods* 2019, **16:**311-314.
36. Wang DF, Gu J: **VASC: Dimension Reduction and Visualization of Single-cell RNA-seq Data by Deep Variational Autoencoder.** *Genomics Proteomics & Bioinformatics* 2018, **16:**320-331.
37. Eraslan G, Simon LM, Mircea M, Mueller NS, Theis FJ: **Single-cell RNA-seq denoising using a deep count autoencoder.** *Nature Communications* 2019, **10:**390
38. Street K, Risso D, Fletcher RB, Das D, Ngai J, Yosef N, Purdom E, Dudoit S: **Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics.** *Bmc Genomics* 2018, **19:**477.
39. van der Maaten L, Hinton G: **Visualizing Data using t-SNE.** *Journal Of Machine Learning Research* 2008, **9:**2579-2605.
40. Becht E, McInnes L, Healy J, Dutertre CA, Kwok IWH, Ng LG, Ginhoux F, Newell EW: **Dimensionality reduction for visualizing single-cell data using UMAP.** *Nature Biotechnology* 2019, **37:**38-44.

41. Chen MJ, Zhou X: **Controlling for Confounding Effects in Single Cell RNA Sequencing Studies Using both Control and Target Genes.** *Scientific Reports* 2017, **7:**13587
42. Stuart T, Satija R: **Integrative single-cell analysis.** *Nature Reviews Genetics* 2019, **20:**257-272.
43. Argelaguet R, Velten B, Arnol D, Dietrich S, Zenz T, Marioni JC, Buettner F, Huber W, Stegle O: **Multi-Omics Factor Analysis-a framework for unsupervised integration of multi-omics data sets.** *Molecular Systems Biology* 2018, **14:**e8124.
44. Newman AM, Liu CL, Green MR, Gentles AJ, Feng WG, Xu Y, Hoang CD, Diehn M, Alizadeh AA: **Robust enumeration of cell subsets from tissue expression profiles.** *Nature Methods* 2015, **12:**453-457.
45. Mohammadi S, Zuckerman N, Goldsmith A, Grama A: **A Critical Survey of Deconvolution Methods for Separating Cell Types in Complex Tissues.** *Proceedings Of the IEEE* 2017, **105:**340-366.
46. Ilicic T, Kim JK, Kolodziejczyk AA, Bagger FO, McCarthy DJ, Marioni JC, Teichmann SA: **Classification of low quality cells from single-cell RNA-seq data.** *Genome Biology* 2016, **17:**29.
47. Wagner F, Yanai I: **Moana: A robust and scalable cell type classification framework for single-cell RNA-Seq data.** *BioRxiv* 2018.
48. Yip SH, Sham PC, Wang J: **Evaluation of tools for highly variable gene discovery from single-cell RNA-seq data.** *Brief Bioinformatics* 2018**:**bby011.
49. Andrews TS, Hemberg M: **M3Drop: Dropout-based feature selection for scRNASeq.** *Bioinformatics* 2018**:**bty1044.
50. Kanter JKd, Lijnzaad P, Candelli T, Margaritis T, Holstege FCP: **CHETAH: a selective, hierarchical cell type identification method for single-cell RNA sequencing.** *BioRxiv* 2019.
51. Hubert L, Arabie P: **Comparing Partitions.** *Journal Of Classification* 1985, **2:**193-218.
52. Studholme C, Hill DLG, Hawkes DJ: **An overlap invariant entropy measure of 3D medical image alignment.** *Pattern Recognition* 1999, **32:**71-86.
53. I.T. J: *Principal Component Analysis.* Springer; 2002.
54. Stone JV: *Independent component analysis: a tutorial introduction.* Cambridge, Massachusetts: MIT 2014.
55. Bartholomew DJ, Steele F, Galbraith J, Moustaki I: *Analysis of Multivariate Social Science Data.* Taylor & Francis; 2008.
56. Lee DD, Seung HS: **Learning the parts of objects by non-negative matrix factorization.** *Nature* 1999, **401:**788-791.

57. Coifman RR, Lafon S, Lee AB, Maggioni M, Nadler B, Warner F, Zucker SW: **Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps.** *Proceedings Of the National Academy Of Sciences Of the United States Of America* 2005, **102:**7426-7431.

Figure 1. Overview of the evaluation workflow for dimensionality reduction methods. We obtained a total of 28 publicly available scRNAseq data from GEO and 10x Genomics website. We also simulated two addition simulation data sets. For each of the 30 data sets in turn, we applied 11 dimensionality reduction (DR) methods to extract the low-dimensional components. Afterwards, we evaluated the performance of DR methods by evaluating how effective the low-dimensional components extracted from DR methods are for downstream analysis. We did so by evaluating the two commonly applied downstream analysis: clustering analysis and lineage reconstruction analysis. In the analysis, we varied the number of low-dimensional components extracted from these DR methods. The performance of each DR method is qualified by normalized mutual information (NMI) and adjusted rand index (ARI) for cell clustering analysis and Kendall correlation coefficient for trajectory inference. We also recorded the stability of each DR method across data splits and recorded the computation time for each DR method. Through the comprehensive evaluation, we eventually provide practical guidelines for practitioners to choose DR methods for scRNAseq data analysis.

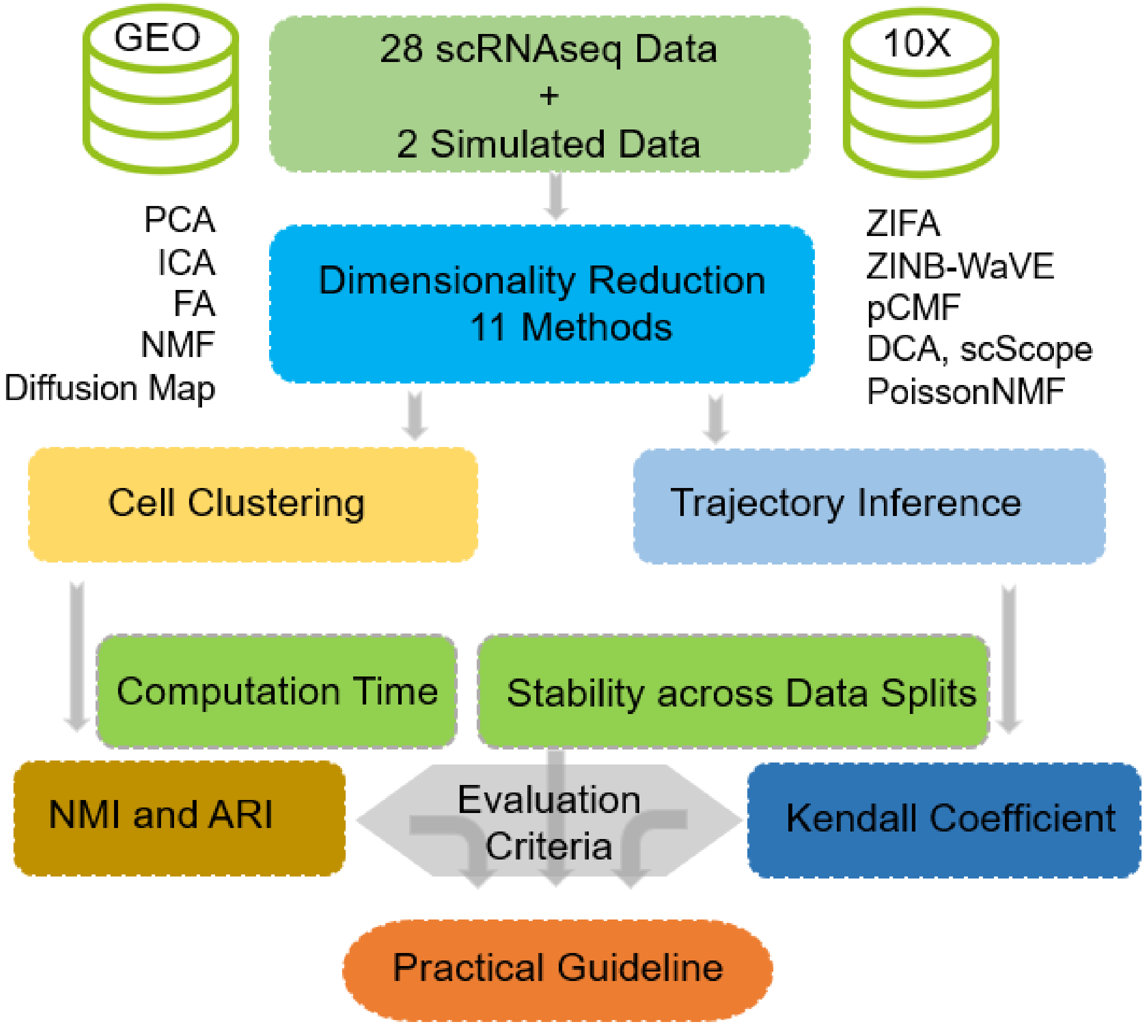


Figure 2. DR method performance evaluated by *k*-means clustering based on NMI in downstream cell clustering analysis. We compared 11 DR methods (columns), including factor analysis (FA), principal component analysis (PCA), independent component analysis (ICA), Diffusion Map, nonnegative matrix factorization (NMF), Poisson NMF, zero-inflated factor analysis (ZIFA), zero-inflated negative binomial based wanted variation extraction (ZINB-WaVE), probabilistic count matrix factorization (pCMF), deep count autoencoder network (DCA), and scScope. We evaluated their performance on 14 real scRNAseq data sets and 2 simulated data sets (rows). The simulated data based on Kumar data is labeled with #. The performance of each DR method is measured by normalized mutual information (NMI). For each data set, we compared the four different number of low-dimensional components. The four numbers equal to 0.5%, 1%, 2%, and 3% of the total number of cells in big data and equal to 2, 6, 14, and 20 in small data (which are labeled with *). For convenience, we only listed 0.5%, 1%, 2%, and 3% on x-axis. No results for ICA are shown in the Pancreatic data (grey fills) because ICA cannot handle the large number of features in that data.

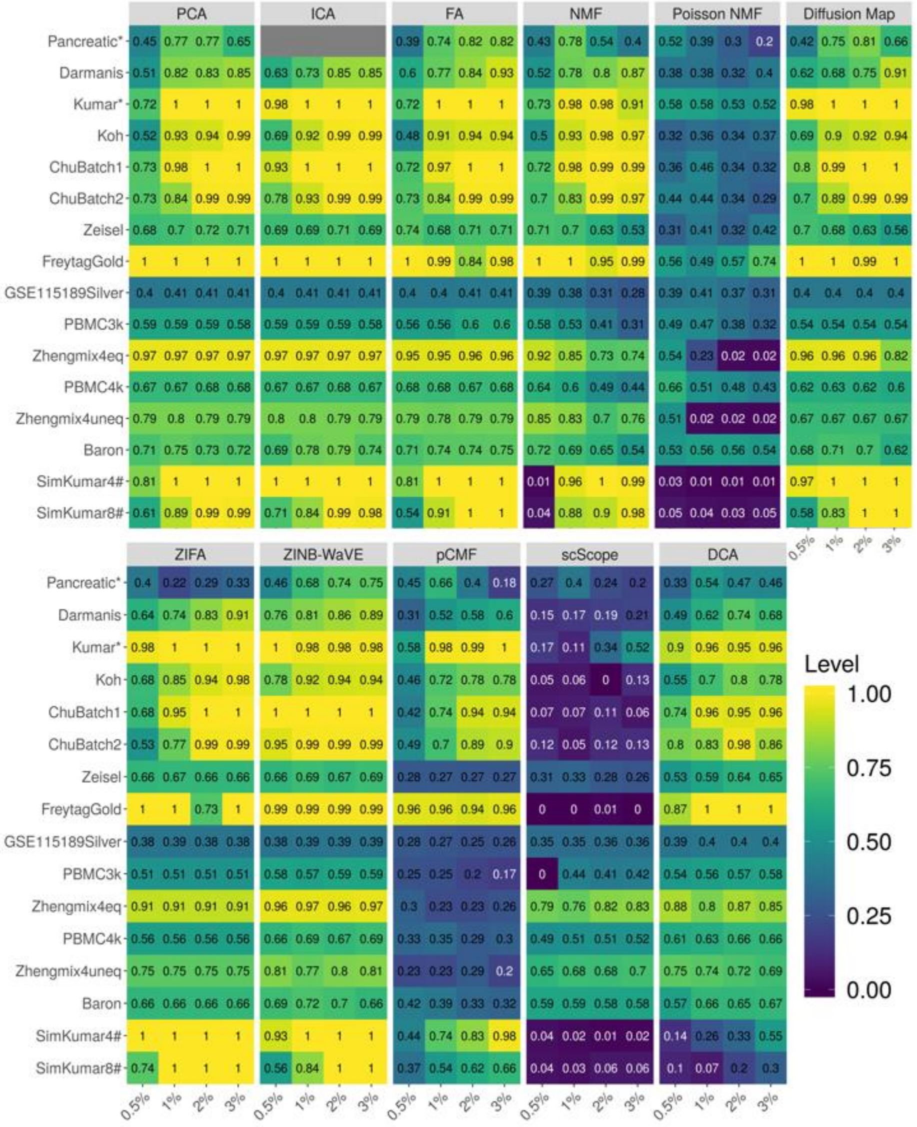


Figure 3. DR method performance evaluated by Kendal correlation in the downstream trajectory inference analysis. We compared 10 DR methods (columns), including factor analysis (FA), principal component analysis (PCA), independent component analysis (ICA), Diffusion Map, nonnegative matrix factorization (NMF), Poisson NMF, zero-inflated factor analysis (ZIFA), zero-inflated negative binomial based wanted variation extraction (ZINB-WaVE), probabilistic count matrix factorization (pCMF), and deep count autoencoder network (DCA). We evaluated their performance on 14 real scRNAseq data sets (rows) in terms of lineage inference accuracy. We used *Slingshot* with *k*-means as the initial step for lineage inference. The performance of each DR method is measured by Kendall correlation. For each data set, we compared four different number of low-dimensional components (2, 6, 14, and 20; four sub-columns under each column). Grey fills in the table represents missing results where *Slingshot* gave out errors when we supplied the extracted low-dimensional components from the corresponding DR method.

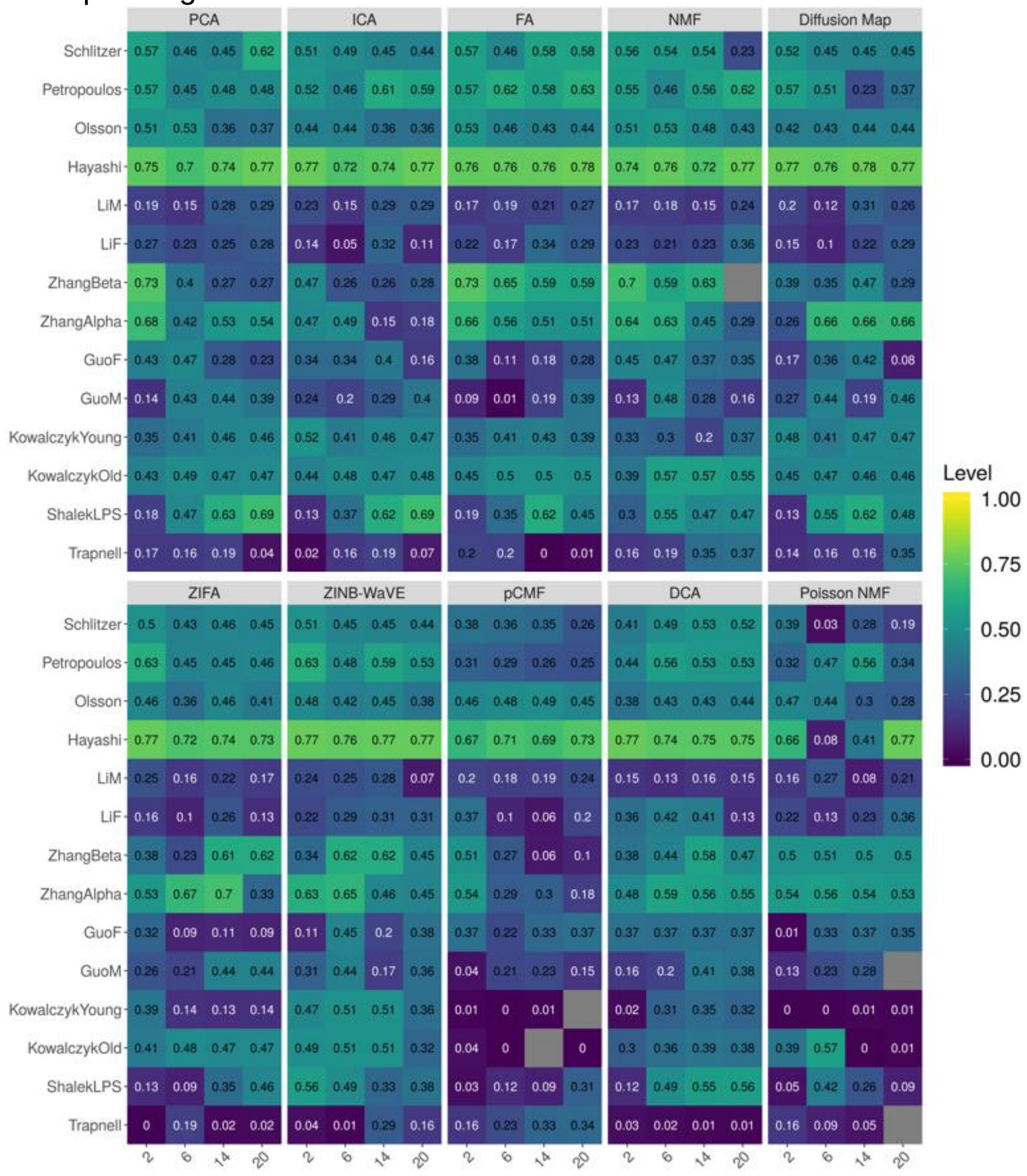


Figure 4. The computation time (in hours) for different DR methods. We recorded computing time for 11 DR methods on simulated data sets with varying number of low-dimensional components and varying number of sample sizes. (**A**) Computation time for different DR methods (y-axis) changes with respect to an increasing number of low-dimensional components (x-axis). The number of cells is fixed to be 500 and the number of genes is fixed to be 10,000 in this set of simulations. Three methods (ZINB-WaVE, pCMF, and ZIFA) become noticeably computationally more expensive than the remaining methods with increasing number of low-dimensional components. (**B**) Computation time for different DR methods (y-axis) changes with respect to an increasing sample size (i.e. the number of cells) in the data. The number of low-dimensional components is fixed to be 22 in this set of simulations. Four methods (ZIFA, pCMF, ZINB-WaVE and ICA) become noticeably computationally more expensive than the remaining methods with increasing number of cells in the data. Computing time is recorded on a single thread of an Intel Xeon E5-2683 2.00 GHz processor. Note that some methods are implemented with parallelization capability (e.g. ZINB-WaVE and pCMF) though we tested them on a single thread for fair comparison across methods. Compared DR methods include: factor analysis (FA; light green), principal component analysis (PCA; light blue), independent component analysis (ICA; blue), Diffusion Map (red), nonnegative matrix factorization (NMF; green), Poisson NMF(pink), zero-inflated factor analysis (ZIFA; light orange), zero-inflated negative binomial based wanted variation extraction (ZINB-WaVE; orange), probabilistic count matrix factorization (pCMF; light purple), deep count autoencoder network (DCA; yellow), and scScope (purple).

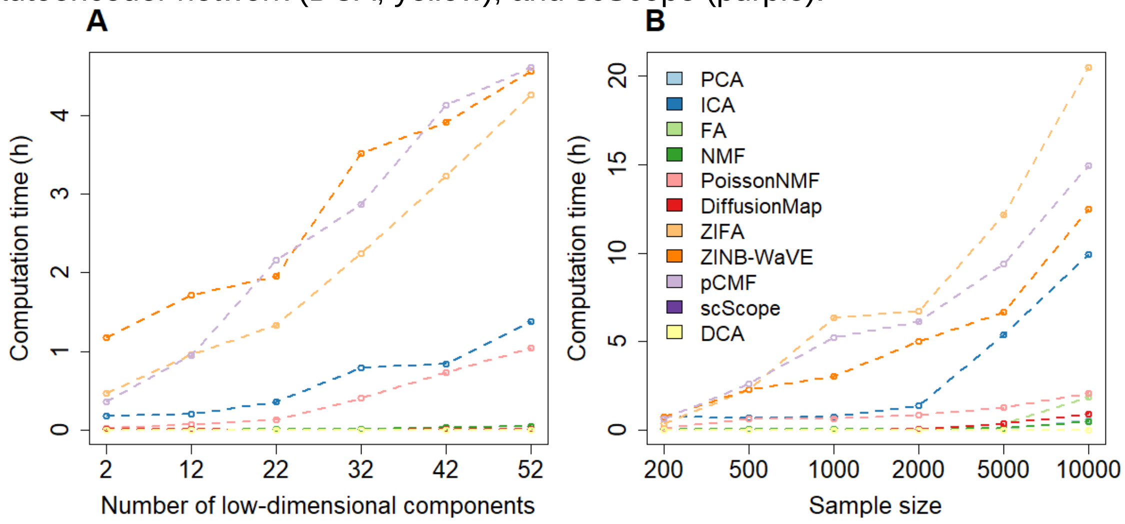
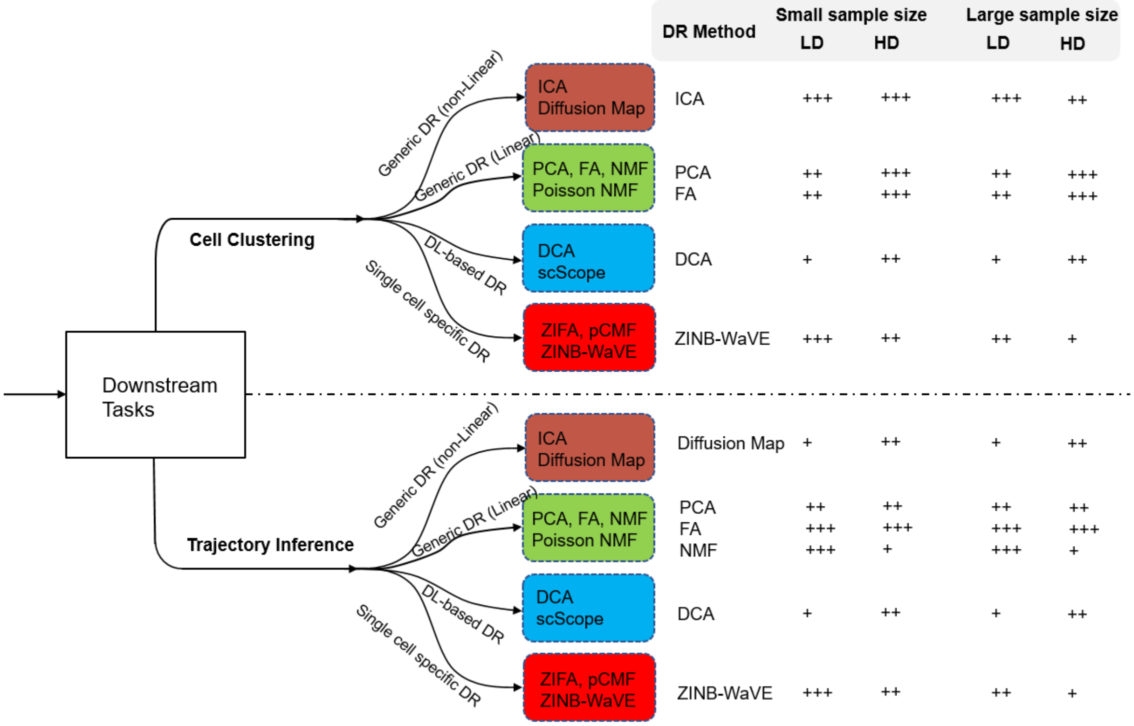


Figure 5. Practical guideline for choosing DR methods in scRNAseq analysis. We categorized the 11 DR methods into four groups: generic DR with linear projection; generic DR with non-linear projection; deep learning (LD) based DR methods; and single cell specific DR methods which aim to model counts and/or dropout events in scRNAseq. Compared DR methods include: factor analysis (FA), principal component analysis (PCA), independent component analysis (ICA), Diffusion Map, nonnegative matrix factorization (NMF), Poisson NMF, zero-inflated factor analysis (ZIFA), zero-inflated negative binomial based wanted variation extraction (ZINB-WaVE), probabilistic count matrix factorization (pCMF), deep count autoencoder network (DCA), and scScope. +: recommendation index, where a higher number represents higher recommendation. LD represents low number of low-dimensional components; HD represents high number of low-dimensional components.



DR Method	Small sample size		Large sample size	
	LD	HD	LD	HD
ICA	+++	+++	+++	++
PCA	++	+++	++	+++
FA	++	+++	++	+++
DCA	+	++	+	++
ZINB-WaVE	+++	++	++	+
Diffusion Map	+	++	+	++
PCA	++	++	++	++
FA	+++	+++	+++	+++
NMF	+++	+	+++	+
DCA	+	++	+	++
ZINB-WaVE	+++	++	++	+

Table 1. A list of compared dimensionality reduction methods. Each dimensionality reduction method has a unique set of strengths and weaknesses. FA: factor analysis; PCA: principal component analysis; ICA: independent component analysis; NMF: nonnegative matrix factorization; Poisson NMF: Kullback-Leibler divergence-based NMF; ZIFA: zero-inflated factor analysis; ZINB-WaVE: zero-inflated negative binomial based wanted variation extraction; pCMF: probabilistic count matrix factorization; DCA: deep count autoencoder network.

No.	**Methods**	**Modeling Counts**	**Modeling Zero Inflation**	**Non-Linear Projection**	**Computation Efficiency**	**Implementation Language**	**Year of Publication**	**Reference**
1	**PCA**	No	No	No	Yes	R, C++, Python, MATLAB or others	1901	[53]
2	**ICA**	No	No	Yes	No	R, C++, Python, MATLAB or others	1994	[54]
3	**FA**	No	No	No	Yes	R, C++, Python, MATLAB or others	1952	[55]
4	**NMF**	No	No	No	Yes	R, C++, Python, MATLAB or others	1999	[56]
5	**Poisson NMF**	Yes	No	No	No	R, C++, Python, MATLAB or others	1999	[56]
6	**Diffusion Map**	No	No	Yes	Yes	R, C++, Python, MATLAB or others	2005	[57]
7	**ZIFA**	No	Yes	No	No	Python	2016	[28]
8	**ZINB-WaVE**	Yes	Yes	No	No	R	2018	[30]
9	**pCMF**	Yes	Yes	No	No	R	2019	[29]

10	**scScope**	No	Yes	Yes	Yes	Python	2019	[35]
11	**DCA**	Yes	No	Yes	Yes	Python	2018	[37]