nature machine intelligence

SUPPLEMENTARY INFORMATION

https://doi.org/10.1038/s42256-019-0037-0

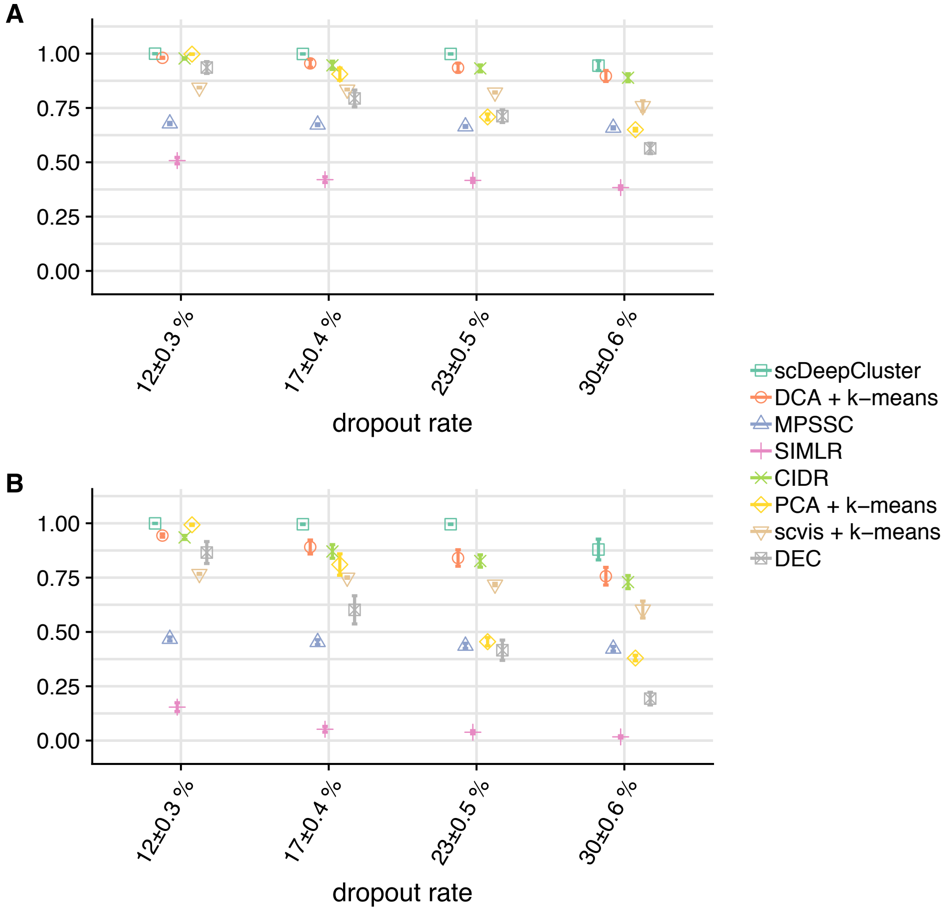
In the format provided by the authors and unedited.

Clustering single-cell RNA-seq data with a model-based deep learning approach

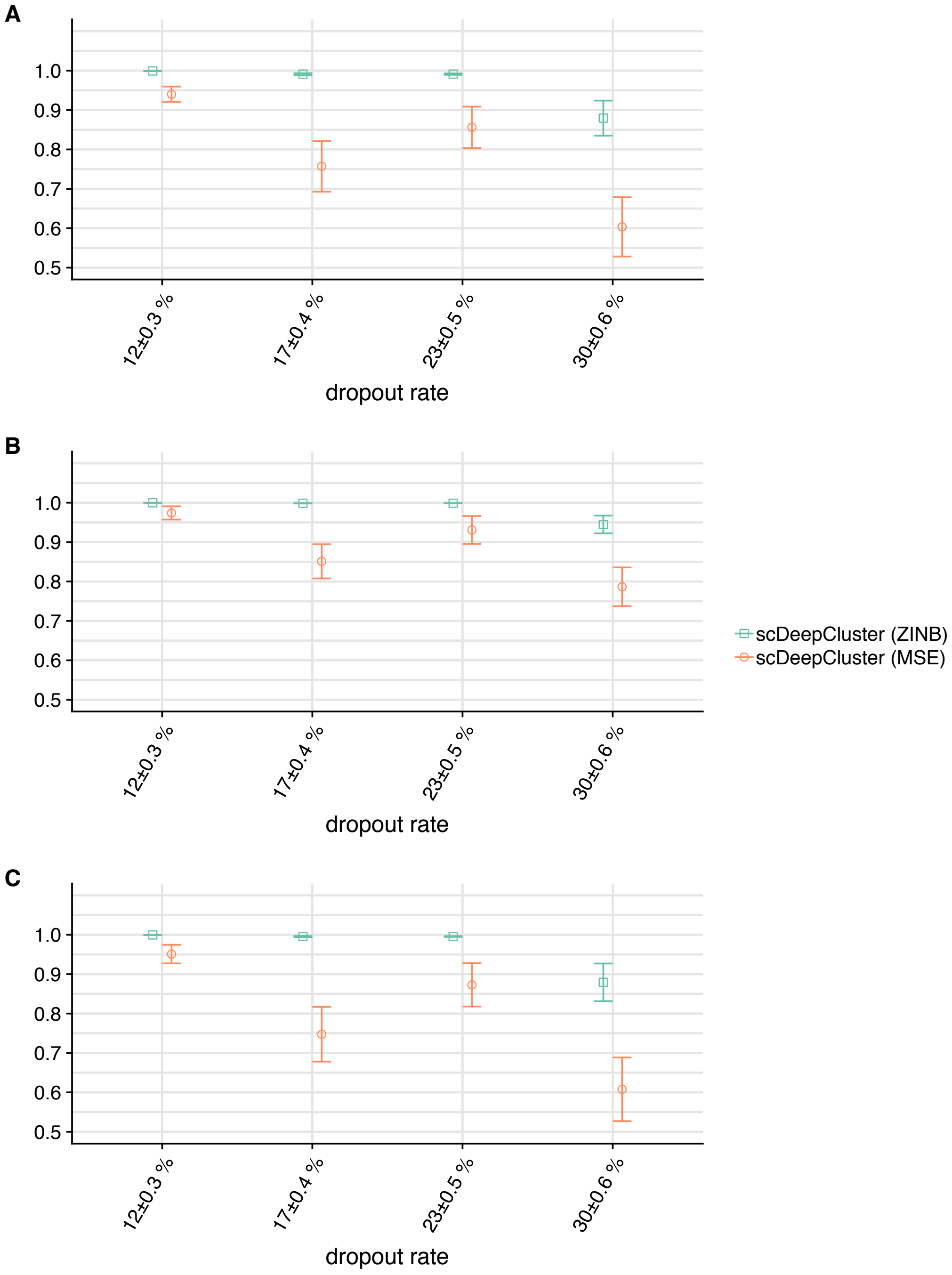
Tian Tian[1,3], Ji Wan[2,3], Qi Song[2] and Zhi Wei[1*]

[1]Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA. [2]CuraCloud Corporation, Seattle, WA, USA. [3]Theses authors contributed equally: Tian Tian, Ji Wan. *e-mail: zhiwei@njit.edu

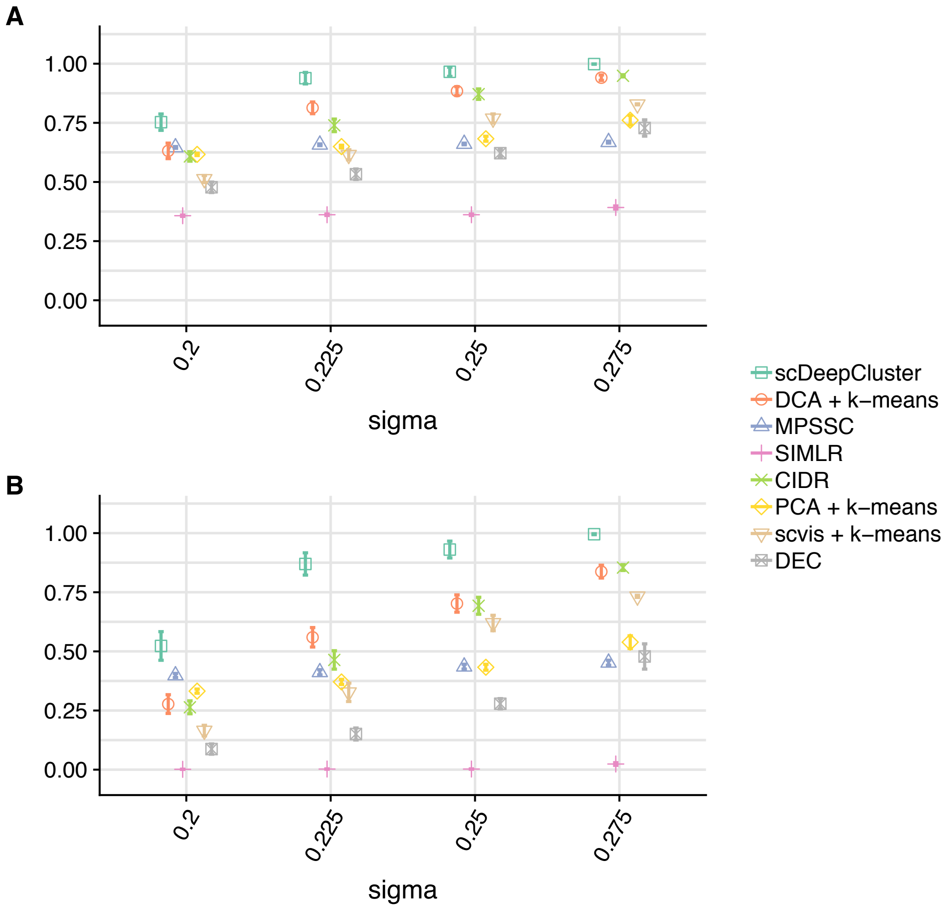
Supplementary Figures



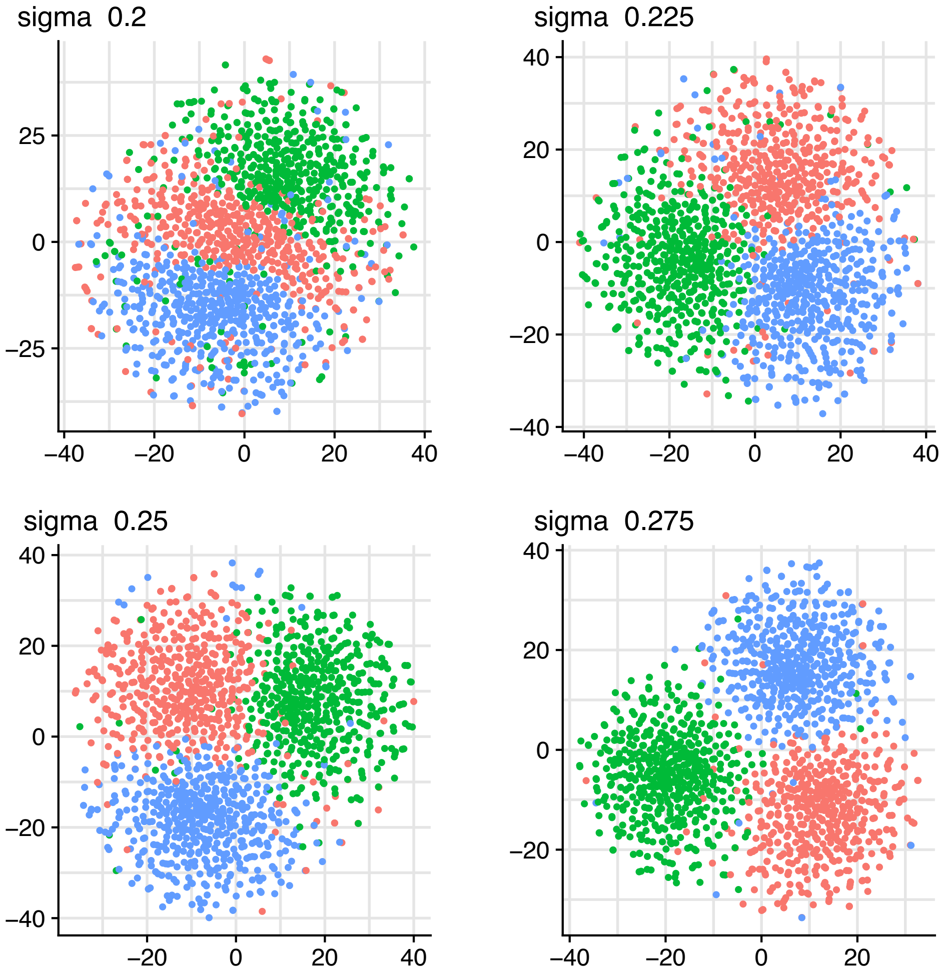
Supplementary Figure 1. Clustering performance of scDeepCluster, DCA + k-means, MPSSC, SIMLR, CIDR, PCA + k-means, scvis + k-means and DEC on simulated data measured by **(A)** Clustering Accuracy (CA) and **(B)** Adjust Rand Index (ARI). The averaged values over 20 repeats with standard errors are shown for each simulation setting. The larger value means the more concordance between predicted labels and true labels.



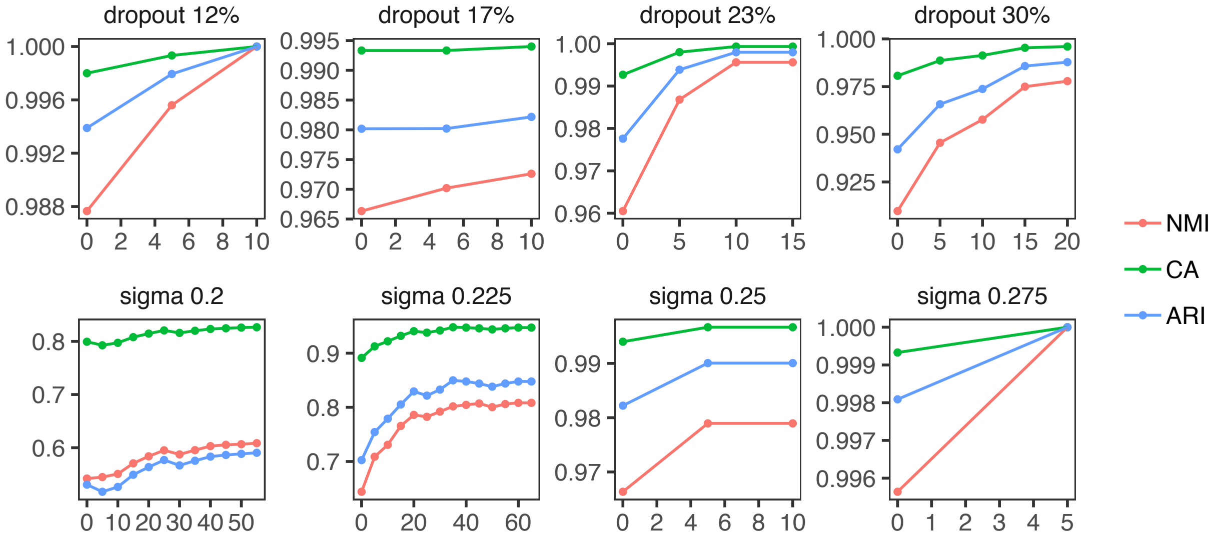
Supplementary Figure 2. Clustering performance of scDeepCluster with ZINB loss and MSE loss on simulated data with various dropout rates measured by **(A)** NMI, **(B)** CA and **(C)** ARI. The averaged values over 20 repeats with standard errors are shown for each simulation setting. The larger value means more concordance between the predicted labels and the true labels.



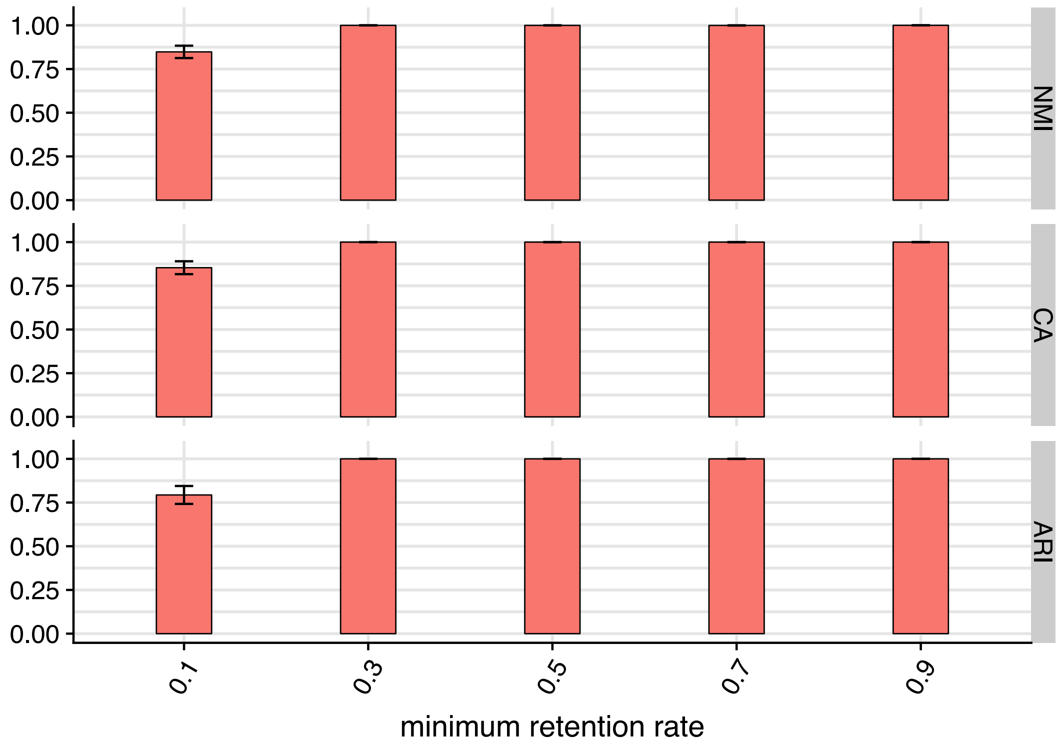
Supplementary Figure 3. Clustering performance of scDeepCluster, DCA + k-means, MPSSC, SIMLR, CIDR, PCA + k-means, scvis + k-means and DEC on simulated data measured by **(A)** Clustering Accuracy (CA) and **(B)** Adjust Rand Index (ARI). The averaged dropout rates are 17% (*dropout.shape* = -1, *dropout.mid* = 0). The averaged values over 20 repeats with standard errors are shown for each simulation setting. The larger value means the more concordance between predicted labels and true labels.



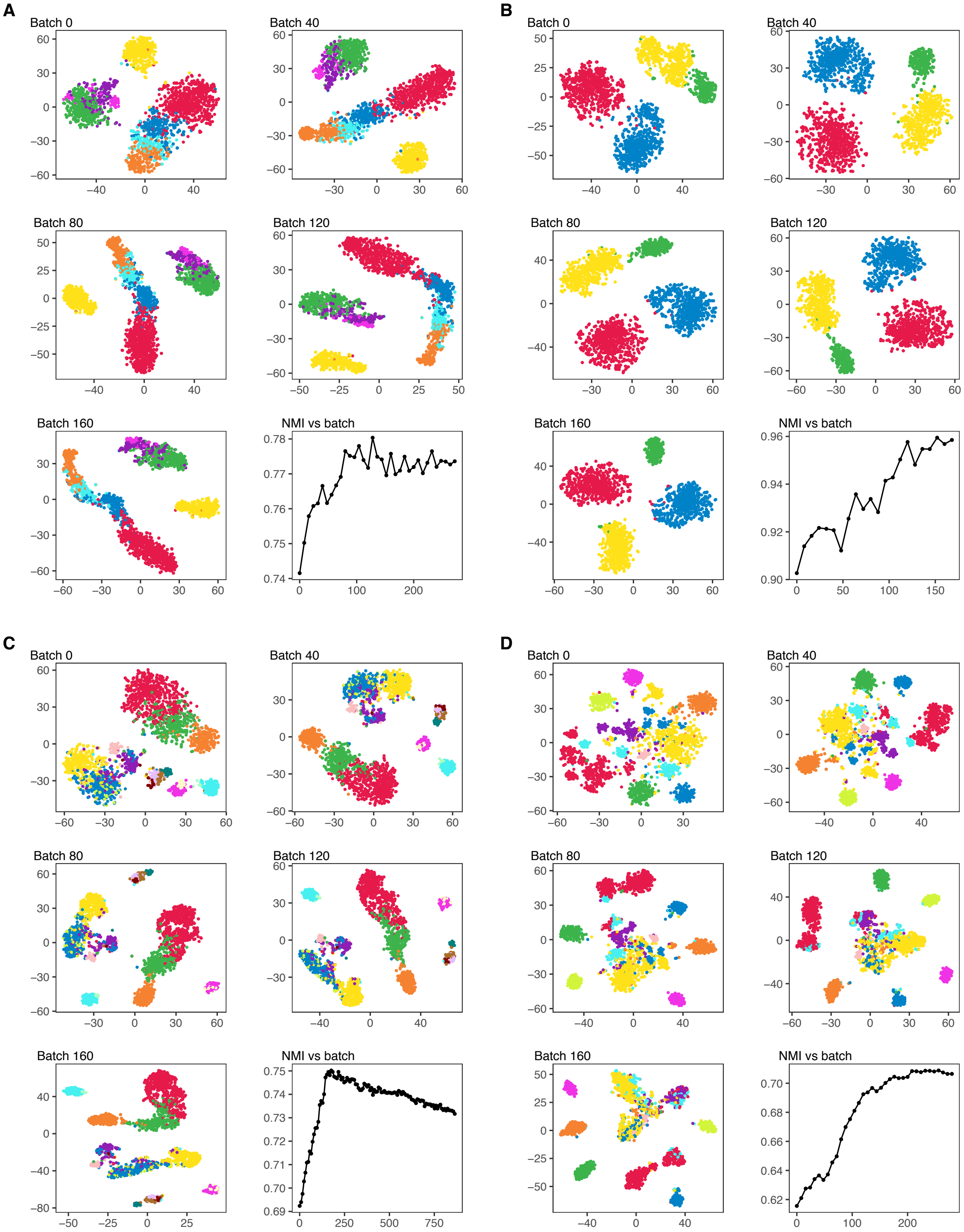
Supplementary Figure 4. The t-SNE plots of simulated data with the different settings of sigma of the log-normal distribution (*de.facScale* parameter in Splatter). One example from each simulation setting is displayed. Distinct colors represent different true labels. The inputs for t-SNE plots are the library size normalized and log-transformed read count matrix.



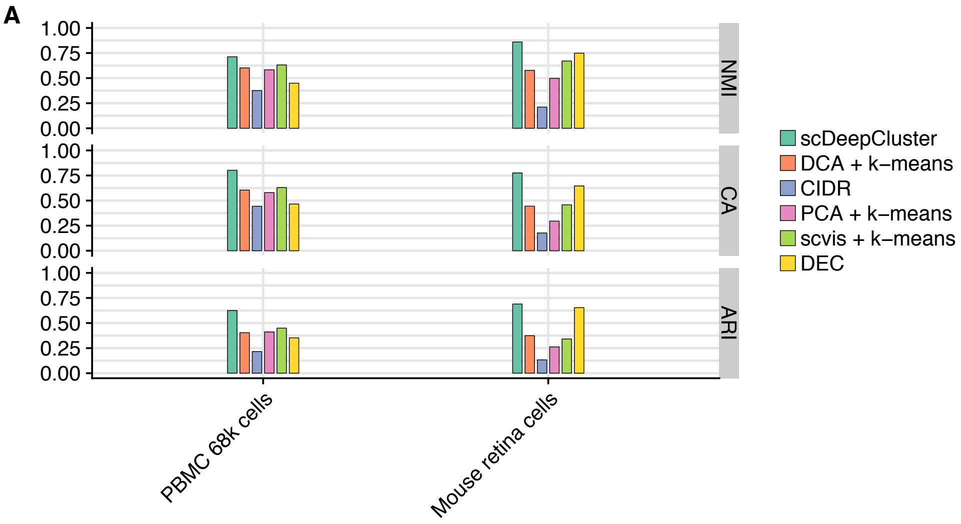
Supplementary Figure 5. Clustering performance per batch of scDeepCluster after k-means initialization on the embedded spaces. One example from each simulation setting is displayed. The x-axes are the batches, and the y-axes are the values of NMI, CA and ARI. We observe the clustering performance improves after k-means initialization, which illustrates the contribution of the clustering loss.

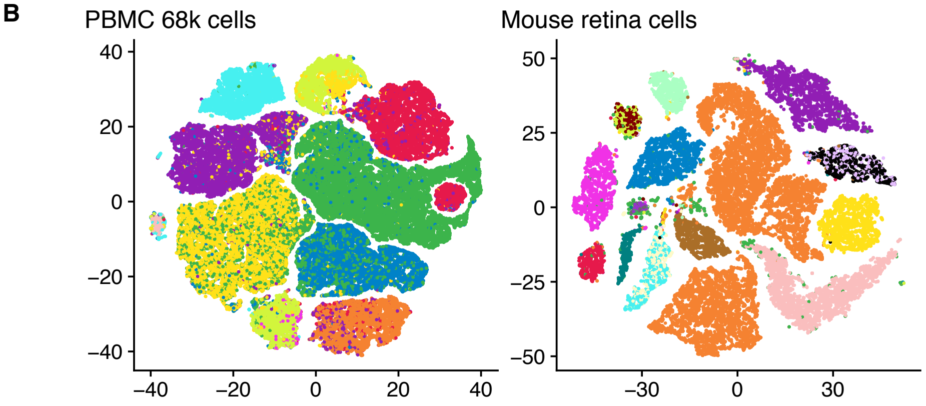


Supplementary Figure 6. Clustering performance of scDeepCluster measured by NMI, CA and ARI on imbalanced simulated scRNA-seq data. The averaged values and standard errors on the 20 datasets of each simulation setting are shown. The larger value means more concordance between the predicted labels and the true labels. The minimum retention rate is a measurement of imbalance levels (*Supplementary Notes "Imbalanced groups"*).

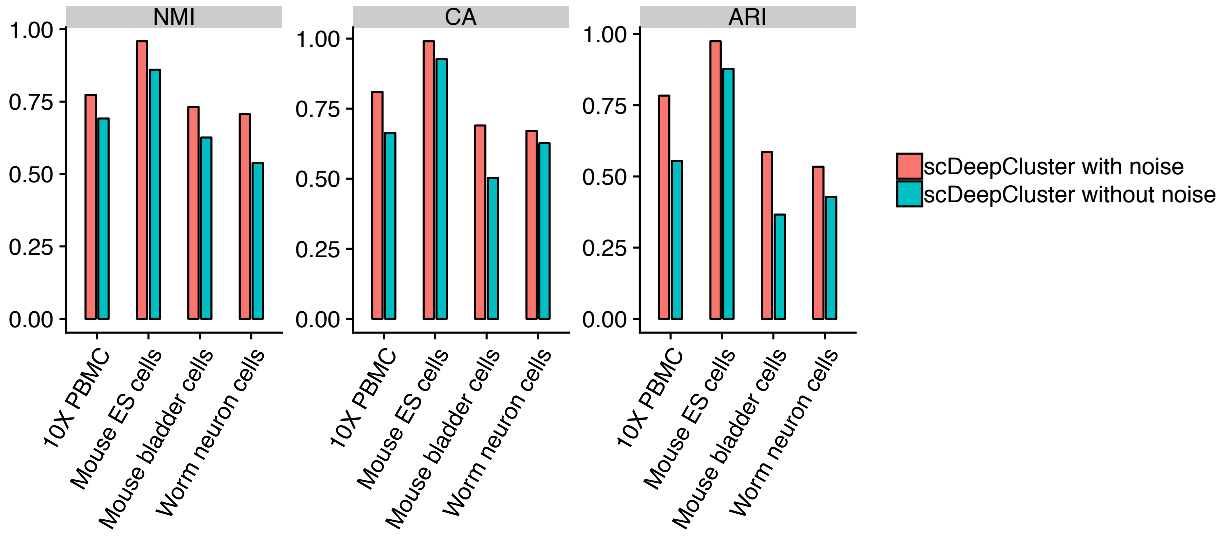


Supplementary Figure 7. Visualization of latent representations during the clustering stage on the 2100 cells subsets of **(A)** 10X PBMC, **(B)** Mouse ES cells, **(C)** Mouse bladder cells and **(D)** Worm neuron cells. Different colors mark different groups of cells. The NMIs per batch are also shown.

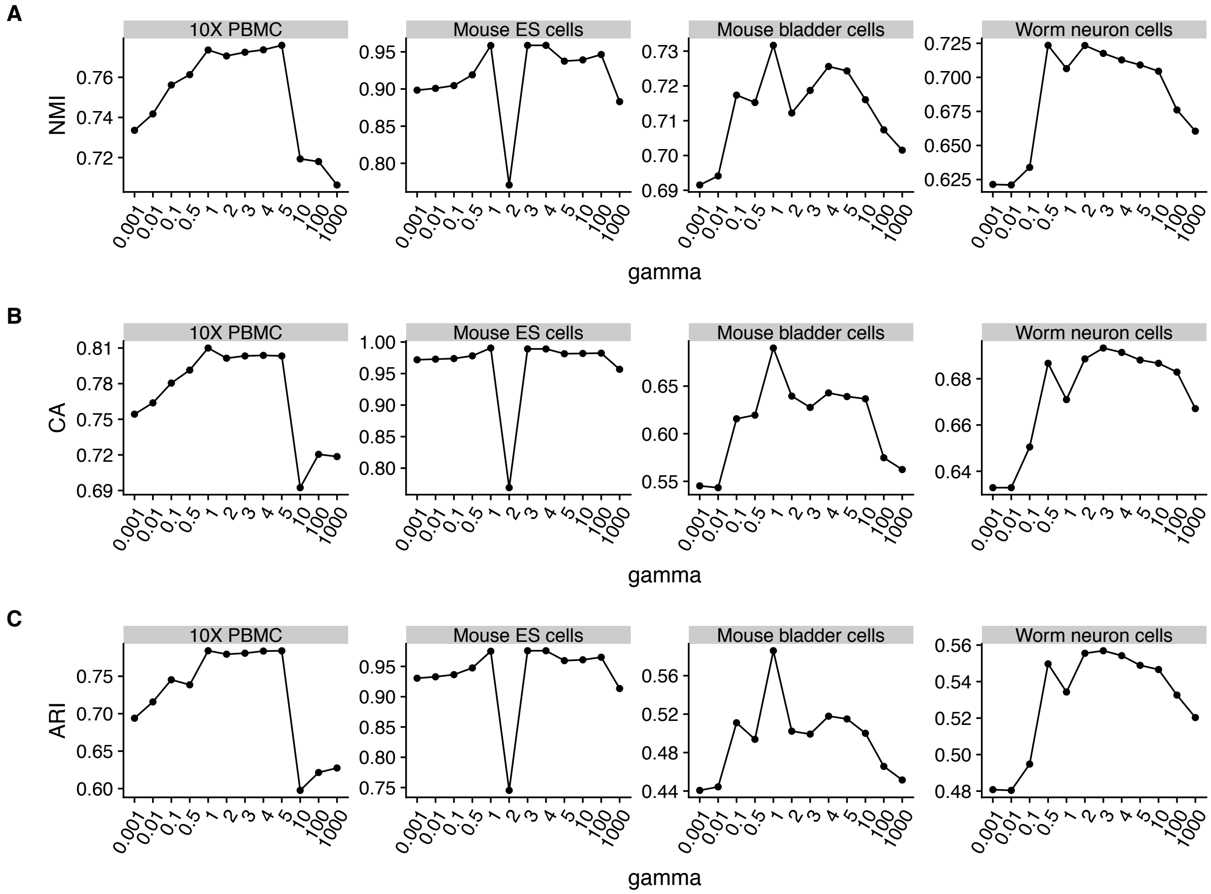




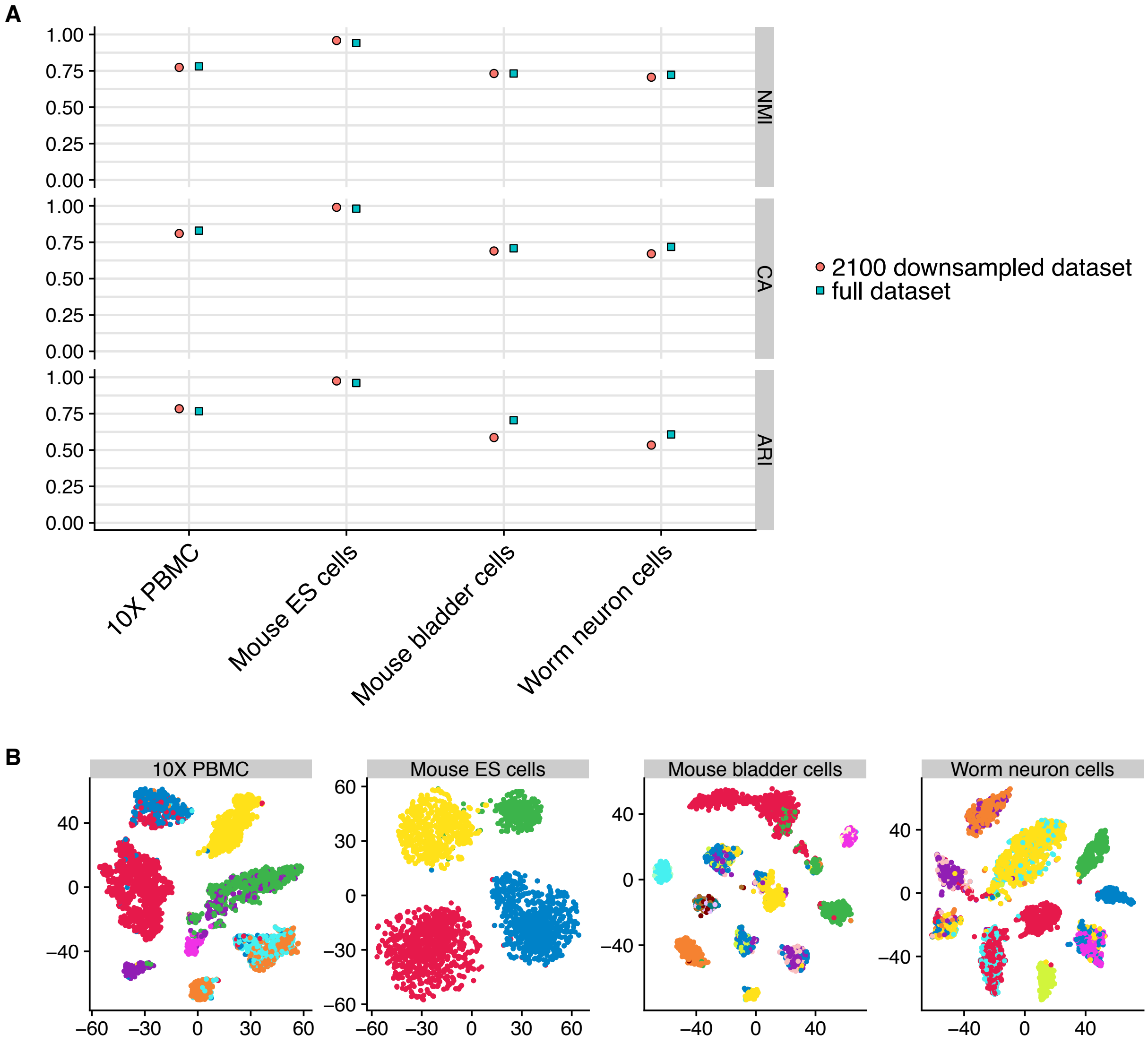
Supplementary Figure 8. Clustering performance of scDeepCluster evaluated on two large real datasets (*Supplementary Notes "Description of the large real datasets"*). **(A)** Cluster results measured by NMI, CA and ARI (The performance for CIDR on PBMC 68k is the results on the down-sampled 20k cells). **(B)** Visualization of the final latent spaces of scDeepCluster on the PBMC 68k and mouse retina cells datasets. The distinct colors of the points represent the true labels.



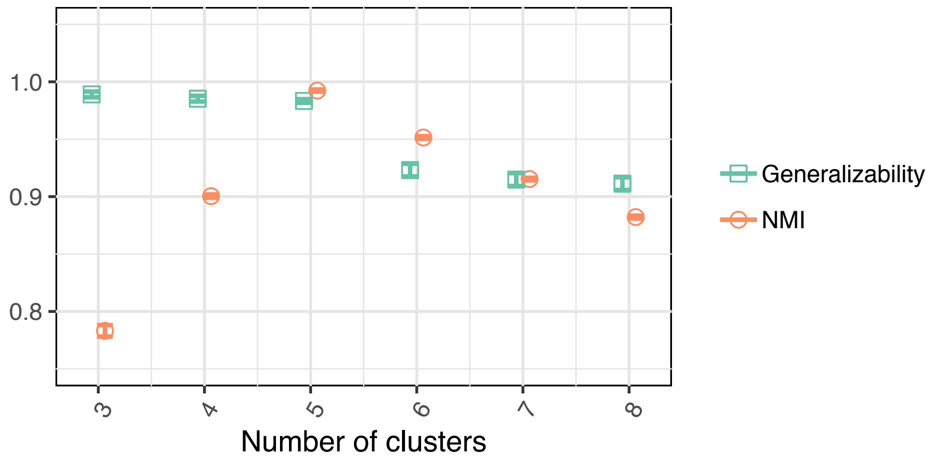
Supplementary Figure 9. The denoising autoencoder helps to learn a more robust feature representation(1, 2). Clustering performance of scDeepCluster with random Gaussian noise and without random Gaussian noise on the 2100 cells subsets of 10X PBMC, mouse ES cells, mouse bladder cells, and worm neuron cells is displayed.



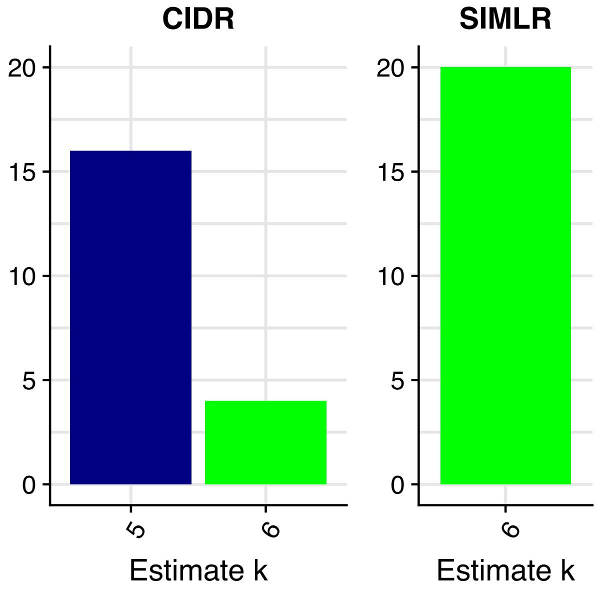
Supplementary Figure 10. The effect of the clustering coefficient γ on the performance for the 2100 cells subsets of 10X PBMC, mouse ES cells, mouse bladder cells, and worm neuron cells.



Supplementary Figure 11. The clustering performance of scDeepCluster on the 2100 cells down-sampled and the full real scRNA-seq datasets. **(A)** The clustering metrics (NMI, CA and ARI) of scDeepCluster on the 2100 cells down-sampled and the full datasets. **(B)** The t-SNE visualization of the final latent spaces of scDeepCluster on the full real scRNA-seq datasets. The clustering performance is robust to down-sampling.



Supplementary Figure 12 Estimate of the number of clusters on the 20 simulated datasets (*Supplementary Notes "Number of clusters"*). A plot of Generalizability and NMI of validation data on the various number of clusters. The means and standard errors among 20 simulated data are plotted.



Supplementary Figure 13 Estimate of the number of clusters (k) by CIDR and SIMLR on the 20 simulated datasets. The x-axis is the values of estimated k, and the y-axis is the counts of the estimations.

Supplementary Table

Dataset	Sequencing Platform	Sample size / Cell Numbers	#Genes	#Groups
PBMC 68k	10X	68,579	20,387	10
Mouse retina cells	Drop-seq	27,499	13,166	19

Supplementary Table 1 Summary of 2 large real datasets

Supplementary Notes

Imbalanced groups

The imbalanced data was reported to hinder the performance machine learning algorithms (3). Imbalance means numbers of objects in groups are not similar. To study the effect of imbalanced groups, we investigated the performance of scDeepCluster when the three groups have imbalanced sample sizes. Following (4), we generated different imbalance levels by varying minimum retention rate r_{min} when allocating the 1500 cells to the three groups (1500 cells by 2500 genes in each dataset, 1500 cells were assigned into 3 groups, the parameters for simulation are *dropout.shape* = -1, *dropout.mid* = 0, and *de.facScale* = 0.4, which are the default values recommended by Splatter). For minimum retention rate r_{min}, data points of group 0 will be kept with relative proportion r_{min} and group 2 with relative proportion 1, with the group 1 with the relative proportion of the middle: $(r_{min} + 1)/2$. So, the resulting the largest group will have $1/r_{min}$ times as large as the minimum one. Each minimum retention rate generates 20 datasets. As shown in *Supplementary Figure 6*, scDeepCluster's performance is fairly robust. It retains perfect clustering concordance (NMI ≈ 1) until r_{min} decreases to 0.3, its performance remains decent (NMI > 0.85) even when the imbalance is extreme ($r_{min} \; = \; 0.1$).

Description of the large real datasets

We evaluate the performance of scDeepCluster on 2 large real datasets (more than 10k cells): PBMC 68k(5) (https://github.com/10XGenomics/single-cell-3prime-paper) and mouse retina cells(6) (https://scrnaseq-public-datasets.s3.amazonaws.com/scater-objects/shekhar.rds). The number of cells and clusters are summarized in *Supplementary Table 1*. We report the clustering results in *Supplementary Figure 8*. We failed to run CIDR on the PBMC 68k cells even with 141G memory, so we down-sampled the dataset to 20k cells for CIDR only (other methods on PBMC 68k use the full dataset).

Number of clusters

We use an intuitive metric generalizability (*G*)(4) to estimate the number of clusters (*k*). Formally, *G* is defined as the ratio between training and validation clustering loss (we split the data to training and validation data):

$$G = \frac{L_{train}}{L_{validation}}$$

We modify the scDeepCluster model, so it has no reconstruction loss during clustering stage. As a result, *G* is the ratio between the clustering loss of training data and validation data. We first train scDeepCluster model (only clustering loss in clustering stage) on training data. Next, we calculate the clustering loss, predict the labels of validation data (by using the model trained on training data) and calculate *G*.

To illustrate how *G* works, we did a simulation experiment. We use Splatter to simulate 20 scRNA-seq datasets which have 5000 cells of 2000 genes in 5 groups. The *de.fracScale* = 0.3, and other parameters were set to be default (e.g. *dropout.shape* = -1, *dropout.mid* = 0). For each simulated data, we split training and validation by the ratio of 9:1. We pre-train 600 epochs for each training data (pre-train the reconstruction loss of denoising ZINB model-based autoencoder), then the pre-trained autoencoder weights are used to initialize the model in clustering stage with various settings of the number of clusters (*k* = 3 - 8 clusters).

As reported in *Supplementary Figure 12*, we observe a sharp drop in generalizability when the cluster number increases from 5 to 6, which suggests that 5 is the optimal number of clusters. We also observe that the NMI values for validation data are highest at 5. This result suggests that generalizability is a good metric to determine the number of clusters.

We also summarized the results of *k* (number of clusters) estimation of the competing methods: CIDR and SIMLR (*Supplementary Figure 13*). As we can see, CIDR estimates k correctly (k = 5) in most cases but makes mistakes in some cases, but SIMLR fails to estimate the correct k in all cases.

Supplementary Reference

1. Vincent P, Larochelle H, Bengio Y, Manzagol P-A. Extracting and composing robust features with denoising autoencoders. Proceedings of the 25th international conference on Machine learning; Helsinki, Finland. 1390294: ACM; 2008. p. 1096-103.
2. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. J Mach Learn Res. 2010;11(Dec):3371-408.
3. Japkowicz N, Stephen S. The class imbalance problem: A systematic study. Intelligent Data Analysis. 2002;6(5):429-49.
4. Xie J, Girshick R, Farhadi A. Unsupervised Deep Embedding for Clustering Analysis. Proceedings of Machine Learning Research; New York, NY, USA: PMLR; 2016. p. 478-87.
5. Zheng GX, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, et al. Massively parallel digital transcriptional profiling of single cells. Nat Commun. 2017;8:14049.
6. Shekhar K, Lapan SW, Whitney IE, Tran NM, Macosko EZ, Kowalczyk M, et al. Comprehensive Classification of Retinal Bipolar Neurons by Single-Cell Transcriptomics. Cell. 2016;166(5):1308-23 e30.