Reliable and efficient gene expression recovery in single-cell transcriptomes using DISC

Yao He[1#], Hao Yuan[1#], Cheng Wu[1#], Zhi Xie[1*]

[1]State Key Laboratory of Ophthalmology, Zhongshan Ophthalmic Center, Sun Yat-sen University, Guangzhou, China. [#] Equally contributed. [*] email: xiezhi@gmail.com

Abstract

Single-cell RNA sequencing (scRNA-seq) measures transcriptome at single cell resolution and reveals cell heterogeneity and diversity. Imputation is a common approach to recover false zero expression due to dropout events. We developed DISC, a novel deep learning model with semi-supervised learning (SSL) to accurately impute dropouts. SSL enables DISC to learn structure of genes and cells from sparse data efficiently. DISC deals with ultra-large datasets containing millions of cells and requires just a portion of computational cost and RAM that other deep learning approaches need. We demonstrated that DISC consistently outperformed state-of-the-art approaches using various imputation performance metrics, including accuracy of dropout recovery, recapture of expression distribution, enhancement of gene-gene and cell-cell relationship, improvement of cell type identification and applicability to datasets generated from different platforms with different dropout levels. Its applicability, scalability and reliability make DISC a promising approach to recover gene expression, enhance gene and cell structures and improve cell type identification for large-scale scRNA-seq data.

Keywords

Single cell, transcriptome, deep learning, semi-supervised learning, imputation

Introduction

Single-cell RNA sequencing (scRNA-seq) measures transcriptomes at single cell resolution and is widely used to reveal cell heterogeneity and diversity. The major challenge of analyzing scRNA-seq data is excess false zero expressions, named dropouts, which distort gene expression distribution and cause misclassification of cell types[1]. The recent advances in droplet-or combinatorial indexing- based sequencing technologies have dramatically increased the throughput from thousands to over a million of cells in a single experiment, causing severer dropout problems due to shallow sequencing depth per cell[2-4].

Imputation is a common approach to recover dropout events and has been shown to improve the gene expression structure and other downstream analysis[5]. Most imputation approaches are model-based, such as SAVER, DrImpute and scImpute, that borrow information across cells to predict expression values for missing genes[6-8]. Another related approach is “smoothing” that removes the high frequency signals, including technical noise and dropouts[9]. More recently, deep learning-based approaches have been developed to address the scalability issue since conventional approaches are unable to process large datasets[5]. For example, scVI, scScope and DCA use deep autoencoder (AE) to learn feature representation to recover dropouts[10-12].

Although true expression of genes in each cell can be estimated by learning gene-gene relationship cross cells, it is challenging because more than 90% of genes in scRNA-seq data are zero-counts. While the true and dropout zeros are difficult to distinguish, genes in each cell with detected expression (positive-count genes) are more reliable measurements compared to zeros (zero-count genes). Semi-supervised learning (SSL) approach offers promise when a few labels are available by allowing models to supplement their training with unlabeled data[13]. We hypothesize that SSL can build a reliable imputation algorithm by learning information from both positive- and zero-count genes, which can be treated as labeled and unlabeled data, respectively.

Herein, we present DISC, a Deep-learning Imputation model with semi-supervised learning for Single Cell transcriptomes. DISC integrates an AE and a recurrent neural network (RNN) and uses SSL to train model parameters. We set up a number of evaluations to assess DISC's performance. Compared DISC with the other state-of-the-art imputation approaches, including SAVER, MAGIC and three deep-learning based approaches, DCA, scVI and scScope, we showed that DISC consistently achieved top performance. DISC's accuracy, efficiency and scalability make it a reliable imputation approach for scRNA-seq transcriptome, particularly for ultra-large datasets. DISC was implemented in Python and publicly available at https://github.com/xie-lab/DISC.

Results

Description of DISC

DISC has an integrative structure of an AE and an RNN (Figure 1A). AE is a part of RNN that performs dimension reduction while preserving the manifold of the original data. For each step *t*, encoder of AE projects the high dimensional cell expression profile (x^t) into a low dimensional latent representation (z^t). Then, the latent representation is used to predict the cell expression profile through a predictor matrix and to explore the data manifold through the reconstruction of the expression profile by the decoder of AE, obtaining expression profiles from multiple steps either predicted by the predictor (y^t) or reconstructed by the decoder of AE ($\widehat{y^t}$) (Figure S1). Expression profile by the predictor is feed to the next step as an input. At the end, a soft attention framework computes weighted average of y^t as the imputation result and weighted average of $\widehat{y^t}$ as the reconstruction result to support SSL.

Users do not need to specify parameters in the model. Parameters in the layers are automatically learned from data through back-propagation using SSL (Figure 1B and Methods). Imputer learns from the positive-count genes using "noise-to-noise" method[14]. Reconstructor learns using SSL from a combination of positive-count genes and zero-count genes assigned a pseudo-count (pseudo-count genes) by imputer to search the best latent representation to reconstruct the expression profile after imputation. Predictor learns using SSL from a combination of positive-count genes and pseudo-count genes assigned by decoder to search for the best gene expression structure to preserve the manifold learned by AE. This AE-RNN structure enables DISC to learn biological information not only from the small portion of positive-count genes, but also the large portion of zero-count genes.

DISC also provides a solution to compress the latent representation into a lower dimension (50 by default), which retains the most informative information of the expression matrix, for clustering and visualization (Figure 1C). Ultra-large dataset is beyond the capability of existing analytical tools. Using the 50 dimensions as the low dimensional representation of the ultra-large dataset, visualization and clustering can be performed using existing analytical tools with little comprise in performance (Figure S2).

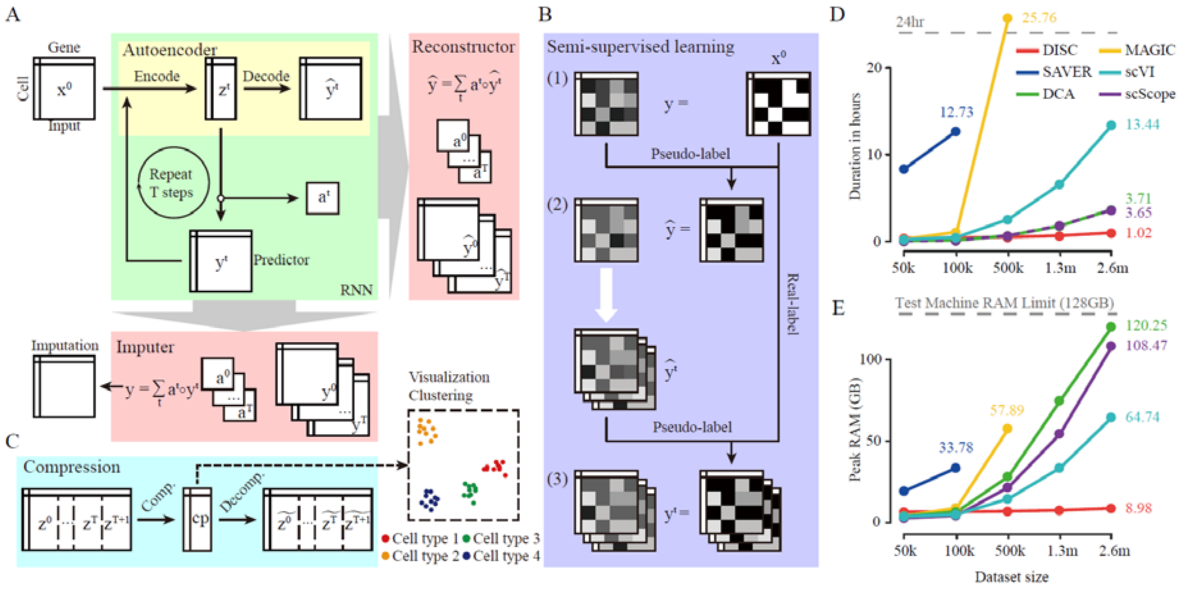

Figure 1. Architecture of DISC and evaluation of computation resource. (**A**) DISC contains an autoencoder, a recursive predictor, an imputer to compute an imputation expression profile and a reconstructor to compute a reconstructed expression profile. (**B**) DISC is trained in a semi-supervised manner: (1) The imputer learns the expression of positive-count genes, (2) The reconstructor learns both the expression of positive-count genes and the pseudo expression of zero-count genes assigned by the imputer, and (3) The predictors learn both the expression of positive-count genes and the pseudo expression of zero-count genes assigned by the decoder of the same step. (**C**) Compression module reduces the large latent representations of multiple steps into a much smaller dimension for visualization and clustering. (**D**) Running time and (**E)** Peak RAM usage for the datasets with different cell numbers.

DISC is scalable to ultra large dataset

For large datasets, loading a complete matrix takes large memory. For example, memory usage is about 100 GB for a matrix with 1,000,000 cells and 10,000 genes. To cope with the large datasets, we designed a novel data reading approach that leverages the ultra-fast chunk reading speed in continuous storage (see Methods). Therefore, DISC needs a constant initial memory before training, but the memory consumption is stable in datasets with increasing data size.

We compared scalability of DISC with the other imputation approaches on speed and memory usage. We used the 1.3 million (m) mouse brain dataset (BRIAN_1.3M) as well as datasets with 50 thousand (k), 100k and 500k down-sampling cells. We also duplicated 1.3m cells to 2.6m cells. All the datasets contained the top 1,000 most variable genes (see Methods). As expected, the deep learning-based approaches were much faster and used much less memory usage (Figure 1D and 1E). For the datasets with 50k and 100k cells, all the approaches had similar performance except SAVER had significantly higher computational costs and memory usage. SAVER failed on 500k dataset due to out of memory. MAGIC was able to complete the 500k dataset but took more than 25 hours and 58 GB memory while four deep-learning approaches took less than 3 hours and less than 25 GB memory. On the 1.3m dataset, only deep-learning approaches could finish the job and DISC (0.92 h) took just around half time of DCA and scScope (1.82 h). On the 2.6m dataset, DISC (1.02 h) took less than 1/3 of time took by DCA and scScope (3.65 and 3.71 h), and 1/13 of scVI (13.44 h). The memory usage of DISC was also considerably less than other approaches. On memory usage, DISC (8.89 GB) took less than 1/7 of scVI (65,74 GB) and less than 1/12 of scScope and DCA (108.47 and 120.25 GB).

A previous study showed that the use of less genes inevitably lost information and increased in gene depth to 10,000 genes improved cell type identification[11]. We therefore tested the imputation performance based on the most 10,000 variable genes and DISC was the only approach that can process 1.3m cells (Figure S3). Overall, DISC offers a scalable solution for datasets with varying sample sizes.

DISC improves gene expression structures validated by FISH

Dropouts severely obscure expression distribution and gene-gene relationship which hinder the downstream analysis[9]. Compared to scRNA-seq, single-cell RNA fluorescence in situ hybridization (FISH) detects a small number of RNA transcripts in single cells and is less suffered from dropouts, which is considered a reliable way to validate expression distribution and gene-gene relationship in single-cell levels[6,15]. To assess DISC's performance to recover lost gene expression structures by dropouts, we compared imputed expression matrix from scRNA-seq to FISH, by Gini coefficient for the gene expression distribution and Fasano and Franceschini's statistics (FF score) for the correlation of gene-gene distributions. Two independent datasets were tested, where both FISH and scRNA-seq measurements are available (see Methods).

Compared the distribution of gene expression across cells on the MELANOMA dataset, DISC recovered expression distribution that resembled the FISH distribution much better than the raw scRNA-seq data (Figure 2A). For all the 19 genes that had both FISH and scRNA-seq measurements, DISC efficiently recovered FISH Gini coefficients (Figure 2B). In addition, DISC improved the correlation of gene-gene distributions (FF score=0.134) than that of raw scRNA-seq data (FF=0.848) (Figure 2C) and the other imputation approaches (Figure S4). Indeed, DISC considerably reduced the FF scores for 75 out of the 81 gene pairs (Figure 2D, $p<2.2e-16$, t-test).

We next compared imputation approaches on both the MELANOMA and SSCORTEX datasets. Compared the scRNA-seq data, all the approaches have improved gene expression distribution (Figure S5**)**, where DISC recovered expression distributions more closely matched to the FISH on the both datasets (Figure 2E). Some approaches, such as scVI, worked well on one but not well on another dataset. In addition, all the approaches improved correlation of gene-gene distributions to FISH compared to that of the raw data (Figure 2F).

DISC again performed the best on MELANOMA (81 gene pairs) and ranked the second on SSCORTEX (528 gene pairs). We also tested correlation of gene co-expression using correlation matrix distance (CMD, see Methods), DISC and SAVER performed well on the both datasets (Figure S6). All the other four approaches, particularly MAGIC and DCA, introduced spurious correlation for gene pairs that had no co-expressed relationships measured by FISH (Figure S7). These results demonstrated that DISC is a reliable model that can recover true gene expression structures.

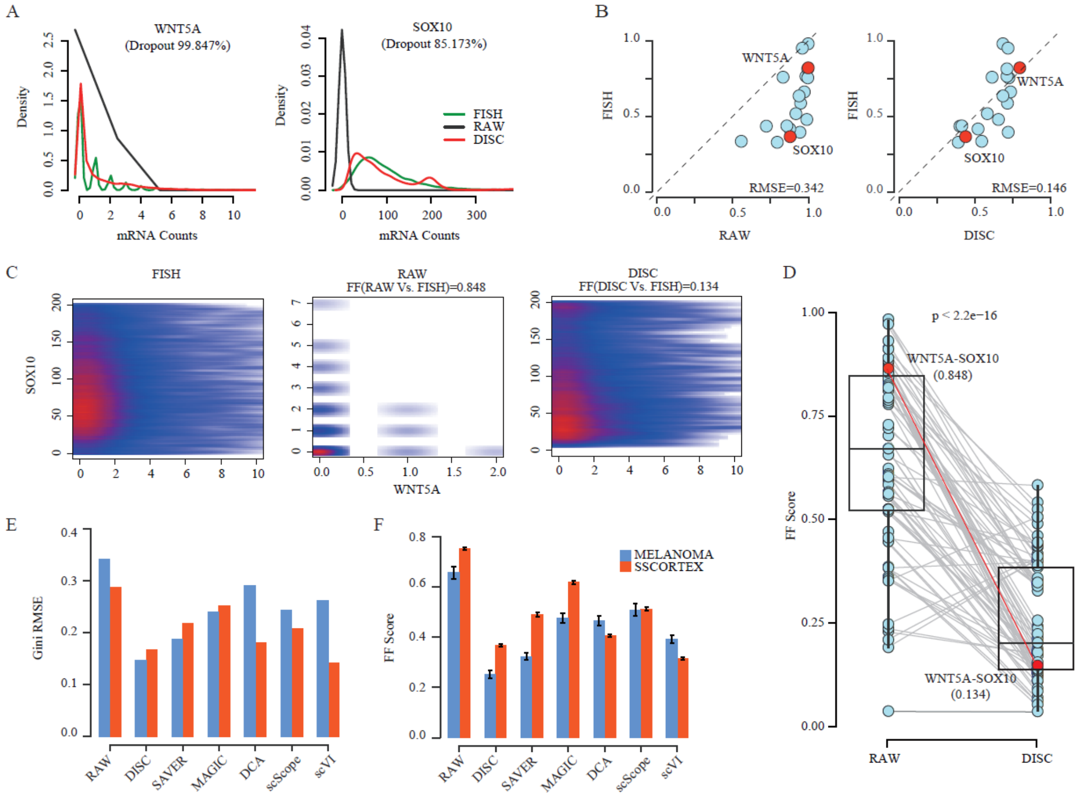

Figure 2. Evaluation of imputation performance by FISH. (**A**) Gene distribution of WNT5A and SOX10 genes from FISH, the raw scRNA-seq (RAW) and DISC, where dropout levels are shown. **(B)** Comparison of the Gini coefficient of 19 overlapped genes between FISH and RAW (left) and between FISH and DISC (right). (**C**) Scatter plot of WNT5A and SOX10 expression levels in FISH (left), RAW (middle) and DISC (left). FF scores were calculated across n = 13,564 cells for FISH and n = 8,498 cells for RAW and DISC. (**D**) FF score distribution for 81 gene-pairs between RAW and DISC against FISH. (**E**) Root mean square error (RMSE) of Gini coefficient between FISH and the imputation results, FISH to RAW is also shown. (**F**) FF scores between FISH and the imputation results, FISH to RAW is also shown.

DISC recovers dropout events

As the true expression of dropouts in scRNA-seq are not possible to obtain, we conducted down-sampling experiments on four datasets (Methods). To test the robustness of imputation performance, we used datasets generated from three different scRNA-seq platforms (Table S1). Expression matrix before down-sampling（reference), after down-sampling (observed) and imputation based on the observed were compared.

To compare recovery of true gene expression, we used Mean Absolute Error (MAE) (Figure 3A). We found that most imputation approaches have reduced MAEs compared to the observed dataset. The only exception is scScope which generated significant errors for all the datasets.

To compare recovery of expression structure, we used Pearson correlation of gene-gene relationship and cell-cell relationship. For the gene-gene relationship, DISC had the highest correlation on all the datasets (Figure 3B). Notably, compared to the observed dataset, DISC was the only approach that had improved correlations on all the four datasets while no other approaches had improvement on any dataset, illustrating DISC’s superior ability to improve gene-gene relationship. We also used CMD to assess gene co-expression (Figure 3C). CMD of DISC most matched that of the reference compared to the other approaches and the observed data. SAVER also performed well in remaining the gene co-expression while MAGIC, DCA and scScope generated large false co-expressed relationship of genes. It is also notable that this result was consistent with our previous findings where DISC and SAVER best remained gene co-expression using FISH as a validation (Figure S6). For the cell-cell relationship, DISC and SAVER also had the best overall performance while DCA and scVI also worked well on some datasets. MAGIC and scScope not only had significantly lower scores than the other approaches, but also had large variations, indicating its instability in imputation (Figure 3D).

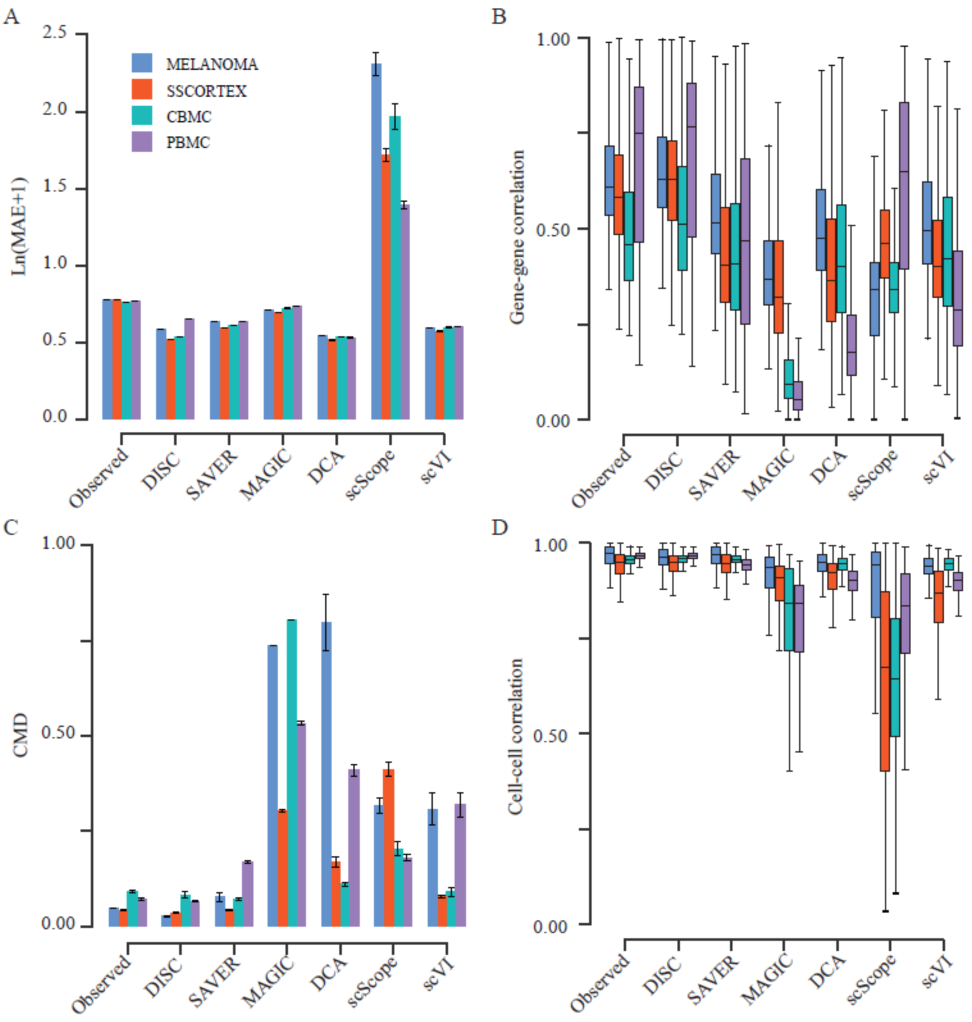

Figure 3. Evaluation of recovery of dropouts in the down-sampling experiments. (**A**) MAE between the reference and the observed/the imputation. (**B**) Gene-gene correlation between the reference and the observed/the imputation. (**C**) CMD between the reference and the observed/the imputation. (**D**) Cell-cell correlation between the reference and the observed/the imputation. (**B, D**) Box plots show the median (center line), interquartile range (hinges), and 1.5 times the interquartile range (whiskers).

DISC improves cell type identification

Having demonstrated DISC’s ability to recover dropouts, we next evaluated whether imputation improved cell type identification. We again used three datasets generated from different single cell platforms, 10X Genomics, Drop-seq and SPLiT-seq (Methods). We down-sampled the datasets to 30% and 50% of the original reads. The average cell library size, reflecting the sequence depth, before and after down-sampling are shown in Table S1.

Both clustering accuracy (ACC) and adjusted rand index (ARI) were used to assess the accuracy of cell type classification using the marker genes shown in Table S2 - S4.

For the RETINA dataset, DISC, SAVER, DCA and MAGIC improved clustering accuracy and DISC had the top performance (Figure 4A and S8). Some rare cell populations, such as RGC, muller glia and VE, completely missed in the observed data due to dropouts, were recovered by DISC (Figure 4B and S8). SAVER and MAGIC also worked well on both major and rare populations. Although DCA improved the overall accuracy, the improvement most came from the major population, Rods, that counts for 66% of the total cell populations and DCA completely missed identification of 6 other cell types. scScope only identified Rods and almost failed to identify all the other cell types.

For the PMBC dataset, DISC was the only approach that has the improved accuracy while SAVER had slightly decreased accuracy than the observed (Figure 4C and S9). Zooming into 8 cell types, DISC had the best accuracy for the 7 cell types among all the approaches (Figure 4D and S9). MAGIC failed identifying cell types using known marker genes due to the loss of marker genes for almost all the cell types (Figure S10).

The BRAIN_SPLiT dataset has 156,049 cells. Followed the original paper, we analyzed the cell types in neurons and non-neurons, separately[3]. Because this dataset is sparse, with just 1,329 mRNA counts per cell on average, and contains complex cell types, the observed dataset only had the ACC scores of 0.2 and 0.17 for neurons and non-neurons after down-sampling to 30% of the original data. DISC had the best performances for both neurons (ACC=0.46) and non-neurons (ACC=0.58). All the other approaches also considerably improved the clustering accuracy except for scScope that almost completely failed to identify any cell types (Figure 4E, 4F, S11 and S12). Zooming into each cell type, DISC and scVI had the best overall performances for all the major and rare cell populations (Figure 4G, 4H, S11 and S12). Particularly, 6 cell types in non-neurons, including major cell types such as astrocyte and rare cell types such as epend, missed due to dropouts after down-sampling, were recovered by DISC and scVI.

To sum up, DISC outperformed all the other approaches and was the only approach consistently improved the accuracy of cell type identification for all the three datasets. DISC not only improved identification for both major and rare cell types, but also worked well on datasets generated from different single cell platforms.

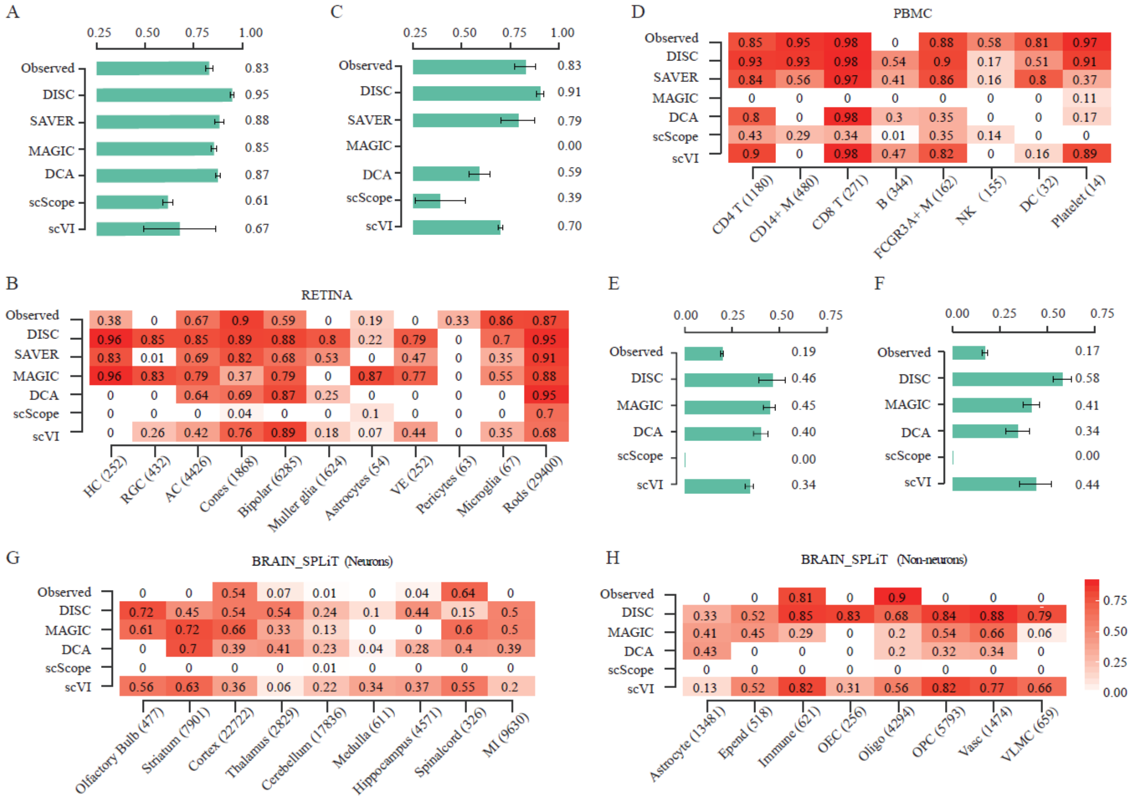

Figure 4. Evaluation of cell type identification. (A, C, E, F) Cell type identification performance using ACC as the metrics for (A) RETINA, (C) PBMC, (E) neurons of BRAIN_SPLiT and (F) non-neurons of BRAIN_SPLiT. Running time of SAVER exceeds 48 hours for BRAIN_SPLiT and is excluded from the evaluation. (**B, D, G, H**) The Jaccard index for each cell type of (B) RETINA dataset, (D) PBMC dataset, (G) neurons of BRAIN_SPLiT and (H) non-neurons of BRAIN_SPLiT.

DISC accurately identifies cell populations in the 1.3 million mouse brain dataset

We finally analyzed the BRAIN_1.3M dataset which were generated from multiple brain regions, including the cortex, hippocampus, and subventricular zone. In total, DISC identified

61 cell clusters (Figure 5A and S13). We assigned each cluster to one of three major cell groups, Glutamatergic neurons, GABAergic neurons and non-neuronal cells, using the known marker genes from the Allen Brain Atlas (Methods, Table S5), which was also used by scScope and PARC[11,16]. Approximately 1.1 million cells from 49 clusters were assigned to known cell types. The proportions of three main cell types are 64% for the Glutamatergic, 18% for the GABAergic and 18% for the non-neuronal, which more closely agree with the composition reported by PARC (65%, 18% and 17%) than scScope (63%, 17% and 20%) (Figure 5B). We assigned cells into 10 major neuronal (Figure 5C) and 6 major non-neuronal cell populations (Figure 5D), marker gene used for cell types are shown in Table S6. The smallest cell population is Microglia (5,774 cells), which had unique cell markers of C1qb and Tgfbr1, counting for 0.44% cells of the dataset (Figure 5C). These cell populations can be further categorized into sub-cell populations. For example, migrating interneurons (MI) can be further sub-grouped into three sub-populations based on distinguishing sub-cell markers (Figure 5E).

Compared to the cell types identified by DISC and scScope, we found a large discrepancy from MI. DISC identified 184,203 MI cells (14.36%) belonging to GABAergic neurons (Figure 5D), while scScope identified 543,779 MI cells (42.40%) belonging to glutamatergic neurons. By visualizing two MI markers, Dlx1 and Dlx6ox1, our analysis clearly showed that MI belongs to GABAergic groups (Figure 5F). To confirm our result, we used commonly used cell type identification tool, Seurat. Because Seurat was not able to handle such a large dataset, we down-sampled 100,000 out of 1.3 million cells. Consistent with our analysis, the analysis by Seurat also showed clear signal that MI belonged to GABAergic neurons, accounting for 14% in 100,000 cells (Figure 5G). These results demonstrate DISC’s ability to efficiently and accurately explore the major and rare cell populations in ultra-large heterogeneous single-cell datasets.

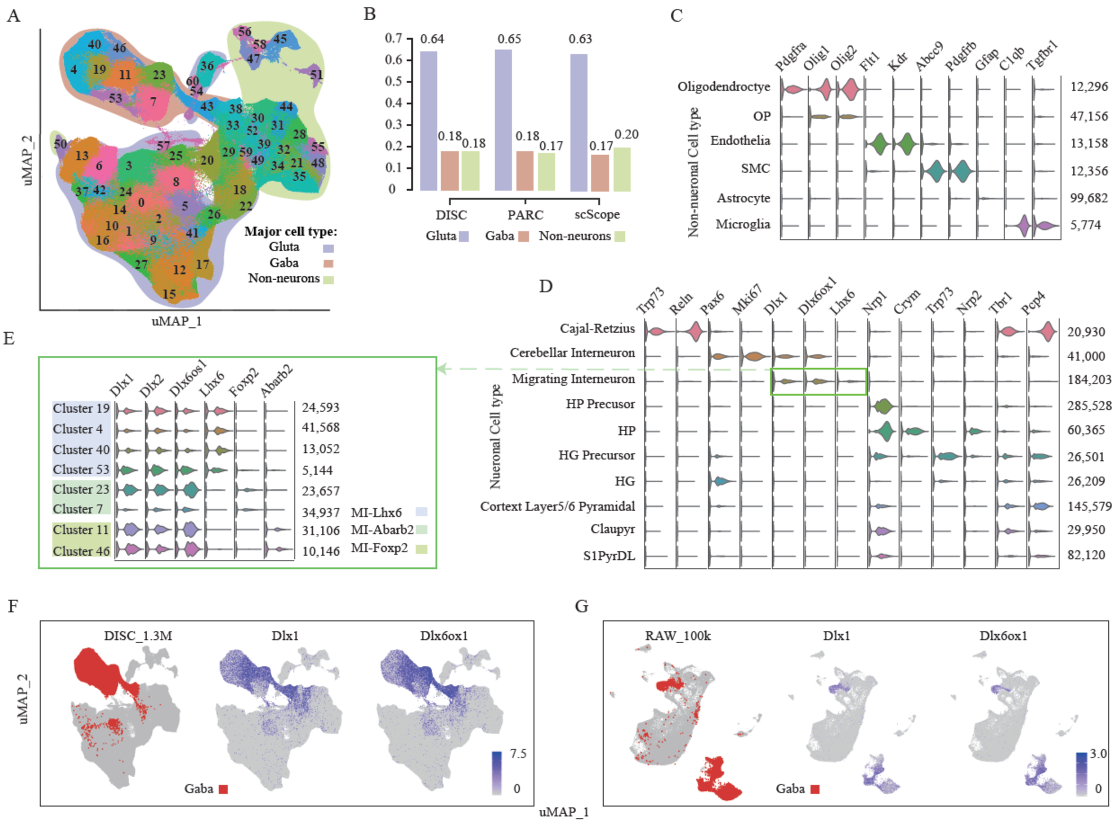

Figure 5. Analysis of BRAIN_1.3M. (A) uMAP visualization using 50 compressed dimensions for 61 clusters identified by DISC. Clusters are split into three main cell types: glutamatergic neurons (Gluta), GABAergic neurons (Gaba) and non-neuronal cells. (**B**) The proportions of three main cell types identified by DISC, PARC and scScope. (**C, D)** Cell types and marker genes for the non-neuronal cells (C) and the neuronal cells (D), the number of cells in each cell type is shown on the right. (**E**) Three sub-cell types and marker genes for MI. (**F, G**) Visualization of Gaba and MI marker genes, Dlx1 and Dlx6ox1, identified by DISC (F) and identified by Seurat (G) on 100,000 down-sampling cells.

Discussion

In the last a few years, advances in scRNA-seq technology has enabled us to obtain a few thousands to over a million of cells in just one study. Moreover, integration of datasets from different studies could provide much more biological insights than the single study does[17,18]. It is an urgent task to establish an analytic method capable of handling ultra large datasets accurately and efficiently. We designed DISC to fulfill this task. We demonstrated that DISC has three advantages. (1) Applicability: DISC is applicable to datasets generated from different platforms with different dropout levels. (2) Reliability: DISC brings significant improvements in recovering dropout events and gene expression structures. In addition, DISC consistently improves cell type identification. (3) Scalability: DISC can readily handle datasets containing millions of cells with tens of thousands of genes, which minimizes information lost from down-sampling of cells or genes.

Many factors such as expression level and distribution, level of noises and heterogeneity of cells affect the performance of imputation. DISC assumes no specific distribution of expression and dropouts. Semi-supervised deep-learning framework allows DISC to learn complex structure of genes and cells from sparse data. Unlike the other imputation approaches, DISC does not down-sample genes for the model input therefore preserves the more complete information from the data. As a result, DISC showed robust performance to datasets with different size, different dropout levels and from different platforms. We expect that DISC will continue working well as the noise distribution changes with emerging the novel platforms of scRNA-seq.

A recent benchmark study showed that many imputation approaches increased sensitivity of recovery of dropouts by scarifying specificity and therefore biases may be introduced by imputation[19]. In our study, we also found that several imputation approaches not only changed genetic and cellular structures of scRNA-seq data, but also significantly decreased accuracy of cell classification after imputation. In contrast, DISC consistently achieved top performance in recovering true gene expression, enhancing gene expression structure and improving accuracy of cell type identification.

Our results demonstrated that DISC should be used for improving cell type identification, particularly for datasets with sparse expressed genes. Making no assumption to data distribution, DISC provides a general solution for analyzing single-cell omics data. It outputs both expression matrix and low dimensional representations, which can be used for clustering and visualization by other analytical tools that have no capability to deal with ultra-large datasets. We expect DISC will be of immediate interest to the fast-growing community of single cell research.

Data and code availability

DISC is implemented in Python and builds on Google TensorFlow. It runs on both CPUs and GPUs. The source code and the datasets are available at https://github.com/xie-lab/DISC.

Reference

1 Pierson, E. & Yau, C. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome biology* **16**, 241 (2015).

2 Macosko, Evan Z. *et al.* Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell* **161**, 1202-1214 (2015).

3 Rosenberg, A. B. *et al.* Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science* **360**, 176-182 (2018).

4 Cao, J. *et al.* Comprehensive single-cell transcriptional profiling of a multicellular organism. *Science* **357**, 661-667 (2017).

5 Luecken, M. D. & Theis, F. J. Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular systems biology* **15**, e8746 (2019).

6 Huang, M. *et al.* SAVER: gene expression recovery for single-cell RNA sequencing. *Nature methods* **15**, 539–542 (2018).

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

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

9 Van Dijk, D. *et al.* Recovering Gene Interactions from Single-Cell Data Using Data Diffusion. *Cell* **174**, 716-729 (2018).

10 Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nature methods* **15**, 1053–1058 (2018).

11 Deng, Y., Bao, F., Dai, Q., Wu, L. F. & Altschuler, S. J. Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning. *Nature methods* **16**, 311-314 (2019).

12 Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature communications* **10**, 390 (2019).

13 Kostopoulos, G., Karlos, S., Kotsiantis, S. & Ragos, O. Semi-supervised regression: A recent review. *Journal of Intelligent & Fuzzy Systems* **35**, 1483-1500 (2018).

14 Lehtinen, J. *et al.* Noise2noise: Learning image restoration without clean data. *arXiv* (2018).

15 Arisdakessian, C., Poirion, O., Yunits, B., Zhu, X. & Garmire, L. X. DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. *Genome Biology* **20**, 211 (2019).

16 Stassen, S. V. *et al.* PARC: ultrafast and accurate clustering of phenotypic data of millions of single cells. *BioRxiv* (2019).

17 Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology* **36**, 411-420 (2018).

18 Hie, B., Bryson, B. & Berger, B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. *Nature biotechnology* **37**, 685-691 (2019).

19 Andrews, T. S. & Hemberg, M. False signals induced by single-cell imputation. *F1000Research* **7**, 1740 (2018).

Methods

Description of DISC

Normalization. The cell expression profile of cell c with M genes in mRNA counts $C_c \in \mathbb{Z}^{\mathrm{M}}$ is firstly normalized by cell library size with log transformation

$$\widetilde{c_c} = \ln\left(sf \frac{C_c}{ls_c} + 1 \right),$$

where ls_c is the library size of cell c, and sf (scale factor) is a constant, defined below.

Outlier detection. We use a scale factor of 1 million and calculate Z scores for the normalized counts over all cells. Z scores greater than 3 are treated as zero-count genes during training.

Input preparation. We use a scale factor of median cell library size for normalization and scale each gene m $(1 \le m \le M)$ by its normalized max (excluding outliers) over all cells to 0 - 1.

$$x_{c,m}^{0} = \frac{\widetilde{c_{c,m}}}{\widetilde{c_{\max,m}}}, ,$$

where $\{x_c^0 \in \mathbb{R}^{\mathrm{M}} | 0 \le x_{c,m}^0 \le 1\}$ is the first step input and RNN will repeats for T steps.

Encoder. The encoder layer $f_{\mathrm{E}}(\cdot)$ projects input of step t x_c^t into a low-dimensional, latent representation $\mathrm{z}_c^t \in \mathbb{R}^S$, $\mathrm{S} < M$. The encoder layer is given by

$$z_c^t = f_{\mathrm{E}}\left(x_c^t\right) = \tanh\left(w_E x_c^t\right),$$

where w_E is a learnable parameter.

Decoder. In contrast to the encoder layer, decoder layers $f_{\mathrm{D}}(\cdot)$ reverses the latent representation back into a reconstructed normalized expression profile $\left\{\widehat{y_c^t} \in \mathbb{R}^{\mathrm{M}} | 0 \le \widehat{y_{c,m}^t} \le 1\right\}$, given by

$$\widehat{y_c^t} = f_D\left(z_c^t\right) = \mathrm{sigmod}\left(2\left(\varphi+1\right) \circ \left(w_E{}^T z_c^t + b_D\right)\right),$$

where $w_E{}^T$ is the transpose of w_E and $\varphi \in \mathbb{R}^{\mathrm{M}}$, b_D are learnable parameters.

Prediction Matrix. The prediction matrix contains M channels, each channel $f_{P,m}(\cdot)$ predicts the expression of a single gene $0 \le y_{c,m}^t \le 1$ from the latent representation z_c^t as

$$y_{c,m}^t = f_{P,m}\left(z_c^t\right).$$

A channel has three layers, given by

- 1st hidden layer. $h1^{t}_{c,m} = \left(\varphi_m + 1\right)\left(w_{h1,m} z^{t}_{c} + b_{h1,m}\right)$,
- 2nd hidden layer. $h2^{t}_{c,m} = \left(\varphi_m + 1\right)\left(w_{h2,m} \cdot \tanh\left(h1^{\mathrm{t}}_{c,m}\right) + b_{h2,m}\right)$,
- Output layer. $y^{t}_{c,m} = \text{sigmod}\left(2\left(\varphi_m + 1\right)\left(\begin{array}{l} \psi^{t}_{c,m}\left(w_{p1,m} \tanh\left(h2^{t}_{c,m}\right) + b_{p1,m}\right) \\ + \left(1\text{-}\psi^{t}_{c,m}\right)\left(w_{p2,m} \tanh\left(h2^{t}_{c,m}\right) + b_{p2,m}\right) \end{array} \right)\right)$,

  $\psi^{t}_{m} = \text{sigmoid}\left(\text{SELU}\left(w_{\psi,m} h2^{t}_{c,m}\right)\right)$,

where $w_{h1,m}$, $w_{h2,m}$, $w_{p1,m}$, $w_{p2,m}$, $w_{\psi,m}$, $b_{h1,m}$, $b_{h2,m}$, $b_{p1,m}$ and $b_{p2,m}$ are learnable parameters for gene m. The output layer is a weighted average over two channels using ψ_{c}^{t} as weight factor. Before sigmoid activation, scaled exponential linear unit[1] (SELU) activation is used to make the channel selection towards the first channel, assuming that most cells obey one expression distribution.

Filter. Input for the next step, x^{t+1}_{c}, is prepared by filtering of y^{t}_{c} to keep the positive-counts as

$$x^{t+1}_{c,m} = \begin{cases} x^{0}_{c,m}, & x^{0}_{c,m} > 0 \\ y^{t}_{c,m}, & x^{0}_{c,m} = 0 \end{cases}.$$

Imputer and Reconstructor. A soft attention assigns a weight vector a^{t}_{c} to the decoding $\widehat{y^{t}_{c}}$ and prediction y^{t}_{c} output from each recurrence. a^{t}_{c} is given by

$$a^{t}_{c,m} = \text{softmax}\left(w_{a,m}\text{SELU}\left(h1^{t}_{c,m}\right)\right),$$

where $w_{a,m}$ is a learnable parameter. After weighted average, $\left\{\widehat{y_{c}} \in \mathbb{R}^{\mathrm{M}} | 0 \le \widehat{y_{c,m}} \le 1\right\}$ and $\left\{y_{c} \in \mathbb{R}^{\mathrm{M}} | 0 \le y_{c,m} \le 1\right\}$ are given by

$$\widehat{y_{c}} = \sum_{t} a^{t}_{c} \circ \widehat{y^{t}_{c}} \text{ and } y_{c} = \sum_{t} a^{t}_{c} \circ y^{t}_{c}.$$

Compressor. The latent representations over all steps, $z_c \in \mathbb{R}^{\mathrm{L \times N}}$, are compressed further to a lower dimension $W \ll S \cdot T$. Compressor is an autoencoder whose encoder is given by

$$cp_c = \tanh\left(w_z z_c + b_{z1}\right),$$

And the reverse decoder is given by

$$\widetilde{z_c} = \tanh\left(w_z{}^T cp_c + b_{z2}\right),$$

where w_z, b_{z1} and b_{z2} are learnable parameters, cp_c is the compressed cell feature where $cp_c \in \mathbb{R}^W$. Autoencoder and compressor modules together form a stacked autoencoder. To evaluate the performance of the compressor, the cell expression profile $\widetilde{y_c}$ is reversed from $\widetilde{z_c}$, given by

$$\widetilde{y_c} = \sum_t a_c^t \circ \widetilde{y_c^t},$$

where a^t is the shared soft attention weight for the imputer and reconstructor modules and $\widetilde{y_c^t}$ is reversed from $\widehat{z_c^t}$ using the decoder module.

Training of DISC

The parameters of DISC are optimized from the data in an end-to-end manner according to a combination of five loss functions, including imputation loss (L_I), reconstruction loss (L_R), prediction loss (L_P), latent representation loss (L_{LR}) and constraint (L_C).

Imputation loss. L_I is formulated based on the idea of "noise to noise" for image imputation[2]. A noise input nx_c^0 for the first step is prepared by assigning an uniform multiplicative noise: $U_c^M\left(0.9, 1.1\right) \circ x_c^0$ and nx_c^0 replaces x_c^0 for filtering of predicted expression profile to produce inputs for the later steps, nx_c^t. In addition, a dropout operation is applied to nx_c^t on zero-count genes in raw data[3]. At the end, a noise imputation output ny_c is produced and L_I is formulated as

$$L_I = \frac{1}{N}\sum_c \left\| \alpha 1_c \circ \left(ny_c - ny_c^{'} \right) \right\|_1 ,$$

where $ny_c^{'}$ is a noise target given by $U_c^M\left(0.9,1.1\right)\circ x_c^0$, N is the number of cells for training. L_I only computes the positive-counts restricted by $\alpha 1$, given by

$$\alpha 1_{c,m} = \begin{cases} 1, & x_{c,m}^0 > 0 \\ 0, & x_{c,m}^0 = 0 \end{cases} .$$

Reconstruction loss. L_R is formulated using semi-supervised learning (SSL) to learn a concordant latent representation which encodes both positive-counts and pseudo-counts assigned by the imputer as:

$$L_R = \frac{1}{N}\sum_c \left\| \alpha 2_c \circ \left(\widehat{y_c} - \widehat{y_c^{'}} \right) \right\|^2 ,$$

where $\alpha 2_{c,m} = \begin{cases} \alpha_R, & x_{c,m}^0 > 0 \\ 1, & x_{c,m}^0 = 0 \end{cases}$, α_r balances the biased portions towards zero-counts, the reconstruction target is $\widehat{y_{c,m}^{'}} = \begin{cases} x_{c,m}^0, & x_{c,m}^0 > 0 \\ y_{c,m}, & x_{c,m}^0 = 0 \end{cases}$.

Prediction loss. L_P uses SSL to search an expression profile structure which underlying both positive-counts and pseudo-counts assigned by the decoder, given by

$$L_P = \frac{1}{N}\sum_t \sum_c \left\| \alpha 3_c \circ \left(y_c^t - y_c^{t'} \right) \right\|^2 ,$$

where $\alpha 3_{c,m} = \begin{cases} \alpha_{P1}, & x_{c,m}^0 > 0 \\ \alpha_{P2}, & x_{c,m}^0 = 0 \end{cases}$ and the prediction target is $y_{c,m}^{t\ '} = \begin{cases} x_{c,m}^0, & x_{c,m}^0 > 0 \\ \widehat{y_{c,m}^t}, & x_{c,m}^0 = 0 \end{cases}$.

Latent representation loss. Prediction of expression profile made by each step is a function of the corresponding latent representation. L_{LR} minimizes the difference between successive latent representations, given by

$$L_{LR} = \frac{1}{N \cdot T} \sum_{t=1}^{T+1} \sum_{c} \left\| x_c^t w_E - x_c^{t-1} w_E \right\|^2 .$$

Constraint. L_C limits the total capacity of imputation counts assuming most zero-counts are either low expressed or unexpressed. L_C is given by

$$L_C = \sum_{t} \sum_{c} \left\| \alpha 4_c \circ f_{de}\left(y_c^t\right) \right\|^2 ,$$

where $\alpha 4_{c,m} = \begin{cases} 0, & x_{c,m}^0 > 0 \\ 1, & x_{c,m}^0 = 0 \end{cases}$ and f_{de} is a function reverses the normalized counts back to counts.

Regularization. We assumed that some genes contribute more (strong connection) to each neuron of the latent representation. However, conventional sparse regularizers, i.e. L1 regularizer and Log regularizer, are unable to restrict the number of genes having strong connections to the neurons. We developed a new regularizer, f_{re}, to restrict the genes as

$$f_{re}(w) = \sum_{i}^{NN_w} \left(\sum_{j \in w_i} j^2 \right)^2 ,$$

where NN_w is the number of output-nodes, w_i is the collection of weights connecting with i^{th} output-node. j^2 removes weights that are very small.

The overall loss function is

$$\begin{aligned}\mathrm{L} = {} & \beta_1 L_I + \beta_2 L_R + \beta_3 L_P + \beta_4 L_{LR} + \beta_5 L_C \\ & + \beta_6 \left(f_{re}(w_E) + f_{re}(w_{h_1}) \right) \\ & + \beta_7 \sum_{w \in w_{h_2}, w_{p_1}, w_{p_2}} w^2 + \beta_8 \sum_{w \in w_a} w^2 + \beta_9 \sum_{w \in \varphi} w^2\end{aligned} \quad .$$

DISC was trained using Adam[4]. Gradient clipping of 5 was used to avoid exploding gradient.

Stop of training. Predictor of DISC is a function of the latent representation, z^t. When the difference of z^t across multiple steps becomes smaller, DISC is convergent to a stable point. Therefore, DISC uses latent representation loss to evaluate the similarity of z^t across multiple steps and to determine the best stop point based on the variance of this loss over multiple batches (10000 batches by default). We chose 5 million cells as an initial point because DISC generally reached optimal points after learning information from approximately 5 million cells in many datasets with a variety of gene and cell numbers. This property makes DISC a stable running time for datasets of various sizes. The procedure is as follows.

1. DISC is first trained for 5 million cells (128 cells per training batch on default) and calculate the standard deviation (STD) of L_{LR} for the last 10000 batches. This STD is set as the minimum STD, and this STD remains as the minimum STD for 1 round (minimum round where a training round is 50000 cells).
2. DISC is trained for another 50000 cells and calculates a new STD of L_{LR} for the last 10000 batches.
3. If the new STD is greater than the minimum STD, minimum round is increased by 1. Otherwise, minimum STD is set as the new STD, minimum round is reset to 1.
4. If minimum round is less than 5, repeat step 2. Otherwise, training is stopped.

Hyperparameter optimization

Hyperparameters for the model architecture, including layer neuron numbers, number of steps and learning rate, are pre-defined (Figure S1), and the other ones were sampled using Latin Hypercube Sampling[5]. The following hyperparameters were used for the initial model:

$\alpha_R = 5$, $\alpha_{P1} = 1.5$, $\alpha_{P2} = 0.35$, $\beta_1 = 1$, $\beta_2 = 1$, $\beta_3 = 1$, $\beta_4 = 1.65 \times M \times 1e^{-5}$, $\beta_5 = 6.3 \times 1e^{-5}$,

$\beta_6 = 1e^{-6}$, $\beta_7 = 1e^{-6}$, $\beta_8 = 1e^{-5}$ and $\beta_9 = 1e^{-4}$.

Generating training batches

To randomize cell orders in an expression matrix, a common practice is to load a complete expression matrix into memory and random sample cell batches. However, loading large expression matrix usually causes out-of-memory errors (OOM). A previous method split the expression matrix into several parts and saved onto hard disk[6]. During training, parts were loaded separately to generate random cell batches. However, by this approach, random sampling was performed locally within the parts and pre-processing required extra work. Here, we developed a novel method to generate globally random cell batches.

1. Cells are indexed by chunks of arbitrary size (32 cells by default).
2. Multiple chunks are loaded randomly (64 chunks by default) into a sub-queue in the memory and cells in the sub-queue are shuffled. Once shuffled, cells are transferred into a main queue in the main thread.
3. Cells are loaded parallelly via parallel sub-queues to reduce the loading delay and cells from different sub-queues are transferred randomly into the main queue.
4. At the end, cell batches are withdrawn from the main queue based on first-in-first-out rule.

Cell type identification

Small dataset. For smaller datasets, including PBMC, RETINA, neuronal cells (129K cells) and non-neuronal cells (27K cells) of BRAIN_SPLiT datasets, Seurat V3.0[7] was used to perform normalization, feature selection, scaling, PCA, clustering and t-SNE/u-MAP visualization. Resolution and PCA-dimension parameters for clustering were selected to produce the best accuracy against cell type labels. Specifically, resolution of 0.5-1.4 (0.5 for PBMC, 1.4 for RETINA, 1.4 for BRAIN_SPLiT）and top 10-50 principal components of

PCA (10 for PBMC, 30 for RETINA, 50 for BRAIN_SPLiT) were used and clustering was based on the graph-based shared nearest neighbor method (SNN). Differential expression analysis was used to identify cluster-specific marker genes where all the clusters are pairwise compared using the Wilcoxon method. Each identified marker gene was expressed in a minimum of 25% of cells and at a minimum log fold change threshold of 0.25.

Large dataset. For the BRAIN_1.3M dataset with 1.3 million cells, traditional methods are unable to cluster cells using the whole expression matrix. Here, compressed features of 50 dimensions from DISC were used for clustering by Seurat, where resolution was set to 1.4 and top 50 PCA-principal components of PCA were used. Differential expression analysis was described above.

Evaluation of imputation performance

Gene selection. Genes match the following conditions were removed for further analysis

1. Expressed in less than 1/1000 cells or less than 10 cells, whichever is greater.
2. Maximum mRNA count is 1.

Comparison of scRNAseq and FISH. Genes overlapped between scRNA-seq (≥10 positive-count cells) and FISH were selected. To compare the expression distributions of scRNA-seq and FISH, each selected gene was normalized by an efficient factor[8], where efficient factor was defined as the ratio of its FISH mean to its scRNA-seq (raw or imputation) mean.

Down-sampling. We randomly sampled transcript reads from scRNA-seq dataset followed a previous research[9]. Transcripts were sampled either 30% or 50% of the original cell library size.

Gini coefficient. We used “reldist” package in R to calculate Gini coefficient to quantify gene expression distribution[10]. The difference of Gini coefficients between scRNA-seq (raw and imputation) and FISH was calculated by rooted mean square error (RMSE), given as

$$Gini\ RMSE_{method} = \sqrt{\frac{\sum_{i=1}^{n}\left(Gini_{FISH,i} - Gini_{method,i}\right)^2}{n}},$$

where n is the number of overlapped genes, i is the index of the genes.

Fasano and Franceschini's Test. Kolmogorov-Smirnov (K-S) distance[11] is a nonparametric estimation of the distance between two one-dimensional probability distributions, based on their cumulative distributions. Fasano and Franceschini's (FF) distance[12] is a multi-dimensional version of K-S distance. Using FISH data as the reference, we used a script (https://github.com/syrte/ndtest/blob/master/ndtest.py) to calculate FF distance as a measurement for similarity of the gene-gene co-expression distribution between scRNA-seq (raw and imputation) and FISH.

Correlation matrix distance (CMD). CMD is a measure of the distance between two correlation matrices[13]. The CMD for two correlation matrices R_1, R_2 is defined as

$$d\left(R_1, R_2\right) = 1 - \frac{tr\left(R_1, R_2\right)}{\left\|R_1\right\|_f \left\|R_2\right\|_f}$$

For comparison with FISH, Pearson’s correlation was calculated for gene pairs in R_1 (FISH) and R_2 (raw or imputation) using all the overlapped genes. For comparison in down-sampling dataset, Pearson’s correlation were calculated for gene pairs in R_1 (reference) and R_2 (observed or imputation) using top 300 variable feature genes selected by Seurat’s “vst” function[7,14].

Mean absolute error (MAE). MAE measures the difference of gene expressions of the observed or imputation data to the reference data, given by

$$MAE = \frac{\sum_{i=1}^{n} \left| C_{ds}^{i} / ds\_ratio - C_{reference}^{i} \right|}{n},$$

where *n* is the number of positive-count genes in the reference data that dropout after down-sampling, C_{ds} is the observed / imputed mRNA counts, $C_{reference}$ is the mRNA counts before down-sampling, ds_ratio is ratio of down-sampling.

Gene-gene and cell-cell correlation. Pearson correlation was calculated at the gene or cell levels before and after down-sampling. At the gene level, genes were included if they express in at least 10% of cells. At the cell level, cells were included if they have at least 10% of gene expressed

Evaluation of cell type annotation accuracy. To evaluate cell type accuracy, three evaluation metrics are used. Accuracy (ACC) and adjusted rand index (ARI) are used to assess the properties of the overall clustering results and Jaccard index is used to calculate the accuracy of each cell types.

ACC is calculated as

$$ACC = \frac{\sum_{i=1}^{n} \delta(r_i, s_i)}{n}$$

where n is the cell number, r_i and s_i are the cell type label and classified cell type, respectively, for i th cell and

$$\delta(x, y) = \begin{cases} 1 & if \quad x = y \\ 0 & otherwise \end{cases}.$$

The overlap between the cell type labels and classified cell type can be summarized in a contingency table, in which each entry denotes the number of objects in common between the two sets.

The ARI is calculated as

$$ARI = \frac{\sum_{i=1}^{|\mathrm{K}|}\sum_{j=1}^{|\mathrm{K}|}\binom{n_{i,j}}{2} - \left[\sum_{i=1}^{|\mathrm{K}|}\binom{a_i}{2}\sum_{j=1}^{|\mathrm{K}|}\binom{b_j}{2}\right]\Big/\binom{n}{2}}{\frac{1}{2}\left[\sum_{i=1}^{|\mathrm{K}|}\binom{a_i}{2}\sum_{j=1}^{|\mathrm{K}|}\binom{b_j}{2}\right] - \left[\sum_{i=1}^{|\mathrm{K}|}\binom{a_i}{2}\sum_{j=1}^{|\mathrm{K}|}\binom{b_j}{2}\right]\Big/\binom{n}{2}},$$

where K is the set of unique cell type labels, $n_{i,j}$ are values from the contingency table, a_i is the sum of the i th row of the contingency table, b_j is the sum of the j th column of the contingency table and $\binom{}{}$ denotes a binomial coefficient and $\binom{n}{2}$ means $\frac{n(n-1)}{2}$.

Jaccard index is calculated as

$$J(c,d,k) = \frac{|c \cap d|}{|c| + |d| - |c \cap d|},$$

respectively, where c is the set of cells with type labels k, d is the set of cells with classified cell type and $k \in \mathrm{K}$.

Comparison of imputation approaches

The imputation approaches were run on a Linux CentOS 7 server with 2 Intel® Xeon® E5-2650 v4 CPUs, 128GB RAM and 1 NVIDIA® Tesla® V100 GPU. Unless otherwise noted, software packages were used with their default settings after gene selection. For all deep learning methods (DISC, DCA, scVI and scScope), GPU were used for training and imputation. The running scripts can be found at https://github.com/xie-lab/DISC/tree/master/reproducibility/source/other_methods

SAVER. We used the R package of SAVER v1.1.1. The output RDS-formatted files were used for gamma or negative binomial prior for its value uncertainty.

MAGIC. We used the Python package of magic-impute v1.5.5. Following its tutorial (https://nbviewer.jupyter.org/github/KrishnaswamyLab/MAGIC/blob/master/python/tutorial_notebooks/emt_tutorial.ipynb), we performed library size normalization and square root transformation before imputation. We then squared and denormalized its output gene expressions after imputation.

DCA. We used the Python package of DCA v0.2.2.

scScope. We used the Python package of scScope v0.1.5. Following its demo script (https://github.com/AltschulerWu-Lab/scScope/blob/master/demo.py), we normalized each cell to have same library size, set the feature dimension as 50 and then imputed dropout values after training with the default setting.

scVI. We used the Python package of scVI v0.3.0, followed the reproducibility script (https://github.com/YosefLab/scVI/blob/aa614bdaf2ff57fbb661394e53a9a2454b950882/tests/notebooks/scVI_reproducibility.ipynb).

Speed and memory comparison. Speed and memory usage were compared using BRAIN_1.3M dataset. Cells express less than 500 or greater than 5,000 genes were removed (approximately 1.3 million cells left). The top 1000 or 10000 most variable genes were selected using “vst” (variance stabilizing transformation) of Seurat. We then randomly sampled 3 subsets in different cell numbers (50k, 100k and 500k cells). We duplicated 1.3M datasets into a 2.6M cell dataset. For each imputation method compared, we ran each dataset 3 times and calculated the average computation time and memory usage.

Datasets

MELANOMA (GSE99330, 8,498 melanoma cells by Drop-seq) with FISH. 8,640 cells from the melanoma WM989 cell line were sequenced using Drop-seq[15], where 32,287 genes were detected. 8,498 cells were extracted according to the previous pipeline[8] and 15,204 genes were left after gene selection. In addition, RNA FISH experiment of across 7,000-88,000 melanoma cells from the same cell line was conducted and 26 were detected[16], in which 19 genes were overlapped with the 15,204 genes, including 9 housekeeping genes (*BABAM1*, *GAPDH*, *LMNA*, *CCNA2*, *KDM5A*, *KDM5B*, *MITF*, *SOX10*, and *VGF*) and 10 drug-resistance markers (*C1S*, *FGFR1*, *FOSL1*, *JUN*, *RUNX2*, *TXNRD1*, *WNT5A*, *EGFR*, *PDGFC*, and *VCL*). RNA-seq data can be found at GSE99330. RNA FISH data can be found at https://www.dropbox.com/s/ia9x0iom6dwueix/fishSubset.txt?dl=0.

SSCORTEX (SRP135960, 3,447 and 3,969 mouse somatosensory cortex cells in 2 replications by 10X Genomics) with FISH. Mouse somatosensory cortex of CD-1 mice at age of p28 and p29 were profiled by 10X where 7,477 cells were detected in total[17]. Cells express less than 500 or greater than 5,000 genes were removed (7,416 cells left) and 13,997 genes were left after gene selection. osmFISH experiment of 4,839 cells from somatosensory cortex, hippocampus and ventricle from a CD-1 mouse at age of p22 was conducted[18]. 4,388 cells from somatosensory cortex were extracted with 33 genes detected where all of the FISH genes were overlapped with the 13,997 genes, including *GAD2*, *SLC32A1*, *CRHBP*, *CNR1*, *VIP*, *CPNE5*, *PTHLH*, *CRH*, *TBR1*, *LAMP5*, *RORB*, *SYT6*, *KCNIP2*, *ALDOC*, *GFAP*, *SERPINF1*, *MFGE8*, *SOX10*, *PLP1*, *PDGFRA*, *BMP4*, *ITPR2*, *TMEM2*, *CTPS*, *ANLN*, *MRC1*, *HEXB*, *TTR*, *FOXJ1*, *VTN*, *FLT1*, *APLN* and *ACTA2*. RNA-seq data can be extracted from http://loom.linnarssonlab.org/clone/Mousebrain.org.level1/L1_Cortex2.loom. The osmFISH data can be found at http://linnarssonlab.org/osmFISH/availability/.

PBMC (2,638 freeze-thaw human PBMC cells by 10X Genomics). 2,700 freeze-thaw peripheral blood mononuclear cells (PBMC) from a healthy donor were profiled by 10X, where 32,738 genes were detect[19]. Cells express less than 200 or greater than 2,500 genes or have >5% mitochondrial counts were removed (2,638 cells left) and 13,997 genes were left after gene selection. RNA-seq data can be found at https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/frozen_pbmc_donor_a.

CBMC (GSE100866, 8,005 human CBMC cells by CITE-seq). Cord blood mononuclear cells were profiled by CITE-seq, where 8,005 human cells were detected in total. We used all detected human (20,400) genes for down-sampling[20]. RNA-seq data can be found at GSE100866.

RERINA (GSE63473, 49,300 retina STAMPs by Drop-seq). Retinas of mice at age of p14 were profiled in 7 different replicates by Drop-seq, where 6,600, 9,000, 6,120, 7,650, 7,650, 8280, and 4000 STAMPs (single-cell transcriptomes attached to micro-particles) were collected with 24,658 genes detected[21]. Cells were merged and 14,871 genes were left after gene selection. 44808 cells labelled STAMPs were used for evaluation. RNA-seq data can be found at GSE63473.

BRAIN_SPLiT (GSE110823, 156,049 mouse brain and spinal cord nuclei by SPLiT-seq). 156,049 mice nuclei from developing brain and spinal cord at age of p2 or p11 mice were profiled by SPLiT-seq, where 26,894 genes were detected[22]. RNA-seq data can be found at GSE110823.

BRAIN_1.3M (1,282,594 mouse brain cells by 10X Genomics). 1,306,127 cells from combined cortex, hippocampus, and subventricular zone of 2 E18 C57BL/6 mice were profiled by 10X, where 27998 genes were detected[19]. Cells expressed less than 500 or greater than 5,000 genes were removed, and 1,282,594 cells were kept for further analysis. RNA-seq data can be found at https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.3.0/1M_neurons.

Method Reference

1 Klambauer, G., Unterthiner, T., Mayr, A. & Hochreiter, S. in *Advances in neural information processing systems.* 971-980.

2 Lehtinen, J. *et al.* Noise2noise: Learning image restoration without clean data. *arXiv* (2018).

3 Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **15**, 1929-1958 (2014).

4 Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

5 McKay, M. D., Beckman, R. J. & Conover, W. J. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239-245 (1979).

6 Deng, Y., Bao, F., Dai, Q., Wu, L. F. & Altschuler, S. J. Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning. *Nature methods* **16**, 311-314 (2019).

7 Stuart, T. *et al.* Comprehensive Integration of Single-Cell Data. *Cell* **177**, 1888-1902.e1821 (2019).

8 Huang, M. *et al.* SAVER: gene expression recovery for single-cell RNA sequencing. *Nature methods* **15**, 539–542 (2018).

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

10 Relative Distribution Methods v. 1.6-6. Project home page at http://www.stat.ucla.edu/~handcock/RelDist (2016).

11 Massey Jr, F. J. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association* **46**, 68-78 (1951).

12 Fasano, G. & Franceschini, A. A multidimensional version of the Kolmogorov–Smirnov test. *Monthly Notices of the Royal Astronomical Society* **225**, 155-170 (1987).

13 Herdin, M., Czink, N., Ozcelik, H. & Bonek, E. in *2005 IEEE 61st Vehicular Technology Conference.* 136-140 (IEEE).

14 Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology* **36**, 411 (2018).

15 Torre, E. *et al.* Rare cell detection by single-cell RNA sequencing as guided by single-molecule RNA FISH. *Cell systems* **6**, 171-179.e175 (2018).

16 Shaffer, S. M. *et al.* Rare cell variability and drug-induced reprogramming as a mode of cancer drug resistance. *Nature* **546**, 431 (2017).

17 Zeisel, A. *et al.* Molecular architecture of the mouse nervous system. *Cell* **174**, 999-1014.e1022 (2018).

18 Codeluppi, S. *et al.* Spatial organization of the somatosensory cortex revealed by osmFISH. *Nature methods* **15**, 932 (2018).

19 Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nature Communications* **8**, 14049 (2017).

20 Stoeckius, M. *et al.* Simultaneous epitope and transcriptome measurement in single cells. *Nature methods* **14**, 865-868 (2017).

21 Macosko, Evan Z. *et al.* Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell* **161**, 1202-1214 (2015).

22 Rosenberg, A. B. *et al.* Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science* **360**, 176-182 (2018).

Acknowledgements

Z.X. was supported by National Natural Science Foundation of China (31829002). Y.H. was supported by China Postdoctoral Science Foundation (2019M663220). This project was also supported by Center for Precision Medicine, Sun Yat-sen University.

Author contributions

Z.X. and Y.H conceived and designed the study. Y.H. designed the model. Y.H. and H.Y developed the software. C.W., Y.H. and H.Y. analyzed the data. Z.X., Y.H., C.W. and H.Y. wrote the manuscript.

Author declaration

The authors declare no competing interests.