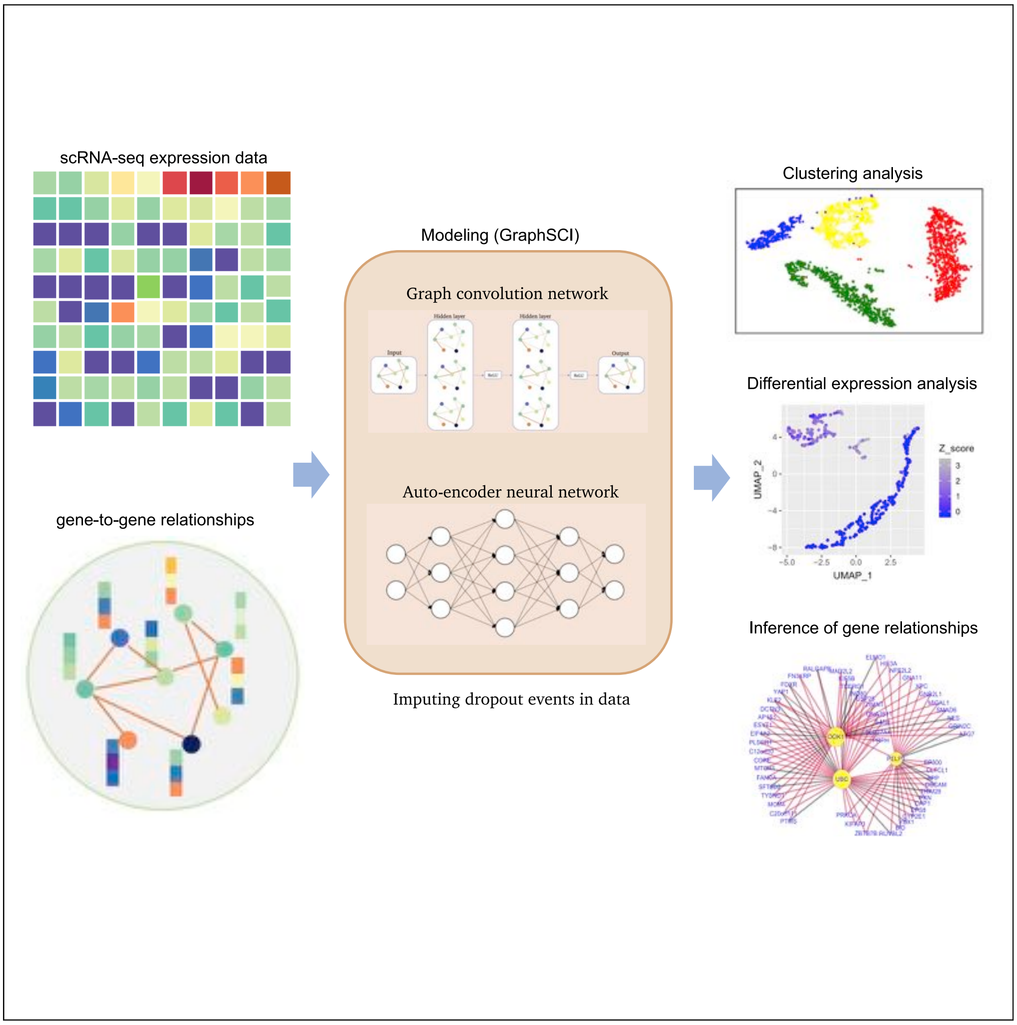
iScience

CellPress
OPEN ACCESS

Article

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


scRNA-seq expression data
gene-to-gene relationships
Modeling (GraphSCI)
Graph convolution network
Input
Hidden layer
Hidden layer
Output
Auto-encoder neural network
Imputing dropout events in data
Clustering analysis
Differential expression analysis
UMAP_2
UMAP_1
Z_score
Inference of gene relationships

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

yangyd25@mail.sysu.edu.cn

Highlights

Graph convolution network is used to impute the dropout events in scRNA-seq data

GraphSCI recovers transcriptome dynamics in scRNA-seq data sets effectively

GraphSCI improves various downstream analyses on scRNA-seq data significantly

GraphSCI is able to accurately infer gene-to-gene relationships during imputation

Rao et al., iScience *24*, 102393
May 21, 2021 © 2021 The Author(s).
https://doi.org/10.1016/j.isci.2021.102393

iScience

CellPress
OPEN ACCESS

Article

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

Jiahua Rao,[1] Xiang Zhou,[1] Yutong Lu,[1] Huiying Zhao,[2] and Yuedong Yang[1,3,4,*]

SUMMARY

Single-cell RNA sequencing technology promotes the profiling of single-cell transcriptomes at an unprecedented throughput and resolution. However, in scRNA-seq studies, only a low amount of sequenced mRNA in each cell leads to missing detection for a portion of mRNA molecules, i.e. the dropout problem which hinders various downstream analyses. Therefore, it is necessary to develop robust and effective imputation methods for the increasing scRNA-seq data. In this study, we have developed an imputation method (GraphSCI) to impute the dropout events in scRNA-seq data based on the graph convolution networks. Extensive experiments demonstrated that GraphSCI outperforms other state-of-the-art methods for imputation on both simulated and real scRNA-seq data. Meanwhile, GraphSCI is able to accurately infer gene-to-gene relationships and the inferred gene-to-gene relationships could also provide powerful assistance for imputation dynamically during the training process, which is a key promotion of GraphSCI compared with other imputation algorithms.

INTRODUCTION

Compared to bulk cell RNA sequencing (Wang et al., 2009) (RNA-seq), single-cell RNA sequencing technology (Kolodziejczyk et al., 2015) (scRNA-seq) has greatly promoted the profiling of transcriptomes at single-cell level and helped researchers to improve understanding of complex biological questions. It allows people to study cell-to-cell variability at a much higher throughput and resolution, such as studies of cell heterogeneity, differentiation and developmental trajectories (Saliba et al., 2014).

Despite its improvements, various technical deviations occurred due to the upgrade of sequencing techniques from bulk samples to single cells. Typically, the low RNA capture rate and sequencing efficiency lead to a large proportion of expressed genes with false zero counts in some cells, defined as 'dropout' event (Svensson et al., 2017; Kharchenko et al., 2014). For example, protocols based on droplet microfluidics (Zilionis et al., 2017) and Fluidigm C1 platform usually have a high dropout rate in the scRNA-seq data due to their technical limitations. And new droplet-based protocols, such as inDrop (Klein et al., 2015) and 10X Genomics (Zheng et al., 2017), have improved the detection rates but still have relatively low sensitivity, leading to the dropout events. On the other hand, although many of the zero counts represent the true absence of gene expression in specific cells, a considerable fraction is due to the dropout phenomenon where a truly expressed gene is undetected in some cells, resulting in zero or low read counts. Therefore, it is important to note the distinction between the truly expressed zeros and the false zeros in statistical analysis. Not all zeros can be considered as the missing values to be imputed. Imputation methods should impute the non-zero space but preserve the true-zero expression.

As a result, methods such as MAGIC (Van Dijk et al., 2018), SAVER (Huang et al., 2018), scImpute (Li and Li, 2018), scVI (Lopez et al., 2018), DCA (Eraslan et al., 2019), and DeepImpute (Arisdakessian et al., 2019) have been developed to correct the false zero read counts in order to recover true expression levels in scRNA-seq data. These approaches estimate ''corrected'' gene expressions by borrowing information across similar genes or cells. For example, MAGIC imputes gene expression data for each gene across similar cells based on Markov transition matrix, while SAVER takes advantage of gene-to-gene relationships by using Bayesian approach to infer the denoised expression. Both MAGIC and SAVER would recover the expression level of each gene in each cell including those unaffected by dropout events. ScImpute, on the other

[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510000, China

[2] Sun Yat-sen Memorial Hospital, Sun Yat-sen University, Guangzhou 510000, China

[3] Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Ministry of Education, Guangzhou 510000, China

[4] Lead contact

*Correspondence: yangyd25@mail.sysu.edu.cn

https://doi.org/10.1016/j.isci.2021.102393

hand, determines the dropout entries based on a mixture model and imputes only the likely dropout entries across similar cells. However, MAGIC and SAVER fail to learn the non-linear relationships and the counting structures in the scRNA-seq data. Thus, with the development of deep learning, neural network-based imputation methods have been proposed such as SAVER-X (Wang et al., 2019), DCA and DeepImpute. By combining a deep autoencoder with a Bayesian model, SAVER-X extracts transferable gene−gene relationships to impute scRNA-seq data sets. DCA proposes an imputation method based on an Autoencoder, a kind of deep neural networks used to reconstruct data in an unsupervised manner. DeepImpute, in another way, constructs multiple sub-neural networks to impute genes in a divide-and-conquer approach, which utilizes dropout layers and loss functions to learn patterns in the data. In the imputation of each cell, DCA minimizes the zero-inflated negative binomial (ZINB) (Risso et al., 2018) model-based loss function to learn gene-specific distribution in scRNA-seq data.

However, these existing imputation methods for scRNA-seq aim at learning the similarity of cells or genes but not considering gene-to-gene relationships and cell-to-cell correlations simultaneously, resulting in the fact that they cannot retain biological variation across cells or genes. And decades of molecular biology research have taught us much about the principles of gene interaction and their influence on gene expression (Bhardwaj and Lu, 2005; Fraser et al., 2004). For example, the gene is truly not expressed due to gene regulation, but imputed by similar cells, which makes it difficult to study cell-to-cell variation and downstream analysis. This means that our imputation method not only needs to take advantage of the information between similar cells but also gene-to-gene relationships. More importantly, as imputation proceeds, the imputed gene expression matrix could infer more accurate gene-to-gene relationships while the inferred gene-to-gene relationship helps improve the accuracy of imputation. Therefore, our imputation method needs to be able to dynamically integrate the imputation of gene expressions and inference of the gene-to-gene relationships during the training process.

Accordingly, in this paper, we developed a **S**ingle-**C**ell **I**mputation method that combines Graph convolution network (GCN) and Autoencoder neural networks, called GraphSCI, to impute the dropout events in scRNA-seq by systematically integrating the gene expression with gene-to-gene relationships. We will use gene-to-gene relationships as prior knowledge to recover gene expression in a single cell because gene-to-gene interactions are likely to affect gene expression sensitively. And the combination of GCN and autoencoder neural networks makes it possible for us to dynamically utilize the increasingly accurate gene-to-gene relationships to impute gene expressions. By stacking the GCN and autoencoder network, GraphSCI is capable of exploring the gene-to-gene relationships in an explicit way, so as to impute the dropout events effectively. Furthermore, the deep generative model with gene-specific distribution such as ZINB and NB distribution could learn the true data distribution of scRNA-seq data and then impute the dropout events and avoid overfitting.

The gene-to-gene relationships can be regarded as a gene graph, in which the gene is the node and the edge is the relationship. As a consequence, the imputation task of gene expression can be converted into the node recovering problem on graphs. GCN (Kipf and Welling, 2017) is a very powerful neural network architecture for machine learning on graphs. It was designed to learn hidden layer representations that encode both local graph structure and features of nodes and edges. A number of recent studies describe applications of GCN such as node recovering problem (Meng et al., 2019; Gong et al., 2014; Chakrabarti et al., 2014; Yang et al., 2017). Inspired by the co-embedding attributed network (Meng et al., 2019), we combine GCN and autoencoder neural network to systematically learn the low-dimensional embedded representations of genes and cells. GCN exploits the spatial feature of gene-to-gene relationships effectively while Autoencoder neural network learns the non-linear relationships of cells and counting structures of scRNA-seq data, and thus the deep learning framework reconstructs gene expressions by integrating gene expressions and gene-to-gene relationships dynamically in the backward propagation of neural networks.

Our proposed method was shown to outperform competing methods over both simulated and real data sets by diverse downstream analyses. To assess the performance of the imputation methods, we evaluate their improvement on several downstream analyses. Firstly, we perform cell clustering and use clustering metrics to demonstrate their effectiveness to impute the dropout events. And then we also perform the differential expression analysis to evaluate their improvement of the identification of differentially expressed genes (DEGs). The evaluation performance illustrates the rationality and effectiveness of our

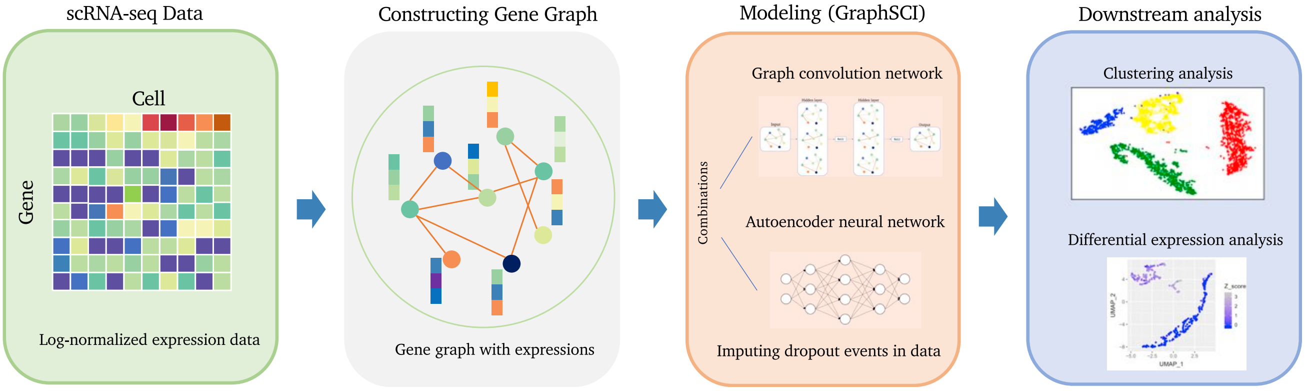

scRNA-seq Data
Cell
Gene
Log-normalized expression data
Constructing Gene Graph
Gene graph with expressions
Modeling (GraphSCI)
Graph convolution network
Combinations
Autoencoder neural network
Imputing dropout events in data
Downstream analysis
Clustering analysis
Differential expression analysis

Figure 1. The overview of GraphSCI algorithm

The input of GraphSCI framework is a gene expression matrix from scRNA-seq, and we construct the gene graph from the raw expression data through PCC. And GraphSCI combines the graph convolution network and autoencoder neural network to impute the dropout events in data. Finally, Extensive downstream analysis experiments demonstrated the effective and robustness of GraphSCI.

proposed method. Furthermore, our method takes advantage of the gene-to-gene relationships in the framework that infers new reliable relationships simultaneously. Altogether, we demonstrate that our proposed method is highly scalable and parallelizable via graphical processing units (GPUs).

RESULTS

Overview of the GraphSCI algorithm

GraphSCI is a deep neural network model that combines the GCN and autoencoder neural network to impute gene expression levels in scRNA-seq data. The overview of our method is shown in Figure 1 and the detailed model architecture is shown in Figure 2.

Usually, the expressions of genes are correlated by their related genes or interacting genes because the co-expressed genes are controlled by the same transcriptional regulatory program, functionally related, or members of the same pathway or protein complex (Weirauch, 2011). Therefore, given the log-normalized expression data X, we first construct gene co-expression networks, called Gene Graph, from the raw expression data through the Pearson correlation coefficient (PCC). When the PCC between two genes is greater than 0.3 or less than -0.3, we assume that the two genes are co-expressed and there is an initial edge between them in the gene network. Obviously, our initial gene co-expression networks have a high rate of false positives because of dropout events in scRNA-seq data. We therefore combine the GCN and autoencoder neural network (AE) to dynamically integrate the imputation of gene expression and the inference of the gene co-expressed network where GCN encodes the gene co-expressed network with expression levels to the latent vector Z and then reconstructs the edges in gene co-expression network. AE encodes the gene expression matrix with gene co-expression network and finally sample Z from ZINB or NB distributions to reconstruct gene expression matrix.

This model enables us to utilize gene-to-gene relationships to impute the dropout events and further refine the gene co-expression network. The gene-to-gene network is an undirected graph, where each node corresponds to a gene and each edge between two genes indicates there is a significant co-expression relationship between them (Stuart et al., 2003). And the excellent characteristics of GCN allow us to regard the gene expression levels in different samples as node (gene) features in gene co-expression network and utilize them in the learning of gene network.

GraphSCI identifies cell types in simulated data

In order to assess our method, we followed the same way as the previous study (Eraslan et al., 2019) to construct two simulated data sets by Splatter (Zappia et al., 2017) package: (1) 2000 cells belonging to two types clustered by expression data of 3000 genes (namely SIM-T2) and (2) 3000 cells belonging to six types of cells clustered by expression data of 5000 genes (namely SIM-T6). On the SIM-T2 data with a simpler case, GraphSCI achieved imputed expression with a mean absolute error of 0.226, which is 21.2% lower than 0.274 by DCA. We further projected the imputed gene expression by t-SNE and clustered

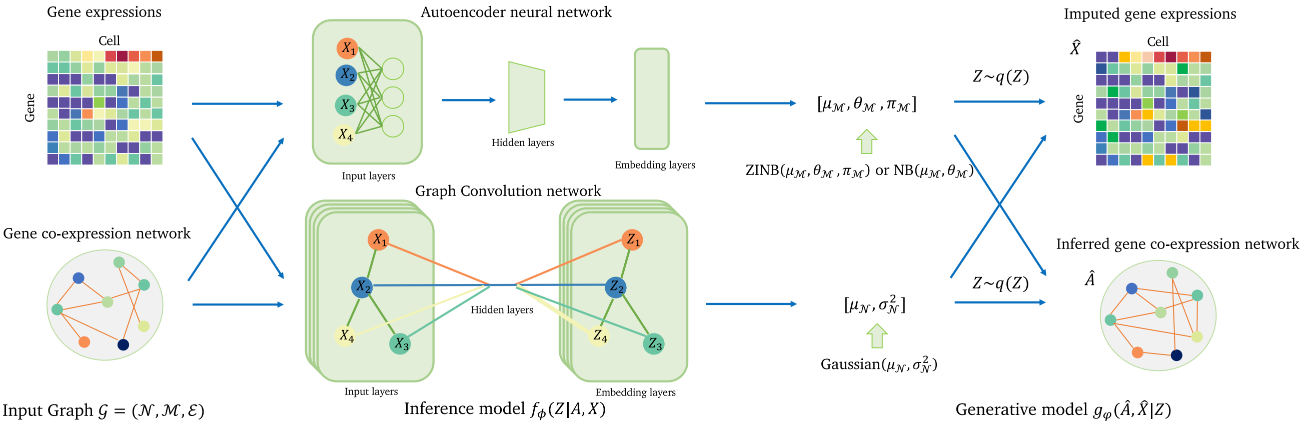

Gene expressions
Cell
Gene
Gene co-expression network
Input Graph $\mathcal{G} = (\mathcal{N}, \mathcal{M}, \mathcal{E})$
Autoencoder neural network
X_1
X_2
X_3
X_4
Input layers
Hidden layers
Embedding layers
Graph Convolution network
X_1
X_2
X_4
X_3
Hidden layers
Z_1
Z_2
Z_4
Z_3
Input layers
Embedding layers
Inference model $f_\phi(Z|A,X)$
$[\mu_\mathcal{M}, \theta_\mathcal{M}, \pi_\mathcal{M}]$
ZINB$(\mu_\mathcal{M}, \theta_\mathcal{M}, \pi_\mathcal{M})$ or NB$(\mu_\mathcal{M}, \theta_\mathcal{M})$
$[\mu_\mathcal{N}, \sigma_\mathcal{N}^2]$
Gaussian$(\mu_\mathcal{N}, \sigma_\mathcal{N}^2)$
$Z \sim q(Z)$
$Z \sim q(Z)$
Imputed gene expressions
$\hat{X}$
Cell
Gene
Inferred gene co-expression network
$\hat{A}$
Generative model $g_\varphi(\hat{A}, \hat{X}|Z)$

Figure 2. The architecture of GraphSCI model

The input of GraphSCI is the gene expression matrix and the gene-to-gene relationships. The Inference model $\boldsymbol{f_\varphi}$ is to learn the low-dimensional representations of genes and cells based on a combination of graph convolution network and Autoencoder neural network. The Generative model $\boldsymbol{g_\phi}$ utilizes the posterior distributions to reconstruct gene expression and gene-to-gene relationships respectively.

the cells by K-Means algorithm. As shown in Figure 3A, GraphSCI achieved 0.977, 0.994, 0.609 for ARI, CA, and SC values with standard deviation of 0.0038, 0.0043, 0.0024, respectively. These results are better than 0.920, 0.922 and 0.581 achieved by DeepImpute. Results with DCA, SAVER-X, scImpute are detailed in Table S1. By comparison, the clustering over original expression data without any imputation achieved 0.716, 0.508, 0.342 for ARI, CA and SC. Figure 3B shows 2000 cells by using the first two principle components obtained from t-SNE (Maaten and Hinton, 2008). Obviously, GraphSCI clearly separates two types of cells, while both DeepImpute and DCA have a small number of cells mixed together. The original data can't separate the cells at all.

When tested on the SIM-T6 data set with six cell groups, similar results were obtained. As shown in Figure S1A, our method achieved 0.818, 0.859, and 0.34 for ARI, CA, and SC values, respectively. These are 5.1, 3.2, and 16.4% higher than those by DeepImpute, and 6.6%, 2.0%, and 19.7% higher than DCA. Table S1 details results by the raw data, SAVER-X and scImpute. The visualization consistently indicated that our method separates the six types of cells better than other methods (Figure S1B). Figure S2 shows the image of gene expression matrix (X) before and after imputation ($\hat{X}$) in our simulated experiments. This comparison again demonstrates that GraphSCI could recover the original cell types effectively both in the Sim-T2 and the Sim-T6 data sets.

GraphSCI recovers transcriptome dynamics in real single-cell data

Another key criterion to evaluate the imputation methods is their ability to recover transcriptome dynamics in real single-cell data set. Therefore, we applied our method to three real scRNA-seq data sets and made comparisons with other methods. The first data set was obtained from mouse ES cells(Klein et al., 2015), which were measured to analyze the heterogeneity of mouse embryonic stem cells in different stages after leukemia inhibitory factor (LIF) withdrawal. We selected four different LIF withdrawal intervals (0, 2, 4, and 7 days) and put all cells together as the input of imputation. The imputed data were clustered by t-SNE. As shown by Figure 4A, GraphSCI separated the four stages of mouse ES cells clearly except that a few blue samples were mixed with the yellow. In comparison, the clustering obtained from the scImpute and DCA methods seriously mixed the blue samples with the yellow ones. As indicated by Figure 4B, ARI, CA, and SC of GraphSCI were significantly higher than DeepImpute and DCA. Results of ARI, CA, and SC with all methods are detailed in Table S2.

GraphSCI was further applied to two large data sets generated by the 10X scRNA-seq platform (Zheng et al., 2017), one of which is involved by the transcriptome of peripheral blood mononuclear cells (PBMCs) from a healthy donor. The data set contains 5247 PBMCs of 11 cell types. Because the same type of cells has similar expression profiles, we randomly selected 80% of PBMCs to train the model and used the remained for the independent test set. The imputed data on the independent set was conducted with dimension reduction results by t-SNE for visualizations. Figure 5A shows that the imputations by GraphSCI could

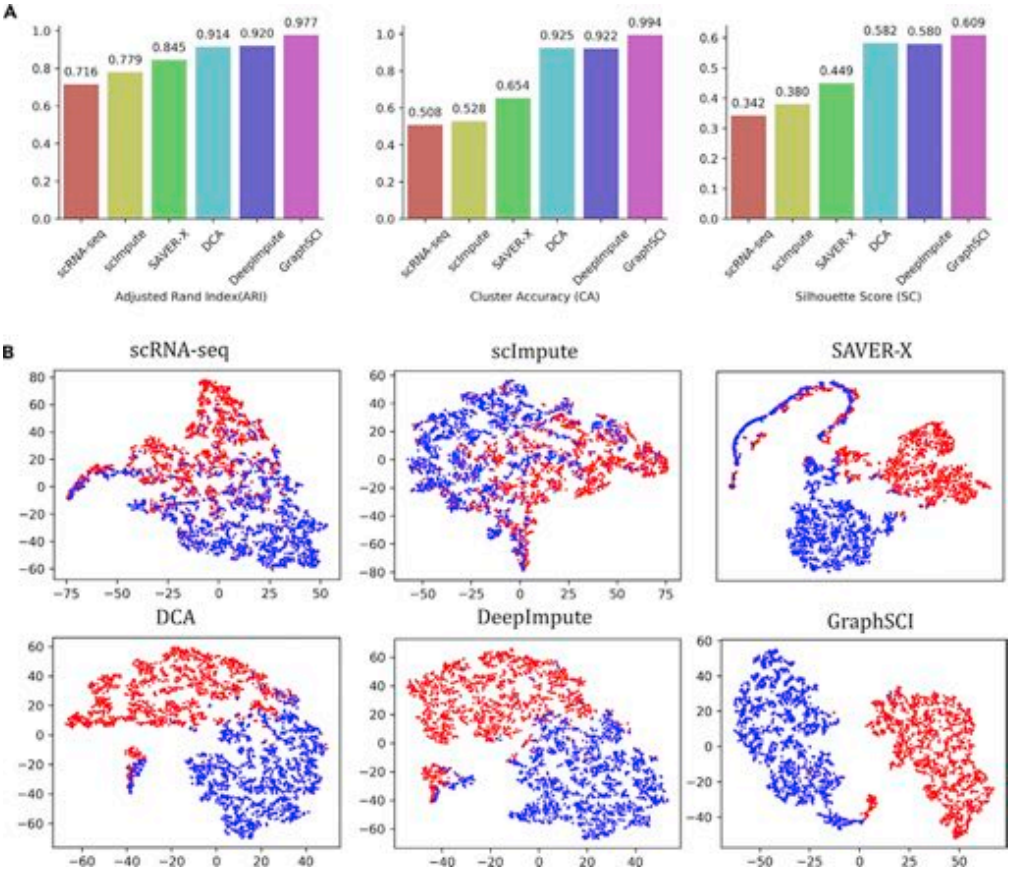


Figure 3. GraphSCI identifies cell types in simulated data with two cell groups (SIM-T2)

(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, GraphSCI. Each cell is colored by cell groups.

separate brown and dark samples well in the low-dimension representation. The orange samples had a diving line with other samples. In comparison, the results obtained by DeepImpute and DCA didn't show obvious differences among the black, red, and green samples. The results of ARI, CA, and SC on the independent test set also showed that GraphSCI outperformed other methods (Figure 5B). In details, our method achieved 0.472, 0.552, and 0.177 for ARI, CA, and SC values, respectively. These are 1.7%, 0.7%, 4.7% and 14.0, 9.7, 34.1% greater than those by DeepImpute and DCA respectively. More detailed results are shown in Table S2.

In the E18 Mouse data set, ~12,000 brain cells of 16 cell types were profiled from the 10X scRNA-seq platform. We applied GraphSCI to the large-scale scRNA-seq data set to demonstrate its robustness and scalability. As shown in Figure S3A, GraphSCI is able to separate cells of 16 cell types effectively in the low-dimension representation, while DeepImpute and DCA have mixed many subcellular types together. GraphSCI again achieved ARI, CA, and SC of 0.316, 0.422, and 0.030, respectively, consistently the greatest among all methods (Figure S3B and Table S2).

GraphSCI recovers gene expression levels in bulk RNA-seq data set

The efficacies of GraphSCI in recovering gene expression levels were further evaluated by a real RNA sequencing data set. The RNA sequencing data was obtained from C. elegans development experiments by Francesconi et al. (Francesconi and Lehner, 2014), which was used to simulate single-cell RNA-seq data with dropout rates ranging from 50% to 70%. The three data sets were generated by adding the single-cell specific noises through gene-wise subtracting values drawn from the exponential distribution. Since bulk

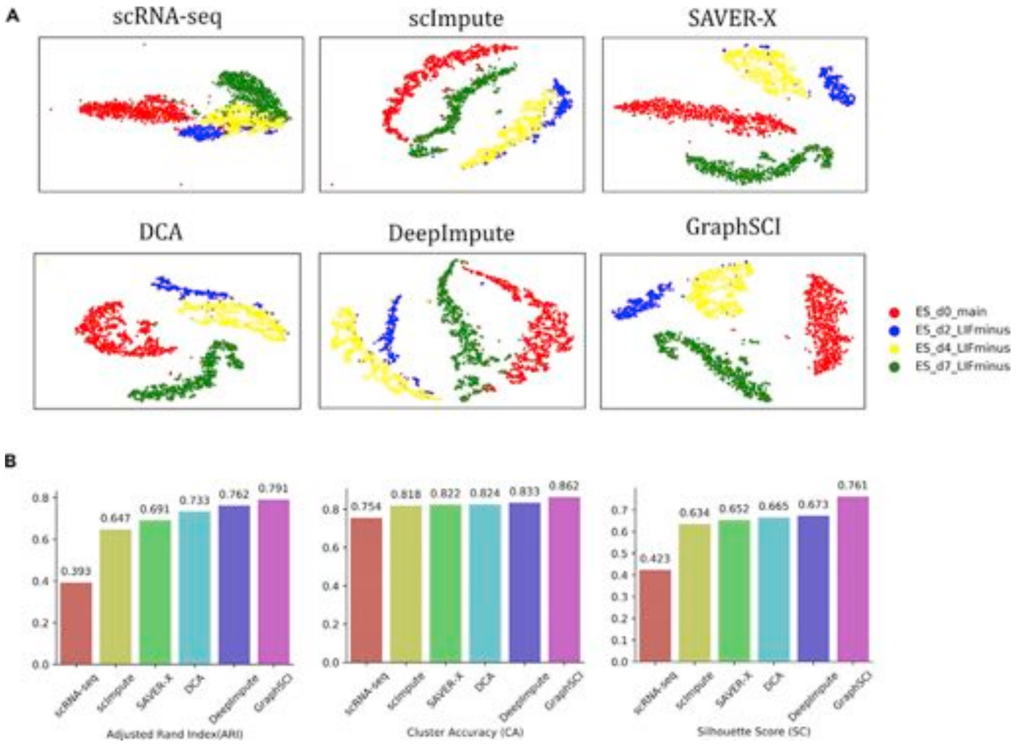


Figure 4. The performances on Mouse embryonic stem cells data set

(A) shows the t-SNE visualization reproduced from scRNA-seq, scImpute, SAVER, DCA, DeepImpute, and GraphSCI from top to bottom, from left to right.

(B) The comparison of clustering performances of scRNA-seq, scImpute, SAVER, DCA, DeepImpute, and GraphSCI, measured by ARI, CA, and SC.

RNA-seq data contains less noise than scRNA-seq, PCC was used to evaluate the effectiveness of imputation on real RNA-seq data set. As shown by Figure 6, GraphSCI outperformed DCA in recovering the gene expression levels in real RNA-seq data set. In details, the median of PCCs reached by GraphSCI in three data sets are 0.858, 0.807, and 0.788 respectively, consistently greater than 0.821, 0.765, and 0.742 achieved by DCA, and 0.787, 0.718, and 0.604 achieved by SAVER.

GraphSCI improves differential expression analysis on scRNA-seq data

An effective imputation method should lead to an improvement in differential expression analysis because scRNA-seq data provide insight into gene expression in a single cell. To evaluate whether the identification of DEGs are more accurate after imputation, we utilized a scRNA-seq data set with corresponding bulk RNA-seq data to compared differential expression analysis results using DESeq2. This data set, generated by Chu et al. from H1 human embryonic stem cells (H1) differentiated into definitive endoderm cells (DEC), has six samples of bulk RNA-seq and 350 samples of scRNA-seq (212 for H1 ESC and 138 for DEC). We applied GraphSCI and DeepImpute to impute the gene expression on scRNA-seq data and performed DE analysis on the raw data and the imputed data respectively. The percentages of zero gene expression are 49.1% in raw scRNA-seq data that results in the lowest DEGs identification results. In contrast, GraphSCI and DeepImpute have improved the identification of DEGs and share more DEGs with bulk samples. In Figure 7A, we defined more quantitative evaluation metrics such as the area under the receiver operating characteristic curve (AUC), the accuracy (ACC), and F-scores for DEGs detection. In detail, our method achieved 0.913, 0.782, and 0.608 for AUC, ACC, and F-score values, respectively. Moreover, Figure 7B and Figure 7C show that the expression profiles of DEC and ESC marker genes (SOX2 and LEFTY1) after GraphSCI imputation could better reflect the gene expression signatures on recovering the expression patterns of signature genes. The performance of other signature genes including NANOG, DNMT3B, GATA6, and CXCR4 et al. has been shown in Figure S4.

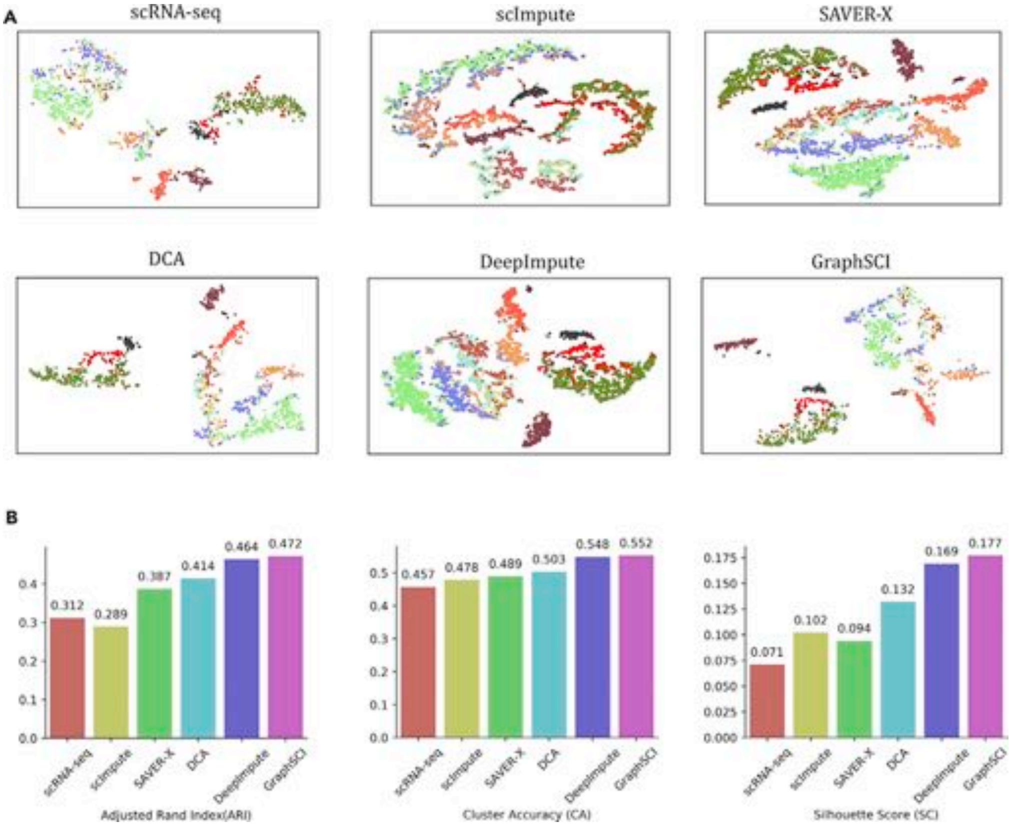


Figure 5. The performances on 5k peripheral blood mononuclear cells (PBMC) data set

(A) shows the t-SNE visualization reproduced from scRNA-seq, scImpute, SAVER, DCA, DeepImpute, and GraphSCI from top to bottom, from left to right.

(B) The comparison of clustering performances of scRNA-seq, scImpute, SAVER, DCA, DeepImpute, and GraphSCI, measured by ARI, CA, and SC.

GraphSCI infers gene-to-gene relationships from scRNA-seq data

GraphSCI can not only impute gene expression data of scRNA-seq effectively, but also infer gene-to-gene relationships from the data. Due to the dropout events in raw scRNA-seq data (Zheng et al., 2017; Iacono et al., 2019), it is challenging to obtain accurate gene interactions directly from correlation coefficients between gene expression (Aibar et al., 2017). Here, we applied our method to raw scRNA-seq data sets and reconstructed gene relations during imputation. By compared with the known interactions from the STRINGdb (Szklarczyk et al., 2017), the gene interactions constructed by GraphSCI had a precision of 0.713 with the threshold of 0.5. Specifically, the true positive (TP) was 232647 and the false positive (FP) was 93,812. Figure 8 shows the imputed gene-to-gene relationships obtained by Cytoscape (Smoot et al., 2010). The true positive (TP) is an outcome where the model correctly predicts the gene-gene relationships and the false positive is an outcome where the model incorrectly predicts the gene-gene relationships. They showed the accuracy of our model to infer the gene-gene relationships from the raw scRNA-seq data. Similar results were also observed in previous experiments on the mouse ES cells data set. The constructed gene relations had a precision of 0.682 with 199291 true positives and 92,924 false positives. As a comparison, we utilized the PCC to infer gene-gene relationships from the raw scRNA-seq data, which obtains the precision of 0.598 and 0.492 respectively. It again verifies the effectiveness of our method, empirically showing that it facilitates the inference of the gene-to-gene relationships during the training process.

DISCUSSION

In this study, we presented an imputation method, GraphSCI, based on GCNs, which are particularly suitable for single-cell RNA-seq data. Our method focused on imputing gene expression levels by integrating the gene expression with gene-to-gene relationships. By using gene-to-gene relationships as prior

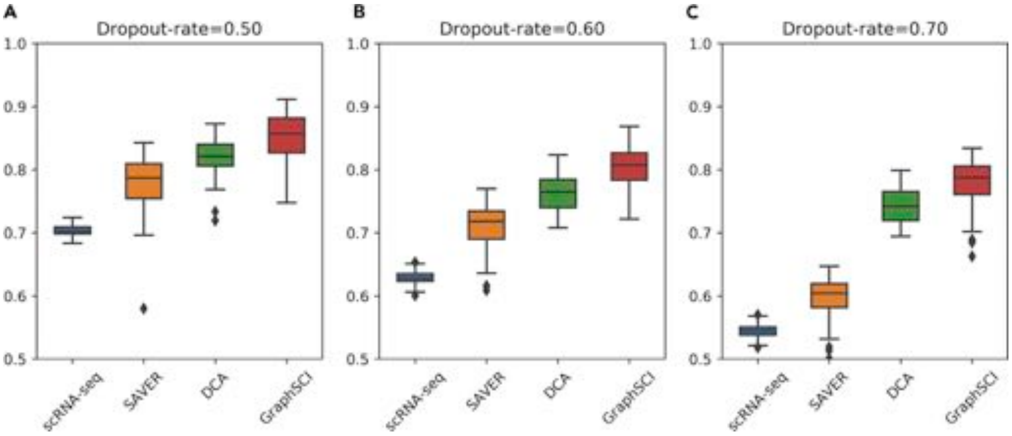


Figure 6. GraphSCI recovers gene expression levels in bulk RNA-seq data

Box diagram (A–C) depict the Pearson correlation coefficient between simulated data or imputed data and original data. And the box represents the interquartile range, the horizontal line in the box is the median, and the whiskers represent 1.5 times the interquartile range.

knowledge, this method avoided introducing excess biases during imputation and removed technical variations resulted from scRNA-seq.

To our best knowledge, this is the first study to integrate gene-to-gene relationships into deep learning framework for imputations on scRNA-seq. It is also the first attempt to employ GCN for learning the representation of gene-to-gene relationships in imputation study. Most importantly, extensive experiments were conducted on different kind of scRNA-seq data sets to demonstrate the superiority of our method.

GraphSCI was evaluated on both simulated and real data, which was identified with the best performances on diverse downstream analyses in comparison with other methods. In simulated data sets, GraphSCI was found to outperform other methods on the data with both small and large numbers of cells and cell types. The better performance of GraphSCI was further observed when it was applied in real data sets like bulk RNA-seq data and scRNA-seq data. In addition, another advantage of GraphSCI is its ability to infer the new gene-to-gene relationships, which is an absence of existing methods.

Applications of GCNs and exploiting gene-to-gene relationships for imputation, however, may also bring uncontrollable errors. For instance, the reliability of gene-to-gene relationships may influence the results of imputation. To solve this problem, we tried a variety of methods to build the gene-to-gene relationships, such as setting different thresholds to build edges or selecting original co-expressed samples to calculate PCC. We found that better performance could be achieved with the adjacency matrix obtained by selecting original co-expressed samples and PCC of >0.3 to determine edges. Figures S5 and S6 show the influence of the input gene-to-gene relationships on the overall results.

Another challenge for real data is that the evaluation of imputation may be difficult due to lack of ground truth. Therefore, we performed many clustering metrics, such as ARI, CA, and SC, to describe the effectiveness and robustness of competing methods, while we also utilized visualization to make the results clearer and more convincing. As shown in Figure S7, we could find that t-SNE showed better display results and GraphSCI consistently yields better performance with different clustering approaches.

The current GraphSCI was tested on data sets including simulated data and real data. The imputation power could be further improved with the increasing number of cells in the training set. Additionally, the deep learning networks by GraphSCI enable parallelization using GPUs to speed up training on large scRNA-seq data sets (Figure S8).

Resource availability

Lead contact

Yuedong Yang (yangyd25@mail.sysu.edu.cn) is the lead contact for this work.

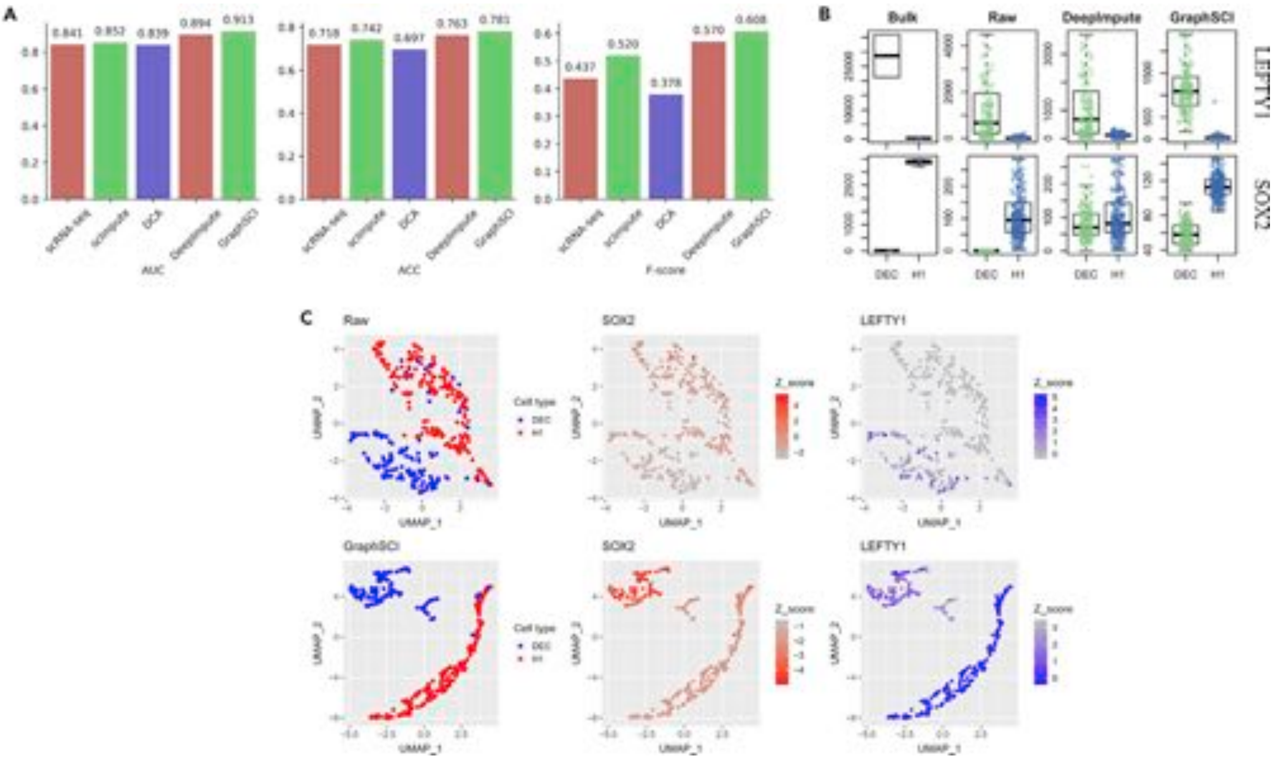


Figure 7. GraphSCI improves g differential expression analysis

(A) The bar plots show the performances of DEG detections from raw and imputed scRNA-seq data sets based on the gold standard defined by the bulk RNA-seq data set.

(B) The expression for selected signature genes (LEFTY1, SOX2) of H1 and DEC cells, respectively.

(C) The UMAP plots of the single cells overlaid by the expression of SOX2 and LEFTY1, which is the marker gene of H1 and DEC cells, respectively.

Materials availability

This study does not generate any new materials.

Data and code availability

The scRNA-seq data sets used in this manuscript are publicly available and their details are summarized in Table S3. The C. elegans time course experimental data was provided by the supplementary material of Francesconi. et al. The mouse embryonic stem cells data was downloaded from GSE65525. The 5k PBMC from a healthy donor and 10K brain cells from an E18 Mouse were provided by the 10X scRNA-seq platform and the website of the data are https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.1.0/5k_pbmc_protein_v3 and https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/neuron_10k_v3. The human embryos cells scRNA-seq data was downloaded from GSE44183. The Human ESC scRNA-seq data set for differential expression analysis was downloaded from GSE75748. The code generated during this study is available at https://github.com/biomed-AI/GraphSCI. We tuned model hyper-parameters based on the experimental results on simulated data sets and used them across all data sets (Figures S9 and S10).

METHODS

All methods can be found in the accompanying transparent methods supplemental file.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2021.102393.

ACKNOWLEDGMENTS

This study has been supported by the National Key R&D Program of China (2020YFB0204803), National Natural Science Foundation of China (61772566), Guangdong Key Field R&D Plan (2019B020228001 and

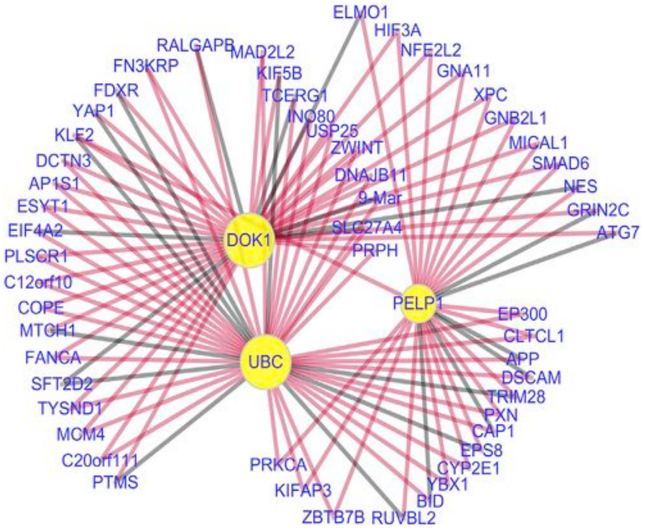


Figure 8. The gene-to-gene relationships after reconstruction

We selected the three genes with the highest degree and their common interactive genes. Compared to STRINGdb, the edges colored by red represent the correct gene-to-gene relationships we inferred, and the black edges represent the false inferred relationships.

2018B010109006), Introducing Innovative and Entrepreneurial Teams (2016ZT06D211), Guangzhou S&T Research Plan (202007030010).

AUTHOR CONTRIBUTIONS

J.Rao, X.Zhou, and Y.Yang contributed concept and implementation. J.Rao and Y.Yang co-designed experiments. J.Rao was responsible for programming. All of the authors contributed to the interpretation of results. J.Rao and Y.Yang wrote the manuscript. All of the authors reviewed and approved the final manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: December 24, 2020
Revised: March 7, 2021
Accepted: March 30, 2021
Published: May 21, 2021

REFERENCES

Aibar, S., González-Blas, C.B., Moerman, T., Imrichova, H., Hulselmans, G., Rambow, F., Marine, J.-C., Geurts, P., Aerts, J., and van den Oord, J. (2017). SCENIC: single-cell regulatory network inference and clustering. Nat. Methods *14*, 1083.

Arisdakessian, C., Poirion, O., Yunits, B., Zhu, X., and Garmire, L.X. (2019). DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. Genome Biol. *20*, 1–14.

Bhardwaj, N., and Lu, H. (2005). Correlation between gene expression profiles and protein–protein interactions within and across genomes. Bioinformatics *21*, 2730–2738.

Chakrabarti, D., Funiak, S., Chang, J., and Macskassy, S.A. (2014). Joint inference of multiple label types in large networks. ICML'14: Proceedings of the 31st International Conference on International Conference on Machine Learning *32*, 874–882.

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

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

Fraser, H.B., Hirsh, A.E., Wall, D.P., and Eisen, M.B. (2004). Coevolution of gene expression among interacting proteins. Proc. Natl. Acad. Sci. *101*, 9033–9038.

Gong, N.Z., Talwalkar, A., Mackey, L., Huang, L., Shin, E.C.R., Stefanov, E., Shi, E., and Song, D. (2014). Joint link prediction and attribute inference using a social-attribute network. ACM Trans. Intell. Syst. Technol. (Tist) *5*, 1–20.

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

Iacono, G., Massoni-Badosa, R., and Heyn, H. (2019). Single-cell transcriptomics unveils gene regulatory network plasticity. Genome Biol. *20*, 110.

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

Kipf, T.N., and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. 5th International Conference on Learning Representations (ICLR-17).

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

Kolodziejczyk, A.A., Kim, J.K., Svensson, V., Marioni, J.C., and Teichmann, S.A. (2015). The technology and biology of single-cell RNA sequencing. Mol. Cel. *58*, 610–620.

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

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

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

Meng, Z., Liang, S., Bao, H. & Zhang, X. Co-embedding attributed networks. Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019. 393-401.

Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., and Vert, J.-P. (2018). A general and flexible method for signal extraction from single-cell RNA-seq data. Nat. Commun. *9*, 284.

Saliba, A.-E., Westermann, A.J., Gorski, S.A., and Vogel, J. (2014). Single-cell RNA-seq: advances and future challenges. Nucleic Acids Res. *42*, 8845–8860.

Smoot, M.E., Ono, K., Ruscheinski, J., Wang, P.-L., and Ideker, T. (2010). Cytoscape 2.8: new features for data integration and network visualization. Bioinformatics *27*, 431–432.

Stuart, J.M., Segal, E., Koller, D., and Kim, S.K.J.s. (2003). A gene-coexpression network for global discovery of conserved genetic modules *302*, 249–255.

Svensson, V., Natarajan, K.N., Ly, L.-H., Miragaia, R.J., Labalette, C., Macaulay, I.C., Cvejic, A., and Teichmann, S.A. (2017). Power analysis of single-cell RNA-sequencing experiments. Nat. Methods *14*, 381.

Szklarczyk, D., Morris, J.H., Cook, H., Kuhn, M., Wyder, S., Simonovic, M., Santos, A., Doncheva, N.T., Roth, A., and Bork, P. (2017). The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible. Nucleic Acids Res. *45*, D362–D368.

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

Wang, J., Agarwal, D., Huang, M., Hu, G., Zhou, Z., Ye, C., and Zhang, N.R. (2019). Data denoising with transfer learning in single-cell transcriptomics. Nat. Methods *16*, 875–878.

Wang, Z., Gerstein, M., and Snyder, M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. Nat. Rev. Genet. *10*, 57.

Weirauch, M.T. (2011). Gene Coexpression Networks for the Analysis of DNA Microarray Data, *1*, pp. 215–250.

Yang, C., Zhong, L., Li, L.-J. & Jie, L. Bi-directional joint inference for user links and attributes on large social graphs. Proceedings of the 26th International Conference on World Wide Web Companion, 2017. 564-573.

Zappia, L., Phipson, B., and Oshlack, A. (2017). Splatter: simulation of single-cell RNA sequencing data. Genome Biol. *18*, 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., and Zhu, J. (2017). Massively parallel digital transcriptional profiling of single cells. Nat. Commun. *8*, 14049.

Zilionis, R., Nainys, J., Veres, A., Savova, V., Zemmour, D., Klein, A.M., and Mazutis, L. (2017). Single-cell barcoding and sequencing using droplet microfluidics. Nat. Protoc. *12*, 44.

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

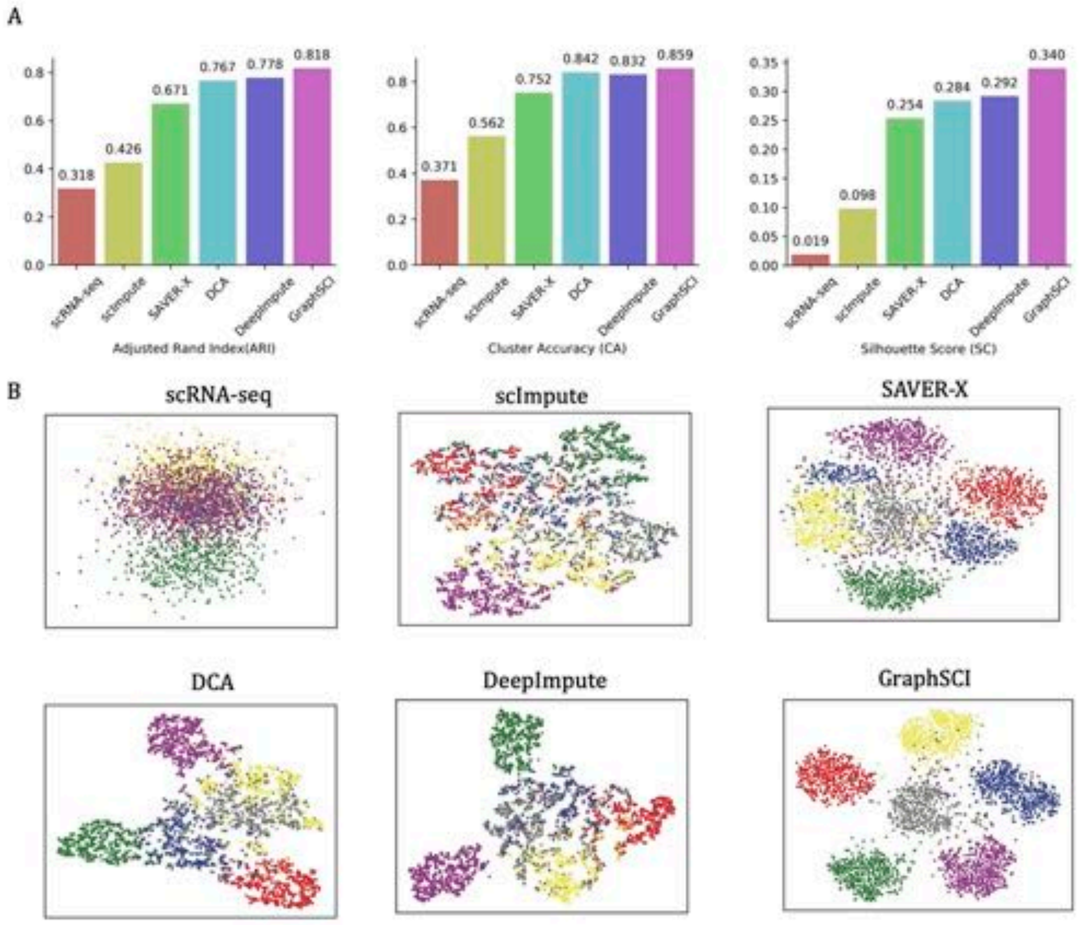


Figure S1. GraphSCI identifies cell types in simulated data with six cell groups (SIM-T6), Related to Figure 3. (A) The comparison of clustering performances of scRNA-seq, scImpute, SAVER, DCA, DeepImpute and GraphSCI, measured by ARI, CA and SC. (B) The two principle components by t-SNE from simulated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

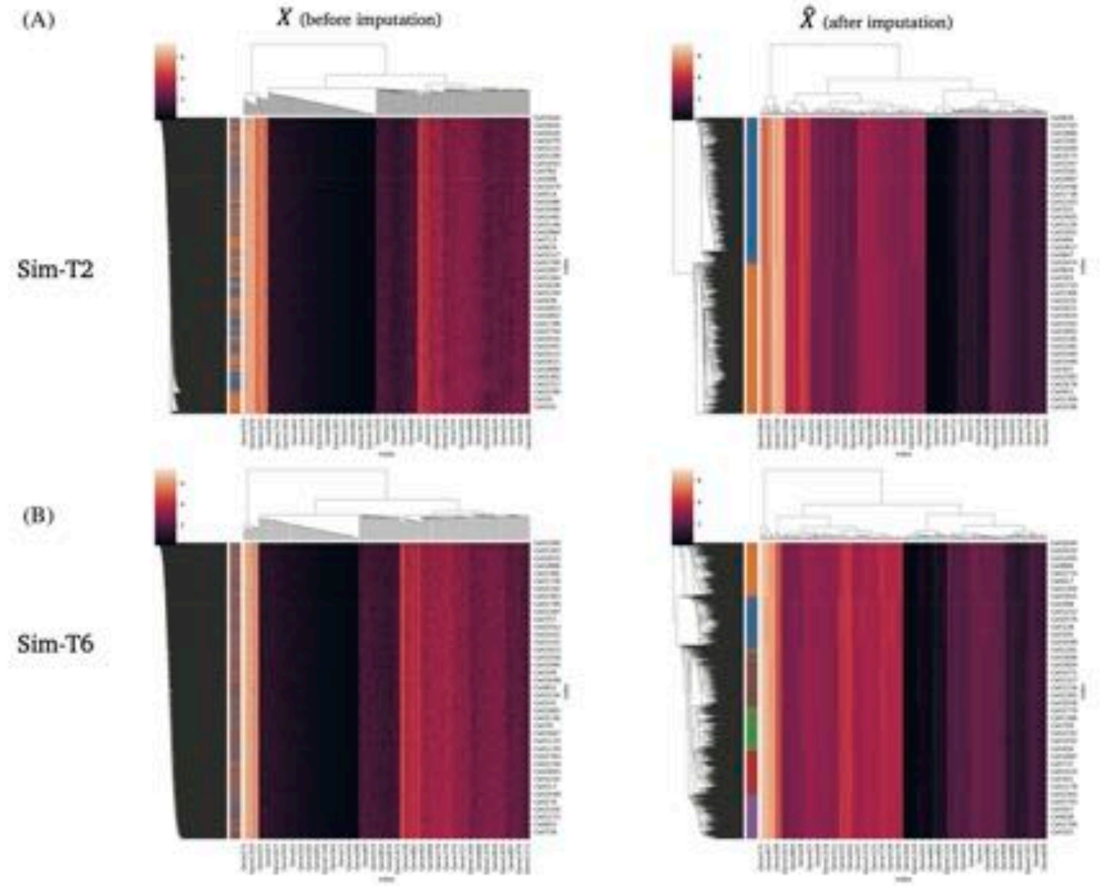


Figure S2. The image of gene expression matrix (X) before and after imputation ($\hat{X}$) in our simulated experiments, Related to Figure 3. The X axis represents cells and arranges the same cell types are nearby and the Y axis represents genes and similar genes nearby. (A) The comparison of gene expression 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.

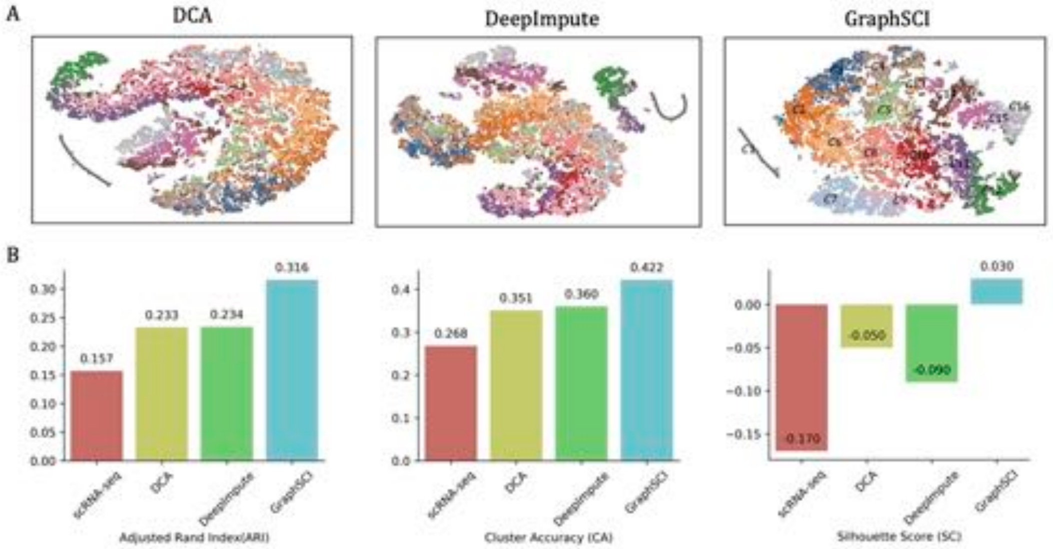


Figure S3. The performances on 10k Brain Cells from an E18 Mouse dataset, Related to Figure 5. (A) shows the t-SNE visualization reproduced from DCA, DeepImpute and GraphSCI from left to right. (B) The comparison of clustering performances of scRNA-seq, DCA, DeepImpute and GraphSCI, measured by ARI,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

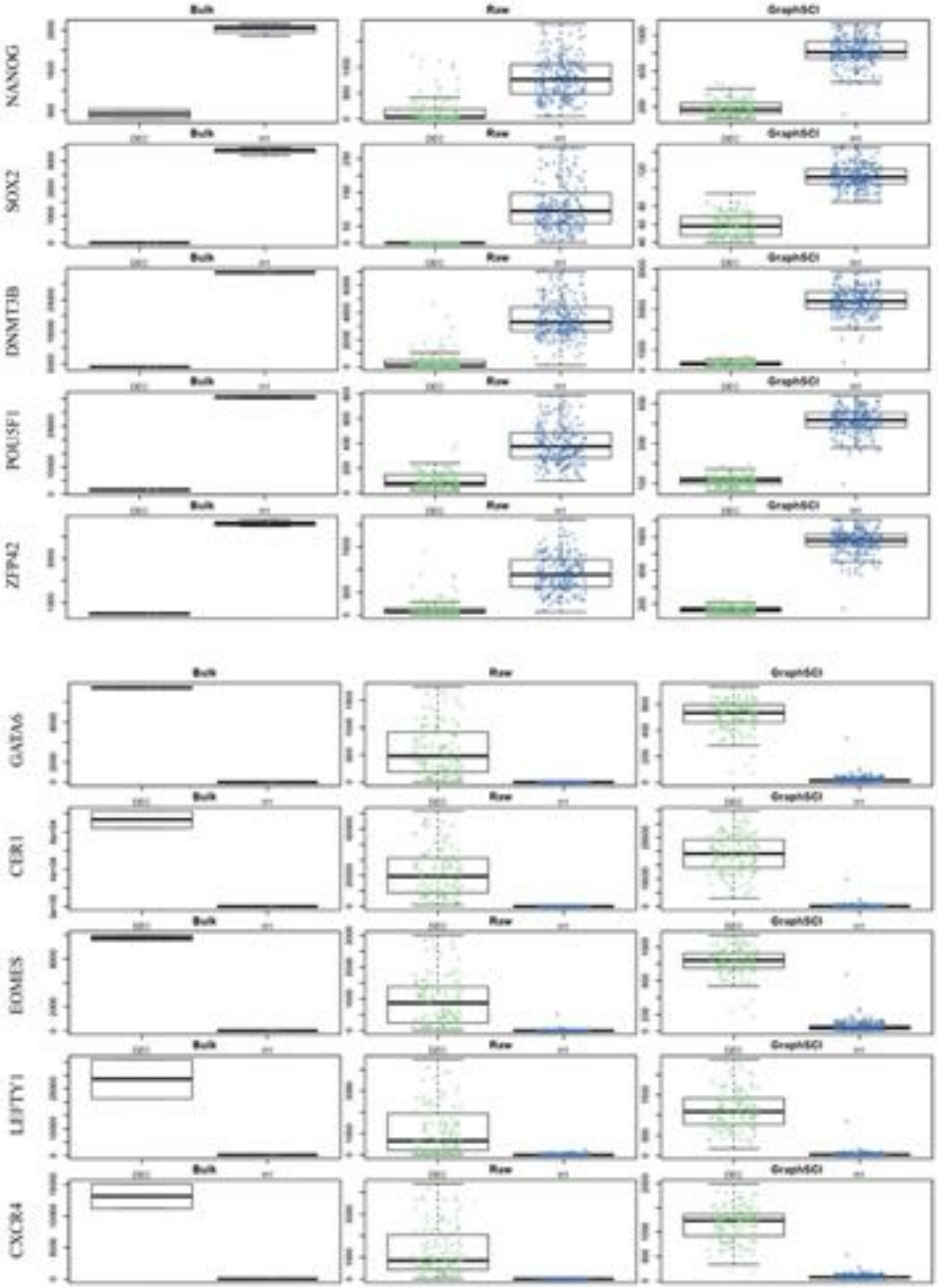


Figure S4. The performances of differential expression analysis, Related to Figure 7. The expression for signature genes (NANOG, SOX2, DNMT3B, POU5F1, ZFP42; GATA6, CER1, EOMES, LEFTY1, CXCR4) of H1 and DEC cells, respectively.

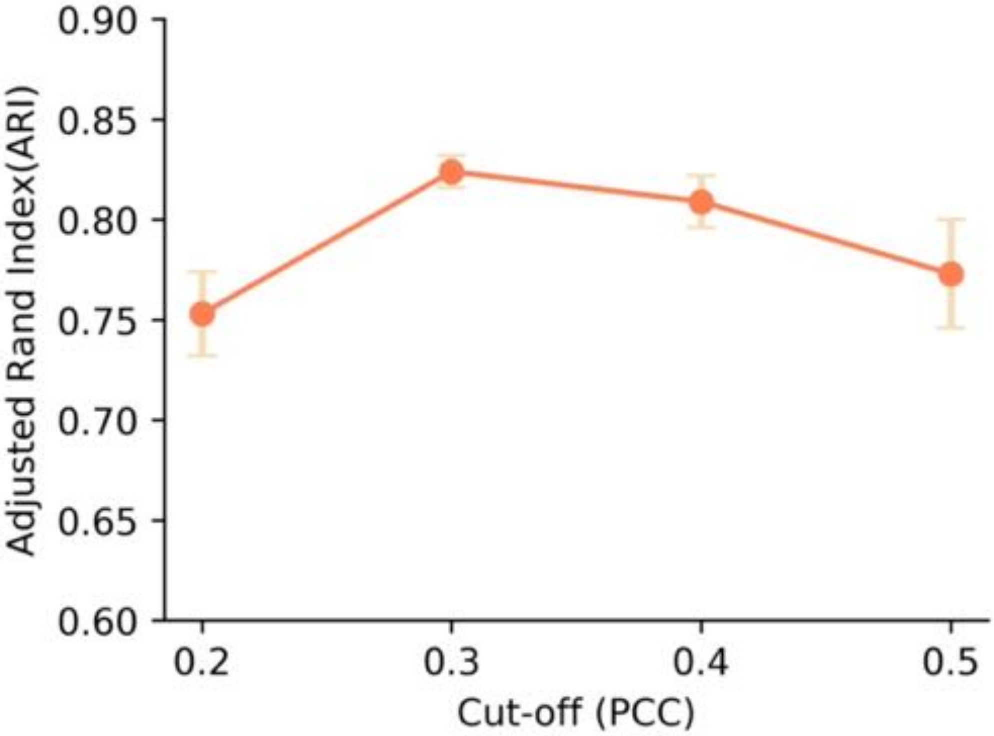


Figure S5. The analysis of different PCC cut-offs to construct the input gene-to-gene relationships, Related to Figure 1-2. We vary the cut-off of Pearson Correlation in {0.2, 0.3, 0.4, 0.5} to investigate their influences on the overall results. We could see that all relatively large cut-offs could 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.

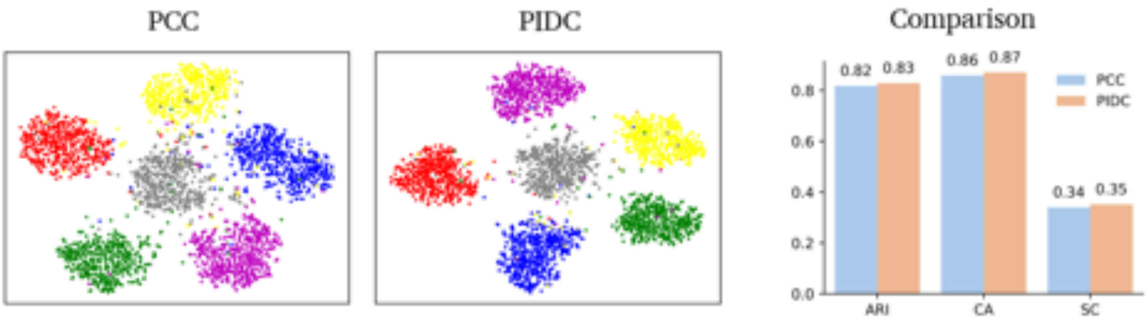


Figure S6. The comparison of different methods to construct gene-to-gene relationships (PCC and PIDC), Related to Figure 1-2. From the visualization and the clustering performance, we could find that the gene regulatory network inference tools such as PIDC could facilitates the imputation of scRNA-seq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

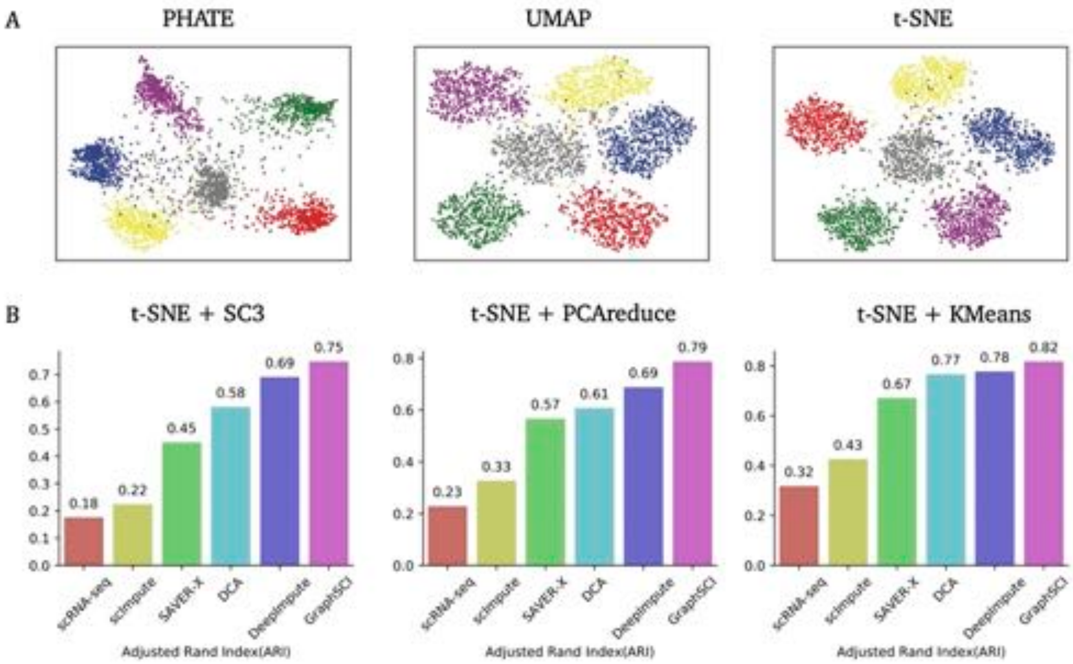


Figure S7. The comparison of different dimensional reduction algorithms and clustering approaches, Related to Figure 3. (A) We examined the influence of different cell visualization algorithms among UMAP, t-SNE, and PHATE from left to right. We could find that t-SNE showed better display results with 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.

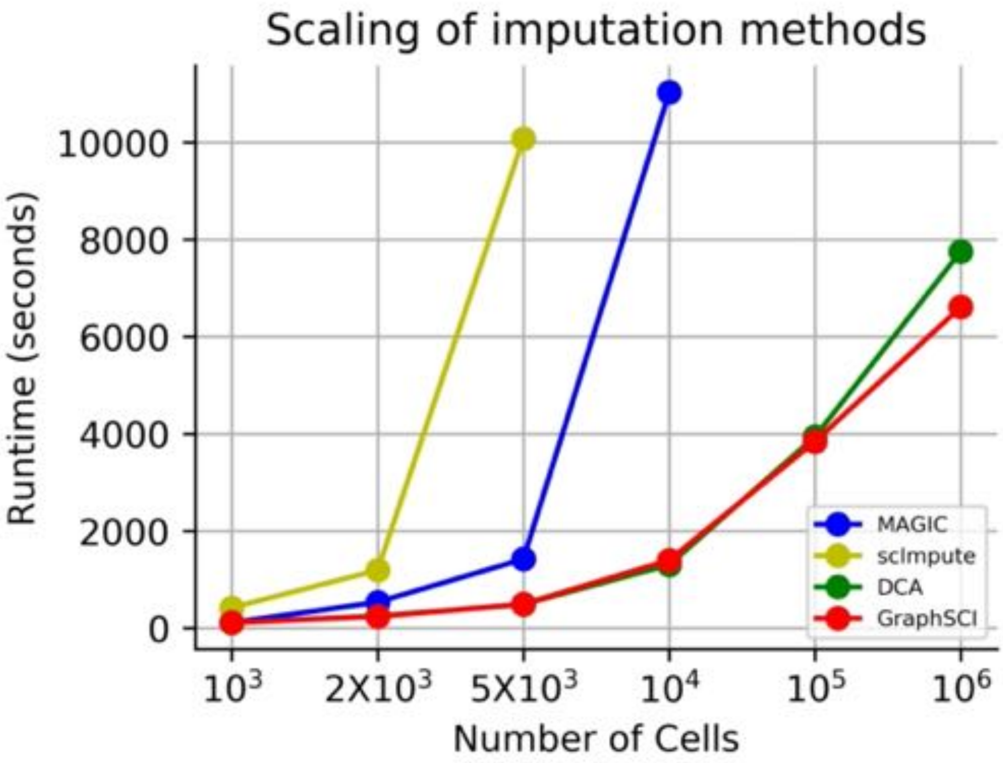


Figure S8. The runtimes for imputation with different numbers of cells down-sampled from 1.3 million mouse brain cells, Related to Figure 1-2. We analyzed the largest scRNA-seq data set in our experiments, which consists of 1.3 million mouse brain cells from 10X Genomics. The 1.3 million cell data matrix 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.

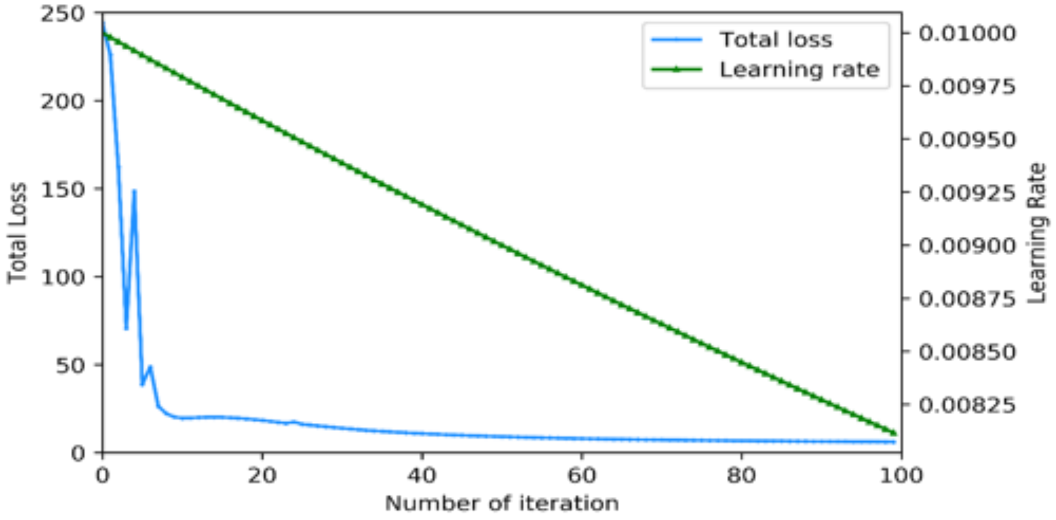


Figure S9. The optimization of our method, Related to Figure 1-2. We utilized the Adam optimizer with an initial learning rate of 0.01 and allowed it to decay exponentially with decay_rate = 0.9 and decay_steps = 50 during learning. The calculation of decayed learning rate in each step is: decayed_learning_rate = 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.

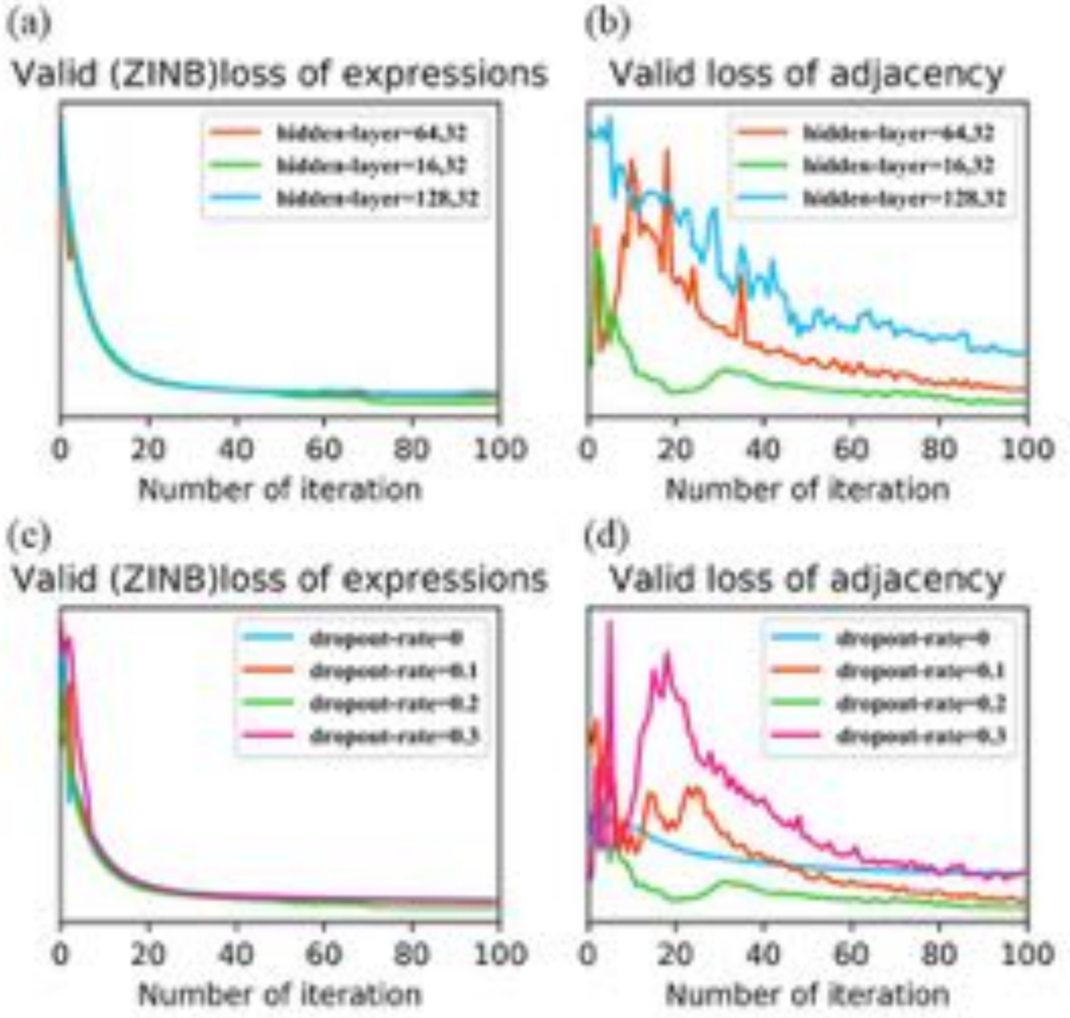


Figure S10. The exploration of Hyper-parameters, Related to Figure 1-2. During training, we randomly sampling 10% samples of each dataset as validation data and evaluate them in each iteration. The loss function of our method could be divided into two parts, one of which is the ZINB loss of gene expressions 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	**<u>0.791</u>**	**<u>0.862</u>**	**<u>0.761</u>**
	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	**<u>0.472</u>**	**<u>0.552</u>**	**<u>0.177</u>**
	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	**<u>0.316</u>**	**<u>0.422</u>**	**<u>0.030</u>**
	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:

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

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:

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

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):

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

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

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

where exp(·) is the exponential function and softplus(·) and sigmoid(·) 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 $([\mu_{\mathcal{N}}, \sigma_{\mathcal{N}}^2], [\mu_{\mathcal{M}}, \theta_{\mathcal{M}}, \pi_{\mathcal{M}}])$ 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 diag(·) is the diagonal matrix constructed by the vector (·) 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 $\mathcal{L}$:

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

$$- D_{KL}\left(q_\phi(Z^{\mathcal{M}}|A,X^T)||p(Z^{\mathcal{M}})\right) - D_{KL}(q_\phi(Z^{\mathcal{N}}|A,X)\middle|p(Z^{\mathcal{N}})\big). \qquad (19)$$

where $\mathbb{E}_{q_\phi}$ is the cross entropy function with the probabilistic distribution q_ϕ and $p_\varphi$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:

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

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

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

$$\mathcal{L}(\phi,\varphi) = \frac{1}{NML}\sum_{l=1}^{L}\left(\sum_{i\in\mathcal{N},j\in\mathcal{M}}\log p_{\varphi_1}\left(X_{ij}\middle|Z_i^{\mathcal{N}^{(l)}},Z_j^{\mathcal{M}^{(l)}}\right)\right)$$

$$+\frac{1}{N^2L}\sum_{l=1}^{L}\left(\sum_{i,j\in\mathcal{N}}\log p_{\varphi_2}\left(A_{ij}\middle|Z_i^{\mathcal{N}^{(l)}},Z_j^{\mathcal{N}^{(l)}}\right)\right)$$

$$-\frac{1}{2M}\sum_{j\in\mathcal{M}}\sum_{d=1}^{D}(\pi_{\mathcal{M}}\delta_0\left(Z^{(d)}\right) + (1-\pi_{\mathcal{M}})NB(Z^{(d)}|\mu_{\mathcal{M}};\theta_{\mathcal{M}}))$$

$$-\frac{3}{2N}\sum_{i\in\mathcal{N}}\sum_{d=1}^{D}\left(1+\log\sigma^2_{\mathcal{N}_{(i)}^{(d)}} - \sigma^2_{\mathcal{N}_{(i)}^{(d)}} - \mu^2_{\mathcal{N}_{(i)}^{(d)}}\right) \qquad (22)$$

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.