iScience, Volume *24*

Supplemental information

Imputing single-cell RNA-seq data by combining graph convolution and autoencoder neural networks

Jiahua Rao, Xiang Zhou, Yutong Lu, Huiying Zhao, and Yuedong Yang

Supplemental information

Supplemental figures and legends

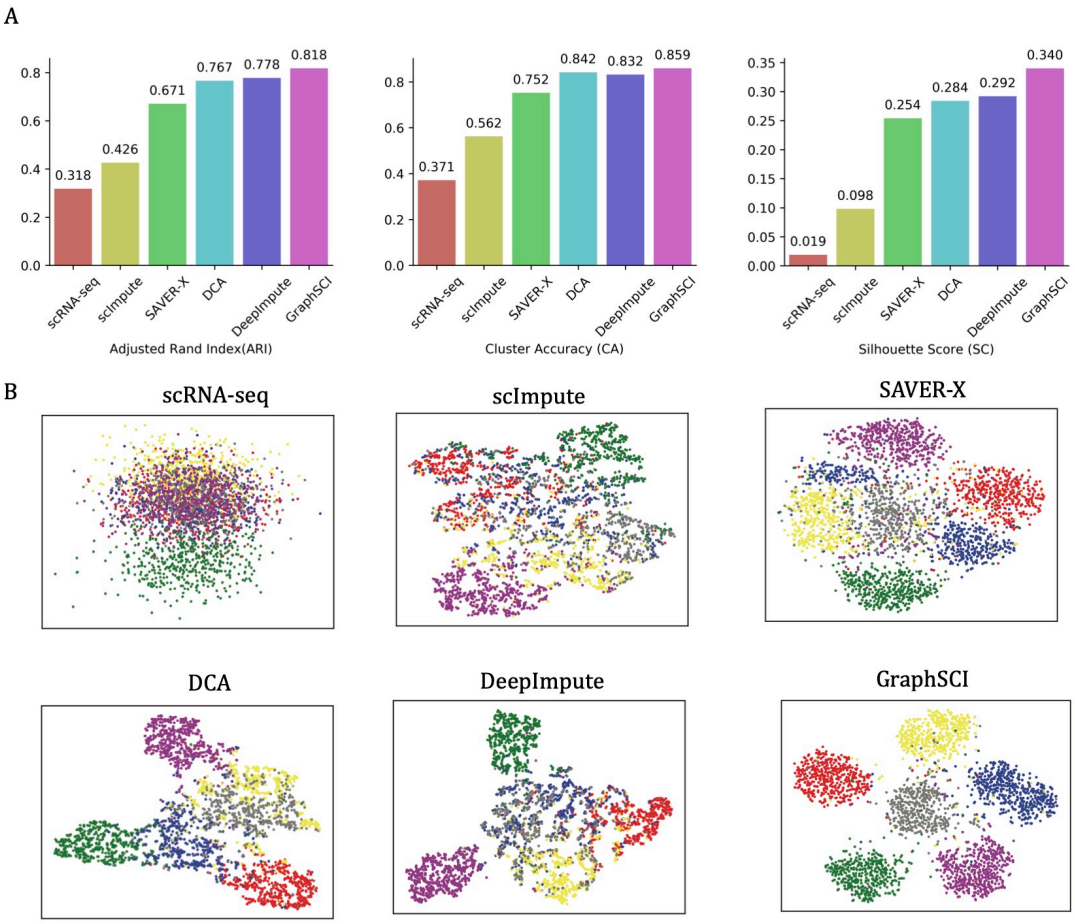

Figure S1. GraphSCI identifies cell types in simulated data with six cell groups (SIM-T6), Related to Figure 3. (A) The comparison of clustering performances of scRNA-seq, scImpute, SAVER, DCA, DeepImpute and GraphSCI, measured by ARI, CA and SC. (B) The two principle components by t-SNE from simulated scRNA-seq data, imputed matrix by scImpute, SAVER, DCA, DeepImpute and GraphSCI. Each cell is colored by cell groups.

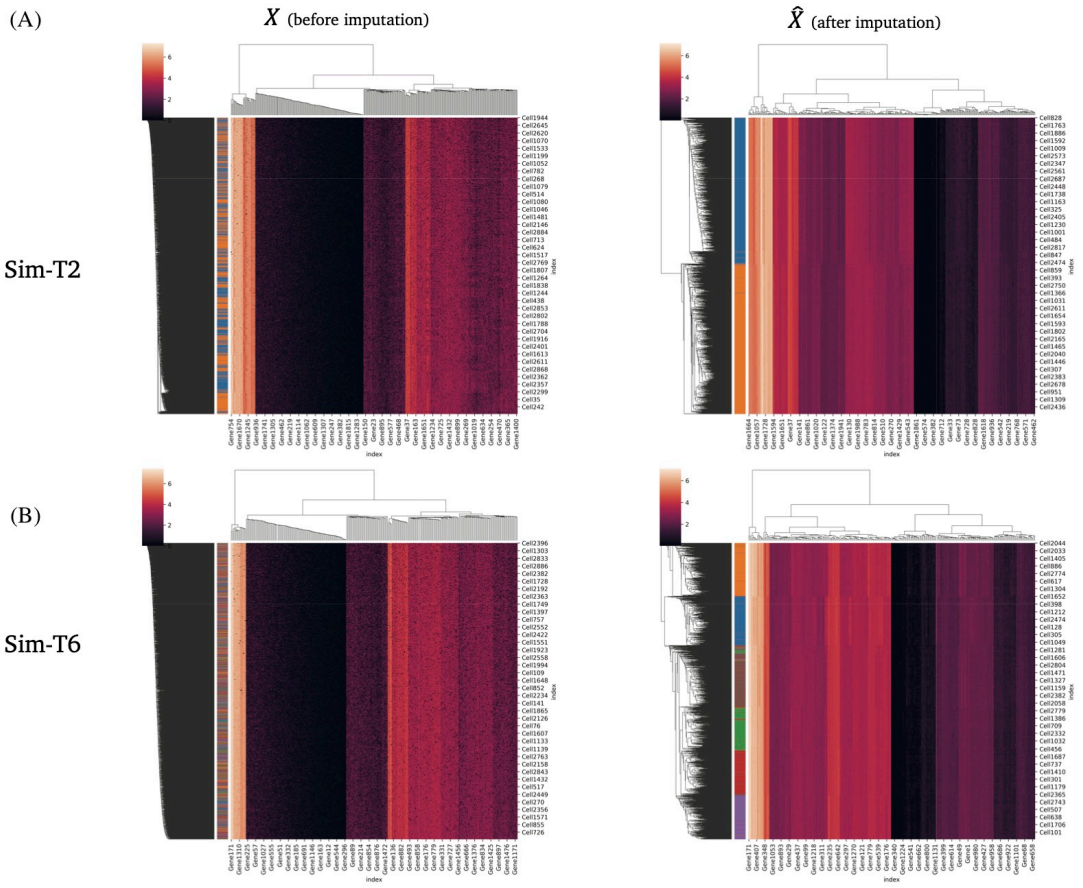

Figure S2. The image of gene expression matrix (X) before and after imputation ($\hat{X}$) in our simulated experiments, Related to Figure 3. The X axis represents cells and arranges the same cell types are nearby and the Y axis represents genes and similar genes nearby. (A) The comparison of gene expression matrix (X) before and after imputation ($\hat{X}$) on Sim-T2. (B) The comparison of gene expression matrix (X) before and after imputation ($\hat{X}$) on Sim-T6. After imputation using GraphSCI, we could find that the original cell-types can be recovered effectively both in the Sim-T2 and the Sim-T6 datasets. The cells of the same cell types are effectively clustered. This result verifies the effectiveness of our algorithm.

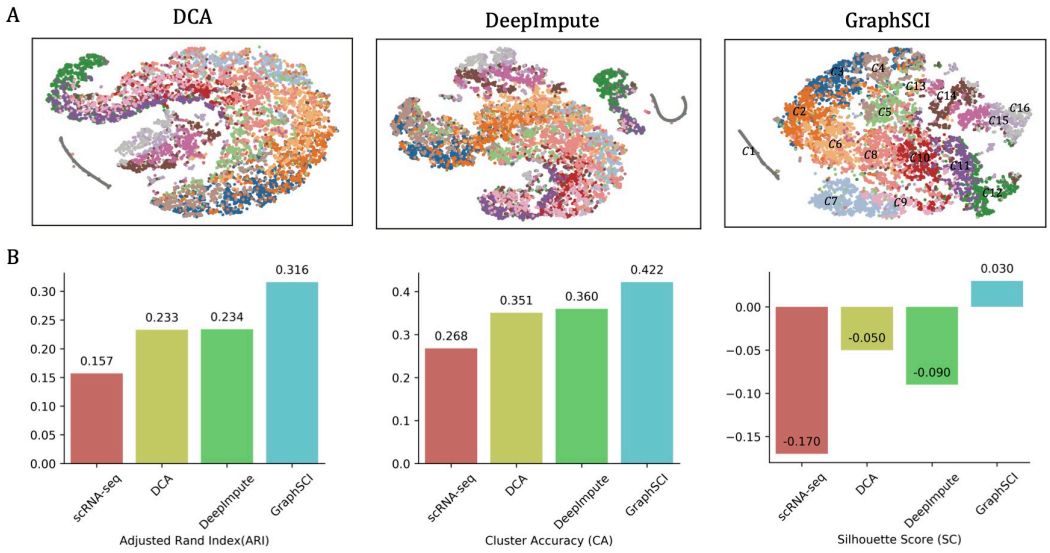

Figure S3. The performances on 10k Brain Cells from an E18 Mouse dataset, Related to Figure 5. (A) shows the t-SNE visualization reproduced from DCA, DeepImpute and GraphSCI from left to right. (B) The comparison of clustering performances of scRNA-seq, DCA, DeepImpute and GraphSCI, measured by ARI, CA and SC.

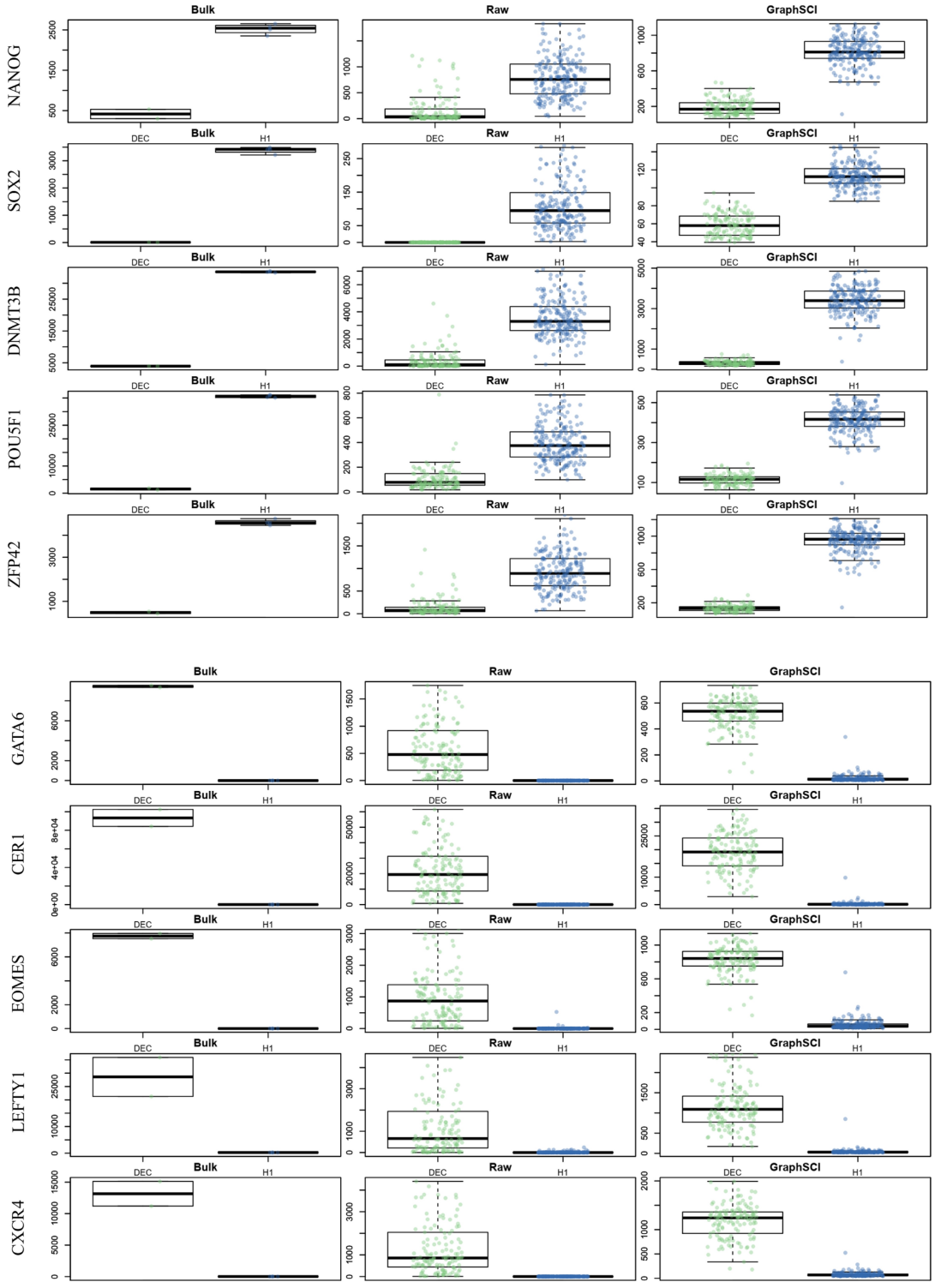

Figure S4. The performances of differential expression analysis, Related to Figure 7. The expression for signature genes (NANOG, SOX2, DNMT3B, POU5F1, ZFP42; GATA6, CER1, EOMES, LEFTY1, CXCR4) of H1 and DEC cells, respectively.

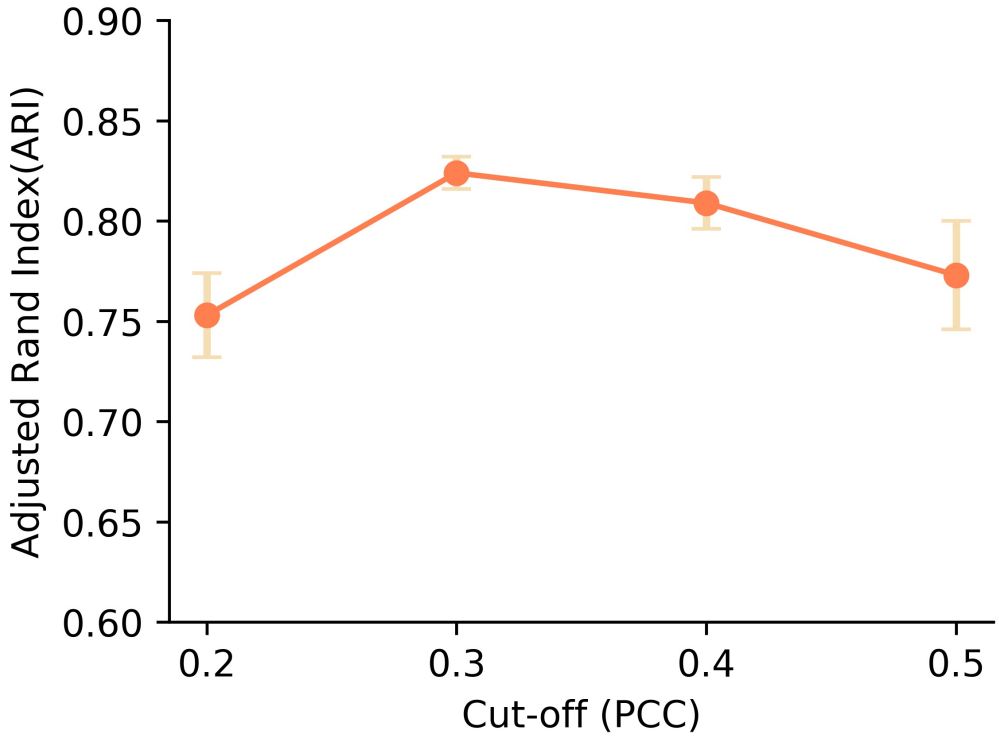

Figure S5. The analysis of different PCC cut-offs to construct the input gene-to-gene relationships, Related to Figure 1-2. We vary the cut-off of Pearson Correlation in {0.2, 0.3, 0.4, 0.5} to investigate their influences on the overall results. We could see that all relatively large cut-offs could achieve convergence, but the middle two could obtain better results. One possible reason is that the highest cut-off of Pearson Correlation might lead to a sparse adjacency matrix while the small cut-offs lead to more false-positive edges. It makes sense since a sparse adjacency matrix or an adjacency matrix with many false-positive edges would prevent our model from obtaining better results. It also proves that our algorithm could achieve stable final results if the cut-off is in a proper range.

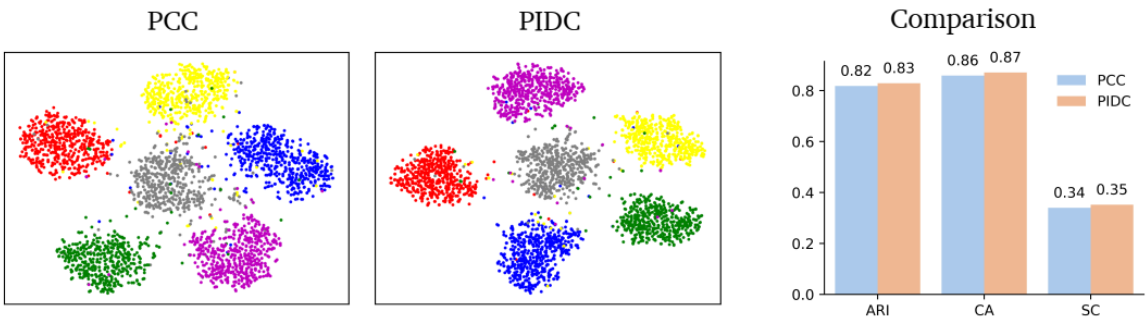

Figure S6. The comparison of different methods to construct gene-to-gene relationships (PCC and PIDC), Related to Figure 1-2. From the visualization and the clustering performance, we could find that the gene regulatory network inference tools such as PIDC could facilitates the imputation of scRNA-seq data using GraphSCI. We attribute the remarkable improvement to the accuracy of the input gene-to-gene relationships.

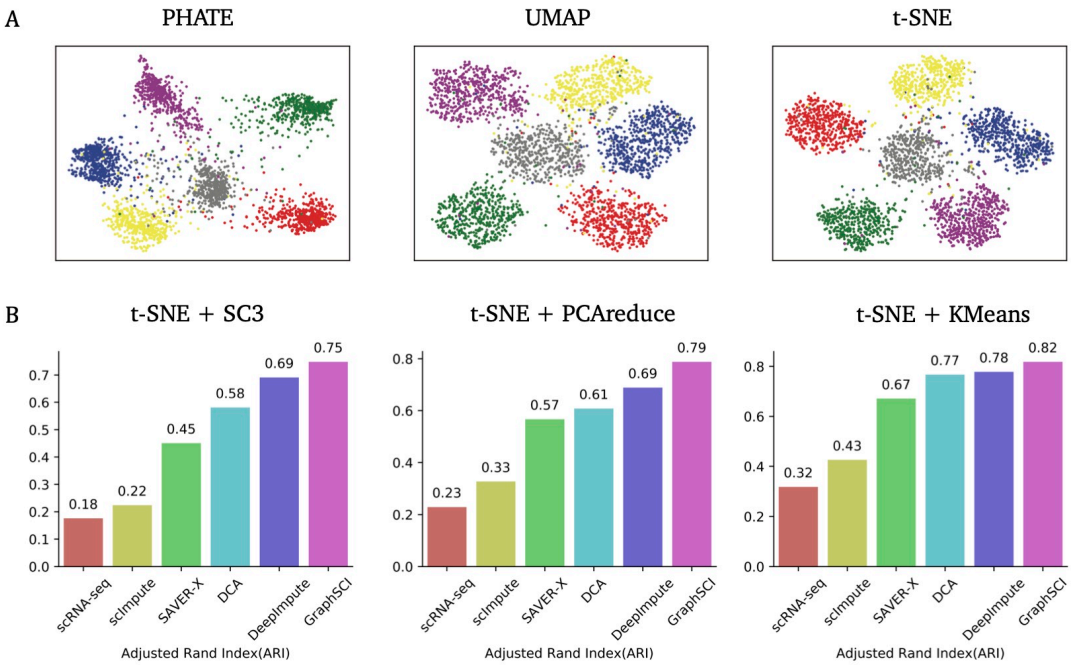

Figure S7. The comparison of different dimensional reduction algorithms and clustering approaches, Related to Figure 3. (A) We examined the influence of different cell visualization algorithms among UMAP, t-SNE, and PHATE from left to right. We could find that t-SNE showed better display results with closer inner-group distance and larger between-group distances. (B) We compared different clustering approaches (PCAreduce, SC3 and KMeans) through the clustering performance (ARI). We observed that GraphSCI consistently yields better performance with different clustering approaches, showing that our algorithm could achieve stable and better results under the same conditions. It again illustrates the rationality and effectiveness of our algorithm.

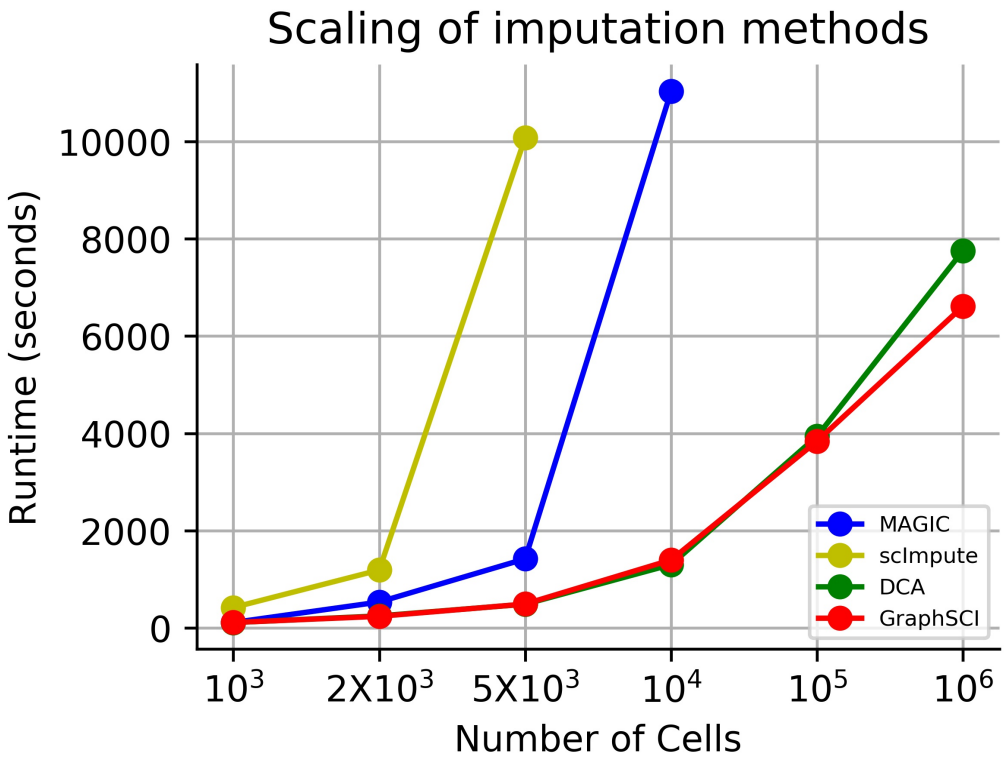

Figure S8. The runtimes for imputation with different numbers of cells down-sampled from 1.3 million mouse brain cells, Related to Figure 1-2. We analyzed the largest scRNA-seq data set in our experiments, which consists of 1.3 million mouse brain cells from 10X Genomics. The 1.3 million cell data matrix was down-sampled to 1,000, 2,000, 5,000, 10,000 and 100,000 cells and each subsampled matrix was imputed, and the runtime measured. We could find that the runtime of DCA and GraphSCI scaled linearly with the number of cells and the other methods took hours to impute 100,000 cells. It makes sense since DCA and GraphSCI are the neural network-based method that could be accelerated by GPU and the other methods failed to run due to the memory limitations on the large dataset.

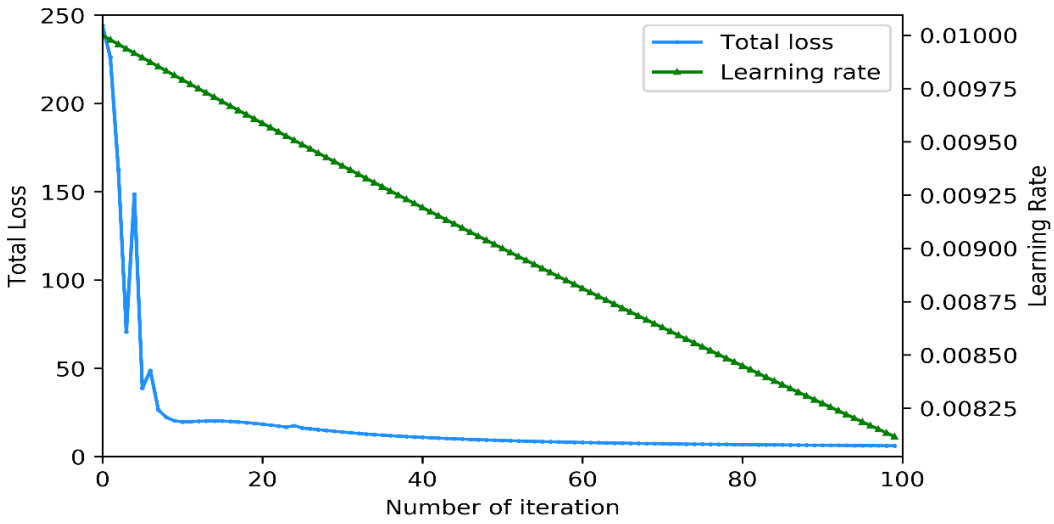

Figure S9. The optimization of our method, Related to Figure 1-2. We utilized the Adam optimizer with an initial learning rate of 0.01 and allowed it to decay exponentially with $\text{decay\_rate} = 0.9$ and $\text{decay\_steps} = 50$ during learning. The calculation of decayed learning rate in each step is: $\text{decayed\_learning\_rate} = \text{learning\_rate} * decay\_rate^{(step/decay\_steps)}$. The green line represents the decay trend of learning rate during training. The blue line illustrates the trend of total loss during training.

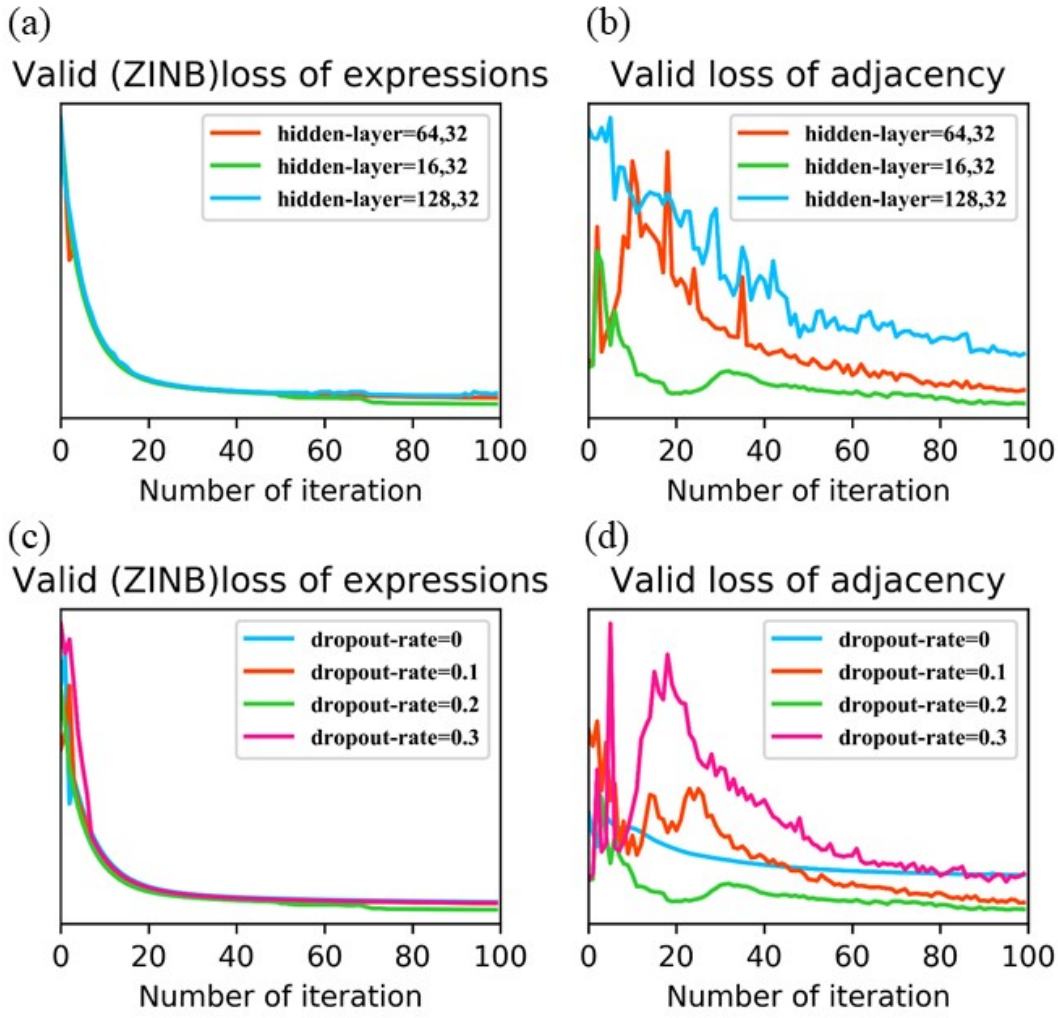

Figure S10. The exploration of Hyper-parameters, Related to Figure 1-2. During training, we randomly sampling 10% samples of each dataset as validation data and evaluate them in each iteration. The loss function of our method could be divided into two parts, one of which is the ZINB loss of gene expressions and the other is the cross entropy of gene-to-gene relationships. Due to the limitation of cluster metrics, we just utilize the losses of expressions and relationships on validation set to explore hyper-parameters in experiments. (a) is the ZINB loss of expressions on validation set with different size of hidden layers. (b) is the cross entropy of adjacency in validation with different size of hidden layers. (c) is the ZINB loss of expressions on validation set with different dropout rates. (d) is the cross entropy of adjacency on validation set with different dropout rates.

Supplemental tables

Table S1. The Results of SIM-T2 and SIM-T6 datasets, Related to Figure 3.

Datasets	Methods	Adjusted Rand Index (ARI)	Clustering Accuracy (CA)	Silhouette Coefficient (SC)
SIM_T2	GraphSCI	**0.977**	**0.994**	**0.609**
	DeepImpute	0.920	0.922	0.580
	DCA	0.914	0.925	0.582
	scImpute	0.779	0.528	0.382
	SAVER-X	0.845	0.654	0.449
	scRNA-seq	0.716	0.508	0.342
SIM_T6	GraphSCI	**0.818**	**0.859**	**0.340**
	DeepImpute	0.778	0.832	0.292
	DCA	0.767	0.842	0.284
	scImpute	0.426	0.562	0.098
	SAVER-X	0.671	0.752	0.254
	scRNA-seq	0.318	0.371	0.019

Table S2. The Results of Mouse ES, PBMC and Mouse Brain Cells datasets, Related to Figure 4-5.

Datasets	Methods	Adjusted Rand Index (ARI)	Clustering Accuracy (CA)	Silhouette Coefficient (SC)
Mouse ES	GraphSCI	**0.791**	**0.862**	**0.761**
	DeepImpute	0.762	0.833	0.673
	DCA	0.733	0.824	0.665
	scImpute	0.647	0.818	0.634
	SAVER-X	0.691	0.822	0.652
	scRNA-seq	0.393	0.754	0.423
PBMC	GraphSCI	**0.472**	**0.552**	**0.177**
	DeepImpute	0.464	0.548	0.169
	DCA	0.414	0.503	0.132
	scImpute	0.289	0.478	0.102
	SAVER-X	0.387	0.489	0.094
	scRNA-seq	0.312	0.457	0.071
Mouse Brain	GraphSCI	**0.316**	**0.422**	**0.030**
	DeepImpute	0.234	0.360	-0.090
	DCA	0.233	0.351	-0.050
	RAW	0.157	0.268	-0.170

Table S3. The summarization of datasets in this manuscript, Related to Figure 3-8

Datasets	Sample size / cell number	Number of genes	Number of cell types
SIM-T2	2000	3000	2
SIM-T6	3000	5000	6
C. elegans time-course	206	15855	-
Mouse ES cells	2717	24175	4
5K PBMC	5247	33570	11
10K Neuron Cells	11843	31053	16
Human ES cells	30	14766	-

Table S4. Main notations in our paper, Related to Figure 1-2.

Symbol	Description
$\mathcal{G}$	an undirected gene network with expressions and relations
$\mathcal{N}$	set of nodes (genes)
$\mathcal{M}$	set of scRNA-seq samples
$\mathcal{E}$	set of edges (gene-to-gene relationships)
$N = \lvert\mathcal{N}\rvert$	number of nodes (genes)
$M = \lvert\mathcal{M}\rvert$	number of samples
D	dimension of latent variables
$A \in \mathbb{R}^{N\times N}$	adjacency matrix of nodes
$X^C \in \mathbb{R}^{N\times M}$	raw gene expression matrix
$X \in \mathbb{R}^{N\times M}$	normalized gene expression matrix
$Z^{\mathcal{N}} \in \mathbb{R}^{N\times D}$	latent representation matrix for all nodes
$Z^{\mathcal{M}} \in \mathbb{R}^{M\times D}$	latent representation matrix for all samples
$\hat{A} \in \mathbb{R}^{N\times N}$	reconstructed adjacency matrix of nodes
$\hat{X} \in \mathbb{R}^{N\times M}$	imputed gene expression matrix

Transparent Methods

The proposed model GraphSCI imputes gene expression levels in scRNA-seq data based on a combination of the graph convolution network and Autoencoder neural network, with the input of gene expression matrix X and gene-to-gene relationships A. In our framework, GCN encodes the gene-to-gene network with expression matrix X to the latent vector Z and then reconstructs the edges in gene-to-gene network. AE encodes the gene expression matrix with gene-to-gene network and finally sample Z from ZINB or NB distributions to reconstruct gene expression matrix.

By using M single cells RNA-seq data with N genes, an undirected gene graph with gene expressions and gene-to-gene relationships can be constructed. Let $\mathcal{N}$ and $\mathcal{M}$ be a set of genes and samples respectively, an undirected gene graph can be denoted as $\mathcal{G} = (\mathcal{N}, \mathcal{M}, \mathcal{E})$, where $\mathcal{E}$ is the set of gene-to-gene relationships. Thus, we introduce an adjacency matrix $A \in \mathbb{R}^{N\times N}$ and a gene expression matrix $X \in \mathbb{R}^{N\times M}$ for $\mathcal{G}$, with A_{ij} representing the edge of the i-th gene and the j-th gene and X_{ij} being the expression value with rows representing genes and columns representing cells. Table S4 summarizes our main notations for scRNA-seq data.

Data processing and normalization. There are two inputs to our proposed model: (1) a gene expression matrix $X \in \mathbb{R}^{N\times M}$, (2) an adjacency matrix $A \in \mathbb{R}^{N\times N}$, and our final goal is to construct an imputed gene expression matrix $\hat{X}$ with the same dimensions. First, in raw scRNA-seq read count matrix X^C, genes with no reads in any cell would be filtered out. Then, the library size of cell i is denoted as l_i and is calculated as the total number of read counts of cell i. The size factor s_i of cell i is l_i divide by the median of total counts per cell. Therefore, we make a normalized matrix X by taking the log transformation with a pseudo count and scale of the read counts:

$$X_{ij} = \log\left(\frac{X_{ij}^C}{\sum_{k=1}^{N} X_{kj}} \times median(X_j) + 1\right) \quad (1)$$

where $i = 1,2,\dots,N$ representing each gene and $j = 1,2,\dots,M$ representing each sample.

Secondly, we attempt to obtain the adjacency matrix $A \in \mathbb{R}^{N\times N}$ from a graph where genes are nodes and edges indicate genes which are likely to be co-expressed. For the simulated datasets generated from Splatter(Zappia et al., 2017), we introduce the adjacency matrix $A \in \mathbb{R}^{N\times N}$ by Pearson correlation coefficient (PCC) as:

$$A_{ij} = \rho_{X_i,X_j} = \frac{Cov(X_i,X_j)}{\sigma_{X_i}\sigma_{X_j}}; i = 1,2,\dots,N;\ j = 1,2,\dots,N \quad (2)$$

where $Cov(X,Y)$ and σ_X is the covariance between X and Y and the standard deviation of X respectively.

Imputation based on graph convolution network. The preprocessed gene expression matrix and adjacency matrix are treated as the input for GraphSCI. Two neural network models, i.e., the inference model f_ϕ and the generative model g_φ were used to constructed the model for the probabilistic encoder q_ϕ and probabilistic decoder p_φ respectively, to preform gradient descent for learning all trainable parameters.

To infer the embeddings of cells and genes, we apply a two-layer graph convolution network and a two-layer fully connected neural network mapping the adjacency matrix A and the gene expression matrix X to the low-dimensional representations of the posterior distribution (i.e. Gaussian distributions and ZINB distributions) respectively. In particular, the two-layer GCN is defined as:

$$H_{\mathcal{N}}^{(1)} = ReLU(\tilde{A}XW_{\mathcal{N}}^{(0)}) \tag{3}$$

$$[\mu_{\mathcal{N}}, \sigma_{\mathcal{N}}^2] = \tilde{A}H_{\mathcal{N}}^{(1)}W_{\mathcal{N}}^{(1)} \tag{4}$$

where $\mu_{\mathcal{N}}$ and $\sigma_{\mathcal{N}}^2$ are the mean and variances of the learned Gaussian distribution parameters, $ReLU(\cdot) = \max(0, \cdot)$ is the non-linear activation function, $\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the symmetrically normalized adjacency matrix with the $\mathcal{G}'$s degree matrix $D_{ii} = \sum_j A_{ij}$, and $\phi = [W_{\mathcal{N}}^{(0)}, W_{\mathcal{N}}^{(1)}]$ are the trainable parameters of GCN layers.

The two-layer fully connected layers for inferring ZINB distribution of single cell samples are defined as:

$$H_{\mathcal{M}}^{(1)} = \tanh\left(X^T\left(W_{\mathcal{M}}^{(0)} \odot A\right) + b^{(0)}\right) \tag{5}$$

$$[\mu_{\mathcal{M}}, \theta_{\mathcal{M}}, \pi_{\mathcal{M}}] = \sigma(H_{\mathcal{M}}^{(1)}W_{\mathcal{M}}^{(1)} + b^{(1)}) \tag{6}$$

where $\mu_{\mathcal{M}}, \theta_{\mathcal{M}}$ and $\pi_{\mathcal{M}}$ are the parameters of the ZINB distribution: mean, dispersion and dropout probability, the operation $\odot$ is the Hadamard (element-wise) product, $\tanh(\cdot)$ and $\sigma(\cdot)$ are the activation functions and $\phi = [W_{\mathcal{M}}^{(0)}, W_{\mathcal{M}}^{(1)}, b^{(0)}, b^{(1)}]$ are the trainable parameters of fully connected layers.

In particularly, the ZINB distribution is applied for count data that exhibit over-dispersion and excess zeros, which is parameterized with the mean (μ) and dispersion (θ) of the negative binomial distribution as well as the dropout probability (π) representing the probability of zeros (dropout events). But droplet-based scRNA-seq (such as 10X) are supposed to follow a NB distribution. A count matrix X that is ZINB-distributed with (μ, θ, π) or NB-distributed with (μ, θ) are denoted as:

$$NB(X|\mu;\theta) = \frac{\Gamma(X+\theta)}{\Gamma(\theta)\Gamma(X+1)}\left(\frac{\theta}{\theta+\mu}\right)^{\theta}\left(\frac{\mu}{\theta+\mu}\right)^{X} \tag{7}$$

$$ZINB(X|\mu;\theta;\pi) = \pi\delta_0(X) + (1-\pi)NB(X|\mu;\theta) \tag{8}$$

where $\Gamma(\mathrm{x})$ and $\delta_0(x)$ is the Gamma function and Dirac function respectively. Therefore, we could estimate the parameters μ, θ, π of ZINB distribution from the hidden layer in Eq. (6):

$$\mu_{\mathcal{M}} = \exp\left(H_{\mathcal{M}}^{(1)}W_{\mathcal{M}}^{(1)} + b^{(1)}\right) \tag{9}$$

$$\theta_{\mathcal{M}} = softplus(H_{\mathcal{M}}^{(1)}W_{\mathcal{M}}^{(1)} + b^{(1)}) \tag{10}$$

$$\pi_{\mathcal{M}} = sigmoid(H_{\mathcal{M}}^{(1)}W_{\mathcal{M}}^{(1)} + b^{(1)}) \tag{11}$$

where $\exp(\cdot)$ is the exponential function and $\mathrm{softplus}(\cdot)$ and $\mathrm{sigmoid}(\cdot)$ are the non-linear activation functions.

After having obtained the parameters of the learned distributions, the reparameterization method could help us transform the latent variables $([\mu_{\mathcal{N}}, \sigma^2_{\mathcal{N}}], [\mu_{\mathcal{M}}, \theta_{\mathcal{M}}, \pi_{\mathcal{M}}])$ to deterministic variables, denoted as $Z^{\mathcal{N}}, Z^{\mathcal{M}}$. Therefore, the generative model in our framework could decode from the deterministic variables $Z^{\mathcal{N}}$ and $Z^{\mathcal{M}}$ to generative random variables, where the gene expressions and gene-to-gene relationships can be reconstructed.

Specifically, given embeddings of gene i and cells j, we compute $\mu'_{\mathcal{M}}, \theta'_{\mathcal{M}}$ and $\pi'_{\mathcal{M}}$ by:

$$[\mu'_{\mathcal{M}}, \theta'_{\mathcal{M}}, \pi'_{\mathcal{M}}] = g_{\varphi_1}(Z_i^{\mathcal{N}}, Z_j^{\mathcal{M}}) \tag{12}$$

where g_{φ_1} is a neural network for reconstructing gene expression matrix and φ_1 is the trainable parameter in g_{φ_1}. Then an imputed gene expression $\hat{X}_{ij}$ can be generated by the following process:

$$p_{\varphi_1}\left(\hat{X}_{ij} \middle| Z_i^{\mathcal{N}}, Z_j^{\mathcal{M}}\right) = ZINB\left(\mu'_{\mathcal{M}(i,j)}, \theta'_{\mathcal{M}(i,j)}, \pi'_{\mathcal{M}(i,j)}\right) \tag{13}$$

$$p_{\varphi_1}\left(\hat{X}_{ij} \middle| Z_i^{\mathcal{N}}, Z_j^{\mathcal{M}}\right) = NB\left(\mu'_{\mathcal{M}(i,j)}, \theta'_{\mathcal{M}(i,j)}\right) \tag{14}$$

where $ZINB\left(\mu'_{\mathcal{M}(i,j)}, \theta'_{\mathcal{M}(i,j)}, \pi'_{\mathcal{M}(i,j)}\right)$ is the ZINB distribution parameterized by $\mu'_{\mathcal{M}(i,j)}, \theta'_{\mathcal{M}(i,j)}$ and $\pi'_{\mathcal{M}(i,j)}$, $NB\left(\mu'_{\mathcal{M}(i,j)}, \theta'_{\mathcal{M}(i,j)}\right)$ is the NB distribution parameterized by $\mu'_{\mathcal{M}(i,j)}$ and $\theta'_{\mathcal{M}(i,j)}$, and p_{φ_1} is the probabilistic decoder given the latent embeddings $Z_i^{\mathcal{N}}$ and $Z_j^{\mathcal{M}}$.

Therefore, we could implement the generative model g_{φ_1} by:

$$\hat{X}_{ij} = g_{\varphi_1}\left(Z_i^{\mathcal{N}}, Z_j^{\mathcal{M}}\right) = diag(\vec{s}_j) \times Z_j^{\mathcal{M}} \tag{15}$$

where $\mathrm{diag}(\cdot)$ is the diagonal matrix constructed by the vector $(\cdot)$ and $\vec{s}_j$ is the size factor of cell j.

Similarly, given embeddings of two genes i and j, we can compute $\mu'_{\mathcal{N}}$ and $\sigma'^2_{\mathcal{N}}$ by:

$$\left[\mu'_{\mathcal{N}}, \sigma'^2_{\mathcal{N}}\right] = g_{\varphi_2}(Z_i^{\mathcal{N}}, Z_j^{\mathcal{N}}) \tag{16}$$

where g_{φ_2} is a neural network for reconstructing gene-to-gene relationships and φ_2 is the trainable parameter in g_{φ_2}. Then an observed edge between two genes i and j can be generated by:

$$p_{\varphi_2}\left(\hat{A}_{ij} \middle| Z_i^{\mathcal{N}}, Z_j^{\mathcal{N}}\right) = Gaussian(\mu'_{\mathcal{N}(i,j)}, \sigma'^2_{\mathcal{N}(i,j)}) \tag{17}$$

where $Gaussian(\mu'_{\mathcal{N}(i,j)}, \sigma'^2_{\mathcal{N}(i,j)})$ is the Gaussian distribution parameterized by $\mu'_{\mathcal{N}(i,j)}$ and $\sigma'^2_{\mathcal{N}(i,j)}$ and p_{φ_2} is the probabilistic decoder given the latent embeddings $Z_i^{\mathcal{N}}$ and $Z_j^{\mathcal{N}}$.

The generative model g_{φ_2} to reconstruct gene-to-gene relationships could be defined as:

$$\hat{A}_{ij} = g_{\varphi_2}\left(Z_i^{\mathcal{N}}, Z_j^{\mathcal{N}}\right) = sigmoid(Z_i^{\mathcal{N}^T} Z_j^{\mathcal{N}}) \tag{18}$$

where sigmoid(·) is the sigmoid function.

Optimization. The optimization was performed to obtain accurate embeddings of both genes and cells in an unsupervised way. For this purpose, $Z^{\mathcal{N}}$ and $Z^{\mathcal{M}}$ were optimized by the variational lower bound $\mathcal{L}$:

$$\mathcal{L}(\phi,\varphi) \triangleq \mathbb{E}_{q_\phi}\left[\sum_{i\in\mathcal{N},j\in\mathcal{M}} \log p_{\varphi_1}\left(\hat{X}_{ij}\middle|Z_i^{\mathcal{N}},Z_j^{\mathcal{M}}\right)\right] + \mathbb{E}_{q_\phi}\left[\log \sum_{i,j\in\mathcal{N}} \log p_{\varphi_2}\left(\hat{A}_{ij}\middle|Z_i^{\mathcal{N}},Z_j^{\mathcal{N}}\right)\right]$$
$$- D_{KL}\left(q_\phi(Z^{\mathcal{M}}|A,X^T)||p(Z^{\mathcal{M}})\right) - D_{KL}(q_\phi(Z^{\mathcal{N}}|A,X)\Big|\Big|p(Z^{\mathcal{N}})\Big). \quad (19)$$

where $\mathbb{E}_{q_\phi}$ is the cross entropy function with the probabilistic distribution q_ϕ and p_φ and $D_{KL}(q||p) = \sum p(\cdot)\log\frac{p(\cdot)}{q(\cdot)}$ is the Kullback-Leibler (KL) divergence between q(·) and p(·). In the above equation, $q_\phi(Z^{\mathcal{M}}|A,X^T)$ and $q_\phi(Z^{\mathcal{N}}|A,X)$ is defined as the probabilistic encoder with the input of A,X^T and A,X respectively, aiming at producing the representations $Z^{\mathcal{N}},Z^{\mathcal{M}}$. Similarly, $p_{\varphi_1}\left(\hat{X}_{ij}\middle|Z_i^{\mathcal{N}},Z_j^{\mathcal{M}}\right)$ and $p_{\varphi_2}\left(\hat{A}_{ij}\middle|Z_i^{\mathcal{N}},Z_j^{\mathcal{N}}\right)$ are the probabilistic decoders for construct the imputed gene expression matrix $\hat{X}$ and gene-to-gene relationships $\hat{A}$. Furthermore, the KL divergence in optimization could be interpreted as the regularization to make the predicted posterior distributions closer to the prior distributions $p(Z^{\mathcal{M}}),p(Z^{\mathcal{N}})$.

With the help of reparameterization trick, we could represent the distributions with deterministic variables:

$$[\mu_{\mathcal{M}},\theta_{\mathcal{M}},\pi_{\mathcal{M}}] \in ZINB(X|\mu_{\mathcal{M}},\theta_{\mathcal{M}},\pi_{\mathcal{M}}) \text{ or } [\mu_{\mathcal{M}},\theta_{\mathcal{M}}] \in NB(X|\mu_{\mathcal{M}},\theta_{\mathcal{M}}) \quad (20)$$

$$[\mu_{\mathcal{N}},\sigma_{\mathcal{N}}^2] \in Gaussian(\mu_{\mathcal{N}},\sigma_{\mathcal{N}}^2) \quad (21)$$

These deterministic variables are differentiable and capable to be calculated in backpropagation process. We could directly derivate Eq. (18) based on Monte Carlo estimates:

$$\begin{aligned}\mathcal{L}(\phi,\varphi) = {} & \frac{1}{NML}\sum_{l=1}^{L}\left(\sum_{i\in\mathcal{N},j\in\mathcal{M}}\log p_{\varphi_1}\left(X_{ij}\middle|Z_i^{\mathcal{N}^{(l)}},Z_j^{\mathcal{M}^{(l)}}\right)\right)\\ & + \frac{1}{N^2L}\sum_{l=1}^{L}\left(\sum_{i,j\in\mathcal{N}}\log p_{\varphi_2}\left(A_{ij}\middle|Z_i^{\mathcal{N}^{(l)}},Z_j^{\mathcal{N}^{(l)}}\right)\right)\\ & - \frac{1}{2M}\sum_{j\in\mathcal{M}}\sum_{d=1}^{D}\left(\pi_{\mathcal{M}}\delta_0\left(Z^{(d)}\right) + (1-\pi_{\mathcal{M}})NB(Z^{(d)}|\mu_{\mathcal{M}};\theta_{\mathcal{M}})\right)\\ & - \frac{3}{2N}\sum_{i\in\mathcal{N}}\sum_{d=1}^{D}\left(1+\log\sigma^2_{\mathcal{N}_{(i)}^{(d)}} - \sigma^2_{\mathcal{N}_{(i)}^{(d)}} - \mu^2_{\mathcal{N}_{(i)}^{(d)}}\right)\end{aligned} \quad (22)$$

Therefore, with the optimization, the gradient-based optimization techniques can be used to train the end-to-end model.

Evaluation metrics. To evaluate the accuracy of imputation, we examine the reconstruction accuracy and clustering performance to the scRNA-seq datasets. The reconstruction accuracy on the simulated dataset can be measured by mean absolute error (MAE), which is the reconstruction error between the true expression matrix and imputed matrix. Clustering performance can be measured by the clustering metrics: adjusted Rand index (ARI)(Hubert and Arabie, 1985), clustering accuracy (CA) and Silhouette

Coefficient(Rousseeuw, 1987) (SC). To fairly quantitate the performance of differentially expressed genes (DEGs) detection using scRNA-seq data, we calculated the accuracy (ACC), F-score and AUC for each DEG detection.

The adjusted Rand index (ARI) is the corrected-for-chance version of the Rand index. The Rand index is a measure of the similarity between two data clustering and the ARI is adjusted for the chance grouping of elements. Given a set of n samples, the two clusters of these samples are $V = \{V_1, V_2, \dots, V_r\}$ and $U = \{U_1, U_2, \dots, U_t\}$ and n_{ij} is defined as $n_{ij} = |V_i \cap U_j|$. Let $a_i = \sum_{j=1}^{t} n_{ij}, i = 1, \dots, r$ and $b_j = \sum_{i=1}^{r} n_{ij}, j = 1, \dots, t$, the ARI could be defined as

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}} \tag{23}$$

The CA is defined as the accuracy of the clustering assignments. Given a sample i, let s_i be the ground truth label and r_i be the assignments of clustering, then the CA is

$$CA = \max_m \frac{\sum_{i=1}^{n} \delta(s_i, m(r_i))}{n} \tag{24}$$

where n is the number of samples, m is the set of one-by-one mapping between clustering assignments and true labels and $\delta(\mathrm{x}, \mathrm{y})$ = 1 if $\mathrm{x} = \mathrm{y}$ otherwise 0.

The SC measured the similarity between a single cell and its cluster. The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster. It could be defined as

$$SC = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{25}$$

where $a(i)$ is the mean distance between sample i and all other samples in the same cluster and $b(i)$ is the minimum distance of sample i to all points in any other cluster.

In the experiments of differential expression analysis, we took the DEG detection as the problem of predicting a gene is DEG or not, and the gold standard are obtained from bulk RNA-seq. Therefore, the accuracy (ACC), F-score and AUC could be calculated by:

$$ACC = \frac{the\ gene\ is\ DEG}{DEGs} \times 100\% \tag{26}$$

The F-score is calculated from the precision and recall of the DEG predictions, where the precision is the number of correctly detected genes divided by the number of all DEGs and the recall is the number of correctly detected genes divided by the number of all DEGs that should have been detected. It could be defined as:

$$F1 - score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{27}$$

where TP is the true positive meaning that the correct DEG have been detected, FP is the false positives and FN is the false negatives.

AUC - ROC curve is a performance measurement for the classification problems at various threshold settings, which could be applied to evaluate the detection of DEGs. The AUC is calculated by the area under the ROC-curve, which represents the degree or measure of separability.

Simulated datasets. Our simulated data are generated by Splatter(Zappia et al., 2017) R package, a widely used package for simulating the scRNA-seq count data. First, we simulated a dataset with two cell groups, 2000 cells of 3000 genes by setting 27% of data values to zero mimicking dropout events. During the simulation, we set the parameter $dropout.shape = -1$, $dropout.mid = 0$ and $de.fracScale = 0.3$ for simulating the dropout events and the other parameters are set to default values. Hence, we could obtain the true counts before dropout and the raw counts after dropout, which are the simulated scRNA-seq data. Furthermore, we simulated a complex dataset of 3000 cells by 5000 genes to evaluate the robustness of our model, The 3000 cells are divided into six groups and the parameter were set to $dropout.shape = -1$, $dropout.mid = 0$, $de.fracScale = 0.3$ and the other parameters with default values.

C. elegans time course experimental data. We obtain the bulk transcriptomics data from the supplementary material of Francesconi. et al, which contains 15855 detected genes during 12 hours of C. elegans development(Francesconi and Lehner, 2014). We analyzed the dataset after simulating single-cell transcriptomics dropout noises and the bulk transcriptomics data can be the ground truth for evaluation. Hence, we compared our method with the existing method DCA(Eraslan et al., 2019) by Pearson correlation coefficient.

Mouse embryonic stem cells data. Klein. et al. profiled the single-cell transcriptomics by droplet-microfluidic approach and applied it on embryonic stem cells(Klein et al., 2015). They analyzed the heterogeneity of mouse embryonic stem cells differentiation after leukemia inhibitory factor (LIF) withdrawal. Here, we selected the four different LIF withdrawal intervals (0, 2, 4, 7 days) and construct a scRNA-seq dataset with 2717 cells of 24175 detected genes. And the cell types are determined by the intervals of LIF withdrawal.

Human ESC scRNA-seq dataset for differential expression analysis. Chu et al generated bulk and scRNA-seq data from H1 human embryonic stem cells (H1) differentiated into definitive endoderm cells (DEC). This dataset contains six samples of bulk RNA-seq (four for H1 ESC and two for DEC) and scRNA-seq of 350 single cells (212 for H1 ESC and 138 for DEC). The percentage of zero expression is 14.8% for the bulk RNA-Seq dataset and 49.1% for the scRNA-Seq dataset.

5k peripheral blood mononuclear cells (PBMC) from a healthy donor. The dataset was provided by 10X scRNA-seq platform(Zheng et al., 2017), profiling the transcriptome of the peripheral blood mononuclear cells (PBMCs) from a healthy donor. The total number of cells was 5247 after filtering process and the cell types were identified by graph-based clustering on the platform.

10K Brain Cells from an E18 Mouse dataset. The dataset was also provided by 10X scRNA-seq platform, profiling the brain cells from a combined cortex, hippocampus and sub ventricular zone of an E18 mouse. We could obtain the dataset containing 11843 mouse brain cells of 31053 detected genes and the cell types were identified by graph-based clustering on the platform.

Human Embryos cells scRNA-seq data. Xue et al. performed a comprehensive analysis of transcriptome dynamics by weighted gene co-expression network analysis(Xue et al., 2013). Therefore, we could obtain the dataset containing 30 samples from oocyte to morula in human embryos samples from their experiments. Here, we utilized the dataset to demonstrate the effectiveness of our method on inferring the gene-to-gene relationships.

Implementation. We implemented the proposed model with Tensorflow 1.11.0(Abadi et al., 2016). In the training process, we utilized the Adam(Kingma and Ba, 2014) optimizer with an initial learning rate of 0.01 and allowed it to decay exponentially with $\mathrm{decay\_rate} = 0.9$ and $\mathrm{decay\_steps} = 50$ during learning. The total loss and learning rate decreased with epoch during training as shown in supplementary Fig. 4. The hidden layers of encoders were set as 16 neurons and we use a 32-dimensional of embedding latent variables in all experiments, denoted as D. To alleviate overfitting, we implemented the regularization methods such as dropout and L2 regularization. Dropout(Srivastava et al., 2014) rate 0.2 was applied on the inference model and the coefficient of L2 regularization was 0.001. We explored hyper-parameters in a wide range and find the above hyper-parameters yields the highest performance, as supplementary Fig. 5 shown. We tuned model hyper-parameters based on the experimental results on simulated datasets and used them across all datasets. All experiments are repeated for 5 times, each with a different random seed.

Supplemental References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G. & Isard, M. Tensorflow: A system for large-scale machine learning. 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), 2016. 265-283.

Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. 2019. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature communications,* 10(1)**,** pp 390.

Francesconi, M. & Lehner, B. 2014. The effects of genetic variation on gene expression dynamics during development. *Nature,* 505(7482)**,** pp 208.

Hubert, L. & Arabie, P. 1985. Comparing partitions. *Journal of classification,* 2(1)**,** pp 193-218.

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

Klein, A. M., Mazutis, L., Akartuna, I., Tallapragada, N., Veres, A., Li, V., Peshkin, L., Weitz, D. A. & Kirschner, M. W. 2015. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell,* 161(5)**,** pp 1187-1201.

Rousseeuw, P. J. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics,* 20(53-65.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research,* 15(1)**,** pp 1929-1958.

Xue, Z., Huang, K., Cai, C., Cai, L., Jiang, C.-y., Feng, Y., Liu, Z., Zeng, Q., Cheng, L. & Sun, Y. E. 2013. Genetic programs in human and mouse early embryos revealed by single-cell RNA sequencing. *Nature,* 500(7464)**,** pp 593.

Zappia, L., Phipson, B. & Oshlack, A. 2017. Splatter: simulation of single-cell RNA sequencing data. *Genome biology,* 18(1)**,** pp 174.

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