nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-021-22197-x OPEN

# scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses

Juexin Wang[1,4], Anjun Ma[2,4], Yuzhou Chang[2], Jianting Gong[1], Yuexu Jiang[1], Ren Qi[2], Cankun Wang[2], Hongjun Fu[3], Qin Ma[2✉] & Dong Xu[1✉]

Single-cell RNA-sequencing (scRNA-Seq) is widely used to reveal the heterogeneity and dynamics of tissues, organisms, and complex diseases, but its analyses still suffer from multiple grand challenges, including the sequencing sparsity and complex differential patterns in gene expression. We introduce the scGNN (single-cell graph neural network) to provide a hypothesis-free deep learning framework for scRNA-Seq analyses. This framework formulates and aggregates cell–cell relationships with graph neural networks and models heterogeneous gene expression patterns using a left-truncated mixture Gaussian model. scGNN integrates three iterative multi-modal autoencoders and outperforms existing tools for gene imputation and cell clustering on four benchmark scRNA-Seq datasets. In an Alzheimer's disease study with 13,214 single nuclei from postmortem brain tissues, scGNN successfully illustrated disease-related neural development and the differential mechanism. scGNN provides an effective representation of gene expression and cell–cell relationships. It is also a powerful framework that can be applied to general scRNA-Seq analyses.

[1] Department of Electrical Engineering and Computer Science, and Christopher S. Bond Life Sciences Center, University of Missouri, Columbia, MO, USA. [2] Department of Biomedical Informatics, College of Medicine, The Ohio State University, Columbus, OH, USA. [3] Department of Neuroscience, The Ohio State University, Columbus, OH, USA. [4] These authors contributed equally: Juexin Wang, Anjun Ma. ✉email: qin.ma@osumc.edu; xudong@missouri.edu

Single-cell RNA-sequencing (scRNA-seq) techniques enable transcriptome-wide gene expression measurement in individual cells, which are essential for identifying cell-type clusters, inferring the arrangement of cell populations according to trajectory topologies, and highlighting somatic clonal structures while characterizing cellular heterogeneity in complex diseases[1,2]. scRNA-seq analysis for biological inference remains challenging due to its complex and un-determined data distribution, which has a large volume and high rate of dropout events. Some pioneer methodologies, e.g., Phenograph[3], MAGIC[4], and Seurat[5] use a k-nearest-neighbor (KNN) graph to model the relationships between cells. However, such a graph representation may over-simplify the complex cell and gene relationships of the global cell population. Recently, the emerging graph neural network (GNN) has deconvoluted node relationships in a graph through neighbor information propagation in a deep learning architecture[6–8]. Compared with other autoencoders used in the scRNA-Seq analysis[9–12] for revealing an effective representation of scRNA-Seq data via recreating its own input, the unique feature of graph autoencoder is in being able to learn a low-dimensional representation of the graph topology and train node relationships in a global view of the whole graph[13].

We introduce a multi-modal framework scGNN (single-cell graph neural network) for modeling heterogeneous cell–cell relationships and their underlying complex gene expression patterns from scRNA-Seq. scGNN trains low-dimensional feature vectors (i.e., embedding) to represent relationships among cells through topological abstraction based on both gene expression and transcriptional regulation information. There are three unique features in scGNN: (i) scGNN utilizes GNN with multi-modal autoencoders to formulate and aggregate cell–cell relationships, providing a hypothesis-free framework to derive biologically meaningful relationships. The framework does not need to assume any statistical distribution or relationships for gene expression data or dropout events. (ii) Cell-type-specific regulatory signals are modeled in building a cell graph, equipped with a left-truncated mixture Gaussian (LTMG) model for scRNA-Seq data[14]. This can improve the signal-to-noise ratio in terms of embedding biologically meaningful information. (iii) Bottom-up cell relationships are formulated from a dynamically pruned GNN cell graph. The entire graph can be represented by pooling on learned graph embedding of all nodes in the graph. The graph embedding can be used as low-dimensional features with tolerance to noises for the preservation of topological relationships in the cell graph. The derived cell–cell relationships are adopted as regularizers in the autoencoder training to recover gene expression values.

scGNN has great potential in capturing biological cell–cell relationships in terms of cell-type clustering, cell trajectory inference, cell lineages formation, and cells transitioning between states. In this paper, we mainly focus on discovering its applicative power in two fundamental aspects from scRNA-Seq data, i.e., gene imputation and cell clustering. Gene imputation aims to solve the dropout issue which commonly exists in scRNA-Seq data where the expressions of a large number of active genes are marked as zeros[15–17]. The excess of zero values often needs to be recovered or handled to avoid the exaggeration of the dropout events in many downstream biological analyses and interpretations. Existing imputation methods[18], such as MAGIC[4] and SAVER[19], have an issue in generating biased estimates of gene expression and tend to induce false-positive and biased gene correlations that could possibly eliminate some meaningful biological variations[20,21]. On the other hand, many studies, including Seurat[5] and Phenograph[3], have explored the cell–cell relationships using raw scRNA-seq data, and built cell graphs with reduced data dimensions and detected cell clusters by applying the Louvain modularity optimization. Accurate cell–cell relationships obey the rule that cells are more homogeneous within a cell type and more heterogeneous among different cell types[22], The scGNN model provides a global perspective in exploring cell relationships by integrating cell neighbors on the whole population.

scGNN achieves promising performance in gene imputation and cell cluster prediction on four scRNA-Seq data sets with gold-standard cell labels[23–26], compared to nine existing imputation and four clustering tools (Supplementary Table 1). We believe that the superior performance in gene imputation and cell cluster prediction benefits from (i) our integrative autoencoder framework, which synergistically determines cell clusters based on a bottom-up integration of detailed pairwise cell–cell relationships and the convergence of predicted clusters, and (ii) the integration of both gene regulatory signals and cell network representations in hidden layers as regularizers of our autoencoders. To further demonstrate the power of scGNN in complex disease studies, we applied it to an Alzheimer's disease (AD) data set containing 13,214 single nuclei, which elucidated its application power on cell-type identification and recovering gene expression values[27]. We claim that such a GNN-based framework is powerful and flexible enough to have great potential in integrating scMulti-Omics data.

## Results

**The architecture of scGNN comprises stacked autoencoders.** The main architecture of scGNN is used to seek effective representations of cells and genes that are useful for performing different tasks in scRNA-Seq data analyses (Fig. 1 and Supplementary Fig. 1). It has three comprehensive computational components in an iteration process, including gene regulation integration in a feature autoencoder, cell graph representation in a graph autoencoder, gene expression updating in a set of parallel cell-type-specific cluster autoencoders, as well as the final gene expression recovery in an imputation autoencoder (Fig. 1).

The feature autoencoder intakes the pre-processed gene expression matrix after the removal of low-quality cells and genes, normalization, and variable gene ranking (Fig. 2a). First, the LTMG model[14,28] is adopted to the top 2,000 variable genes to quantify gene regulatory signals encoded among diverse cell states in scRNA-Seq data (see "Methods" section and Supplementary Fig. 2). This model was built based on the kinetic relationships between the transcriptional regulatory inputs and mRNA metabolism and abundance, which can infer the expression of multi-modalities across single cells. The captured signals have a better signal-to-noise ratio to be used as a high-order restraint to regularize the feature autoencoder. The aim of this regularization is to treat each gene differently based on their individual regulation status through a penalty in the loss function. The feature autoencoder learns a low-dimensional embedding by the gene expression reconstruction together with the regularization. A cell–cell graph is generated from the learned embedding via the KNN graph, where nodes represent individual cells and the edges represent neighborhood relations among these cells[29,30]. Then, the cell graph is pruned from selecting an adaptive number of neighbors for each node on the KNN graph by removing the noisy edges[3].

Taking the pruned cell graph as input, the encoder of the graph autoencoder uses GNN to learn a low-dimensional embedding of each node and then regenerates the whole graph structure through the decoder of the graph autoencoder (Fig. 2b). Based on the topological properties of the cell graph, the graph autoencoder abstracts intrinsic high-order cell–cell relationships propagated on the global graph. The low-dimensional graph embedding

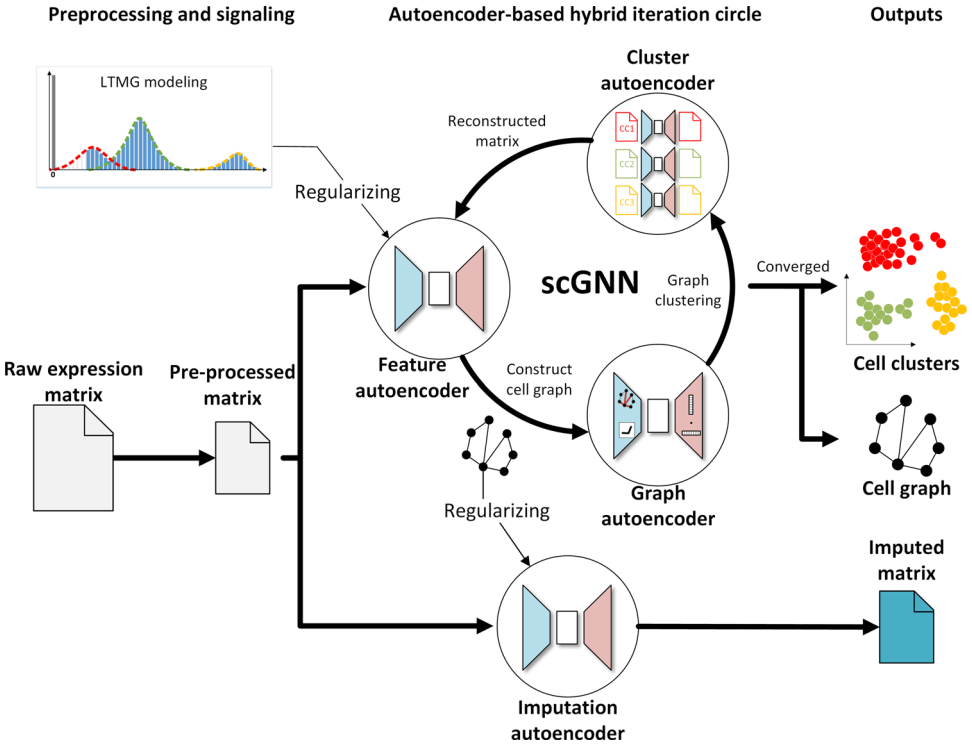


**Fig. 1 The architecture of scGNN.** It takes the gene expression matrix generated from scRNA-Seq as the input. LTMG can translate the input gene expression data into a discretized regulatory signal as the regularizer for the feature autoencoder. The feature autoencoder learns a dimensional representation of the input as embedding, upon which a cell graph is constructed and pruned. The graph autoencoder learns a topological graph embedding of the cell graph, which is used for cell-type clustering. The cells in each cell type have an individual cluster autoencoder to reconstruct gene expression values. The framework treats the reconstructed expression as a new input iteratively until converging. Finally, the imputed gene expression values are obtained by the feature autoencoder regularized by the cell–cell relationships in the learned cell graph on the original pre-processed raw expression matrix through the imputation autoencoder. LTMG is abbreviated for the left-truncated mixed Gaussian model.

integrates the essential pairwise cell–cell relationships and the global cell–cell graph topology using a graph formulation by regenerating the topological structure of the input cell graph. Then the k-means clustering method is used to cluster cells on the learned graph embedding[31], where the number of clusters is determined by the Louvain algorithm[31] on the cell graph.

The expression matrix in each cell cluster from the feature autoencoder is reconstructed through the cluster autoencoder. Using the inferred cell-type information from the graph autoencoder, the cluster autoencoder treats different cell types specifically and regenerates expression in the same cell cluster (Fig. 2c). The cluster autoencoder helps discover cell-type-specific information for each cell type in its individualized learning. Accompanied by the feature autoencoder, the cluster autoencoder leverages the inferences between global and cell-type-specific representation learning. Iteratively, the reconstructed matrix is fed back into the feature autoencoder. The iteration process stops until it converges with no change in cell clustering and this cell clustering result is recognized as the final results of cell-type prediction.

After the iteration stops, this imputation autoencoder takes the original gene expression matrix as input and is trained with the additional L1 regularizer of the inferred cell–cell relationships. The regularizers (see "Methods" section) are generated based on edges in the learned cell graph in the last iteration and their co-occurrences in the same predicted cell type. Besides, the L1 penalty term is applied to increase the model generalization by squeezing more zeroes into the autoencoder model weights. The sparsity brought by the L1 term benefits the expression imputation in dropout effects. Finally, the reconstructed gene expression values are used as the final imputation output.

**scGNN can effectively impute scRNA-Seq data and accurately predict cell clusters**. To assess the imputation and cell clustering performance of scGNN, four scRNA data sets (i.e., Chung[26], Kolodziejczy[23], Klein[24], and Zeisel[25]) with gold-standard cell-type labels are chosen as the benchmarks (more performance evaluation on other data sets can be found in Supplementary Data 1–2). We simulated the dropout effects by randomly flipping a number of the non-zero entries to zeros. The synthetic dropout simulation was based on the same leave-one-out strategy used in scVI[32] (Supplementary Fig. 3). Median L1 distance, cosine similarity, and root-mean-square-deviation (RMSE) scores between the original data set and the imputed values for these synthetic entries were calculated to compare scGNN with MAGIC[4], SAUCIE[10], SAVER[19], scImpute[33], scVI[32], DCA[11], DeepImpute[34], scIGANs[35], and netNMF-sc[36] (see "Methods" section). scGNN achieves the best results in recovering gene expressions in terms of median L1 distance, and RMSE at the 10 and 30% synthetic dropout rate, respectively. While the cosine similarity score of scGNN ranks at the top place for 10% rate and the third place for 30% rate. (Fig. 3a and Supplementary Data 1). Furthermore, scGNN can recover the underlying gene–gene relationships missed in the raw expression data due to the sparsity of scRNA-Seq. For example, two pluripotency epiblast gene pairs, *Ccnd3* versus *Pou5f1* and *Nanog* versus *Trim28*, are lowly correlated in the original raw data but show strong correlations relations, which are differentiated by time points after scGNN imputation and, therefore, perform with a consistency leading to the desired results sought in the original paper[24] (Fig. 3b). The recovered relations of four more gene pairs are also showcased in Supplementary Figure 4. scGNN amplifies differentially expressed

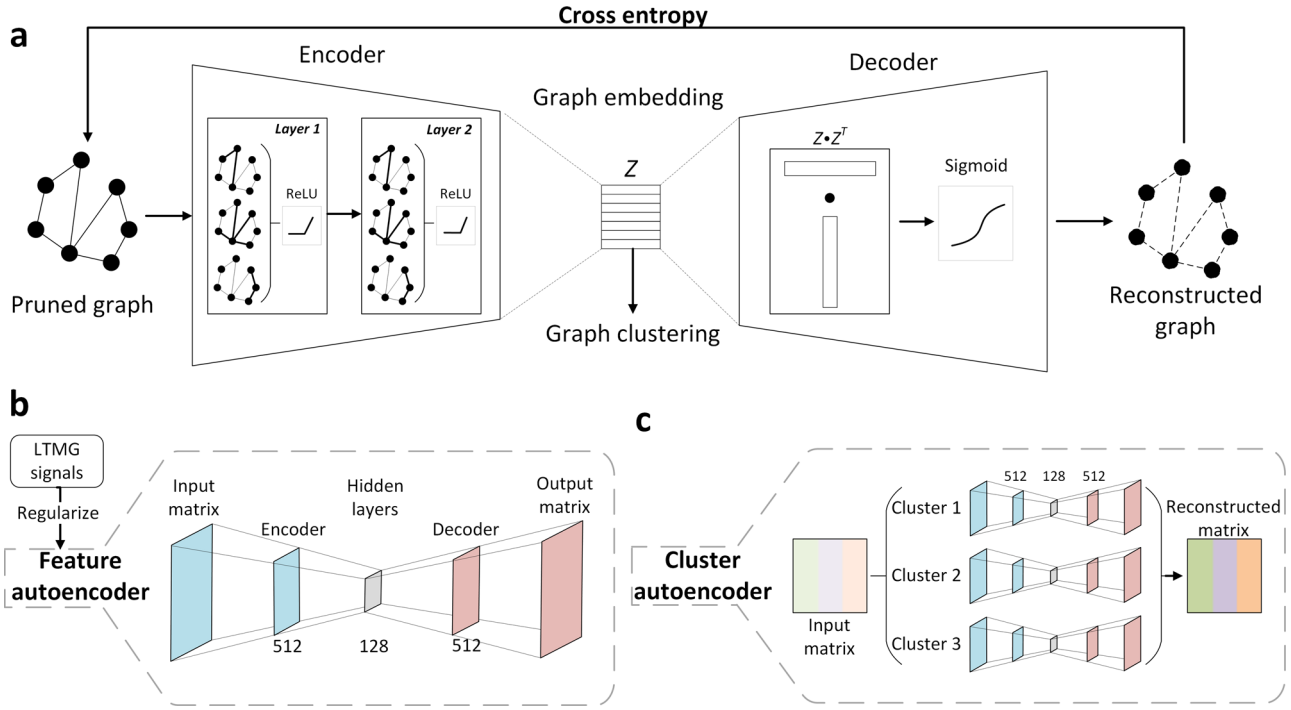


**Fig. 2 The architecture of scGNN Autoencoders. a** The feature autoencoder takes the expression matrix as the input, regularized by LTMG signals. The dimensions of the encoder and decoder layers are 512 × 128 and 128 × 512, respectively. The feature autoencoder is trained by minimizing the difference between the input matrix and the output matrix. **b** The graph autoencoder takes the adjacency matrix of the pruned graph as the input. The encoder consists of two layers of GNNs. In each layer, each node of the graph aggregates information from its neighbors. The encoder learns a low-dimensional presentation (i.e., graph embedding) of the pruned cell graph. The decoder reconstructs the adjacency matrix of the graph by dot products of the learned graph embedding followed by a sigmoid activation function. The graph autoencoder is trained by minimizing the cross-entropy loss between the input and the reconstructed graph. Cell clusters are obtained by applying k-means and Louvain on the graph embedding. **c** The cluster autoencoder takes a reconstructed expression matrix from the feature autoencoder as the input. An individual encoder is built on the cells in each of the identified clusters, and each autoencoder is trained individually. The concatenation of the results from all clusters is treated as the reconstructed matrix.

genes (DEGs) signals with a higher fold change than the original, using an imputed matrix to confidently depict the cluster heterogeneity (Fig. 3c). We also compared the DEG signal changes before and after imputation using other imputation tools. As an example, 744 DEGs (logFC > 0.25) identified in Microglia (benchmark cell label) of Zeisel data were compared logFC value change before and after imputation (Supplementary Fig. 5). The result turned out that scGNN is the only tool that increases all most all DEG signals in Microglia with the strongest Pearson's correlation coefficient to the original data. Other tools showed weaker coefficients and signals in some of the genes were decreased, indicating imputation bias in these tools. Our results indicate that scGNN can accurately restore expression values, capture true gene–gene relations, and increase DEG signals, without inducing additional noises.

Besides the artificial dropout benchmarks, we continued to evaluate the clustering performance of scGNN and the nine imputation tools on the same two data sets. The predicted cell labels were systematically evaluated using 10 criteria including an adjusted Rand index (ARI)[37], Silhouette[38], and eight other criteria (Fig. 4a and Supplementary Data 2). By visualizing cell clustering results on UMAPs[39], one can observe more apparent closeness of cells within the same cluster and separation among different clusters when using scGNN embeddings compared to the other nine imputation tools (Fig. 4b). We also observed that compared to the tSNE[40] and PHATE[41] visualization methods, UMAP showed better display results with closer inner-group distance and larger between-group distances (Supplementary Fig. 6). The expression patterns show heterogeneity along with embryonic stem cell development. In the case of Klein's time-series data, scGNN recovered a complex structure that was not well represented by the raw data, showing a well-aligned trajectory path of cell development from Day 1 to Day 7 (Fig. 4c). Moreover, scGNN showed significant enhancement in cell clustering compared to the existing scRNA-Seq analytical framework (e.g., Seurat using the Louvain method) when using the raw data (Supplementary Fig. 7). We hypothesized that the cell–cell graph constructed from scGNN can reflect cell–cell communications based on ligand–receptor pairs. Using CellChat[42] and curated receptor–ligand pairs, we proved that aggregated interaction probability of cell pairs defined in an scGNN cell–cell graph is significantly higher than randomly selected cell pairs, which strongly indicates the capability of scGNN in capturing the real cell–cell communications and interactions (Supplementary Fig. 8).

On top of that, to address the significance of using the graph autoencoder and cluster autoencoder in scGNN, we performed ablation tests to bypass each autoencoder and compare the ARI results on the Klein data set (Fig. 4d and Supplementary Fig. 9). The results showed that removing either of these two autoencoders dramatically decreased the performance of scGNN in terms of cell clustering accuracy. Another test using all genes rather than the top 2,000 variable genes also showed poor performance in the results and doubled the runtime of scGNN, indicating that those low variable genes may reduce the signal-to-noise ratio and negatively affect the accuracy of scGNN. The design and comprehensive results of the ablation studies on both clustering and imputation are detailed in Supplementary

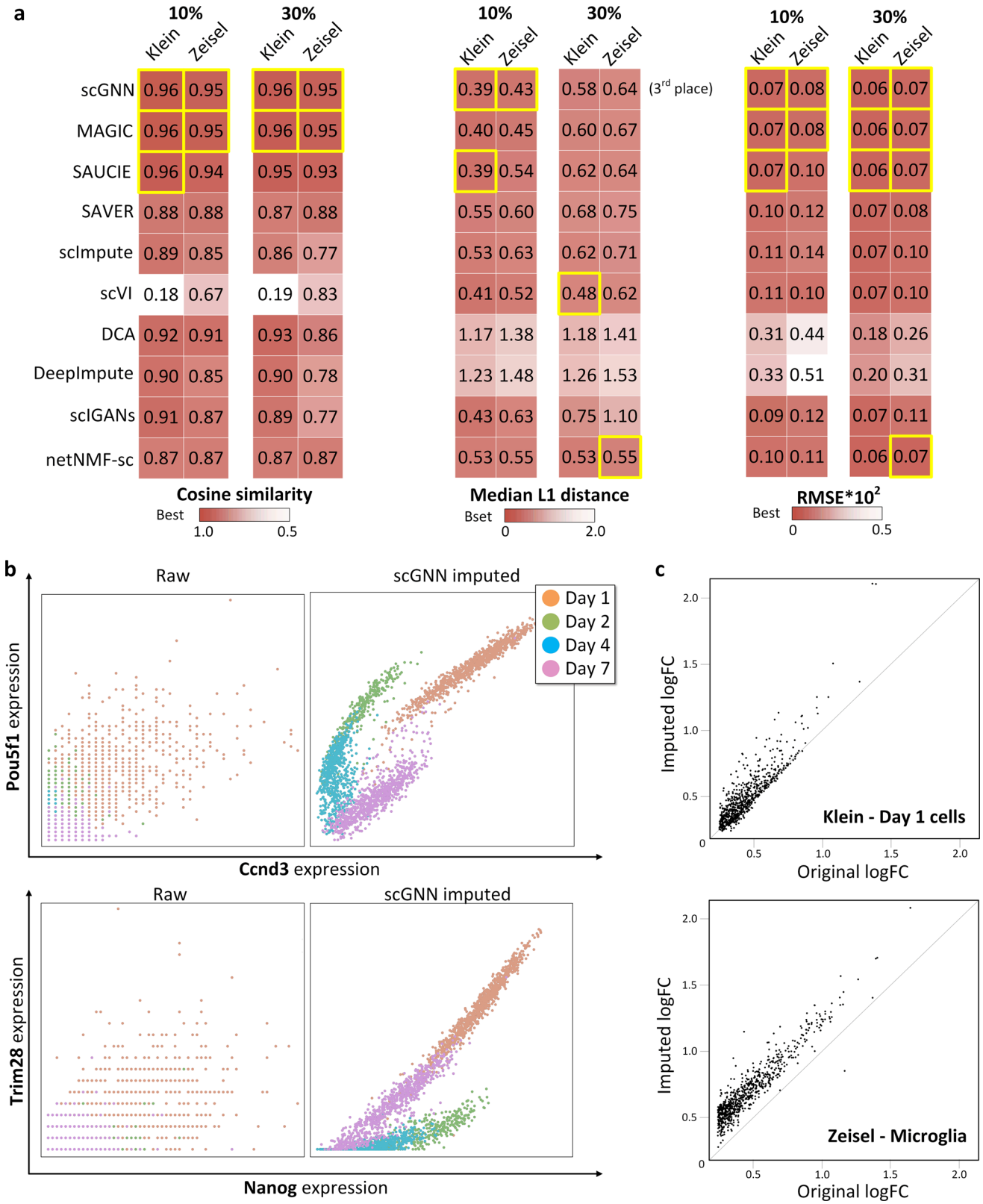


**Fig. 3 Comparison of the imputation performance. a** Comparison of the cosine similarity, median L1 distance, and RMSE scores between scGNN and other nine imputation tools under 10 and 30% synthetic dropout rate. Darker color indicates better performances. The highest score in each column is highlighted with the yellow box. RMSE scores were scaled by multiplying by 100. **b** Co-expression patterns can be addressed more explicitly after applying scGNN on the Klein data. No clear gene pair relationship of *Ccnd3* versus *Pou5f1* (upper panel) and *Nanog* versus *Trim28* (lower panel) is observed in the raw data (left) compared to the observation of unambiguous correlations within each cell type after scGNN imputation (right). **c** Comparison of DEG logFC scores using the original expression value (*x* axis) and the scGNN imputed expression values (*y* axis) identified in day 1 cells of the Klein data (up) and microglial cells of the Zeisel data (bottom). The differentiation signals are amplified after imputation.

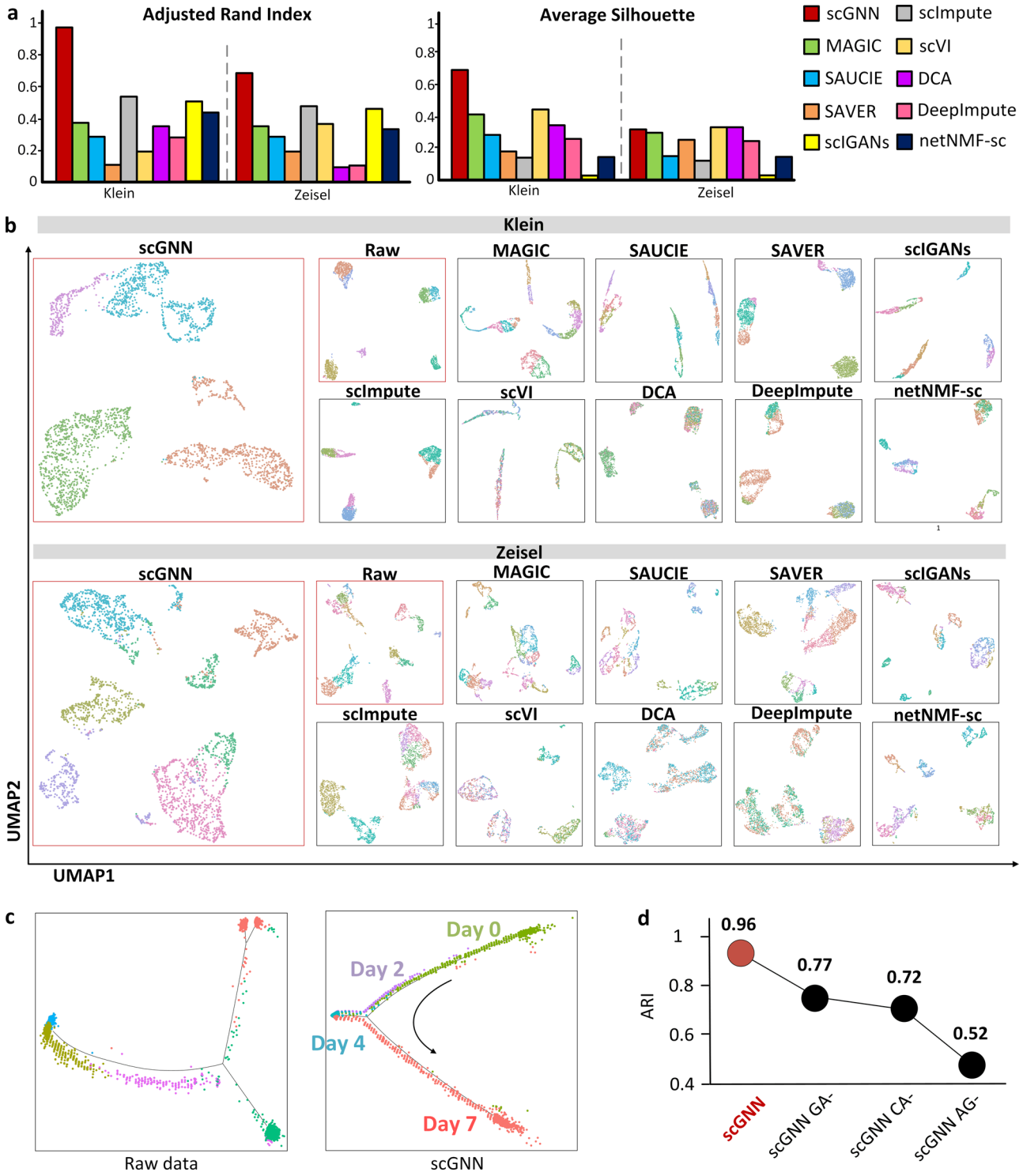


**Fig. 4 Cell clustering and trajectory evaluations. a** Comparison of ARI and Silhouette scores among scGNN and nine tools using Klein and Zeisel data sets. **b** Comparison of UMAP visualizations on the same two data sets, indicating that when scGNN embeddings are utilized, cells are more closely grouped within the same cluster but when other tools are used, cells are more separated between clusters. Cells were clustered via the Louvain method and visualized using UMAP. **c** Pseudotime analysis using the raw expression matrix and scGNN imputed matrix of the Klein data set via Monocle. **d** Justification of using the graph autoencoder, the cluster autoencoder, and the top 2000 variable genes on the Klein data set in the scGNN framework, in terms of ARI. scGNN CA- shows the results of the graph autoencoder's ablation, CA- shows the results of the cluster autoencoder's ablation, and AG shows the results after using all genes in the framework.

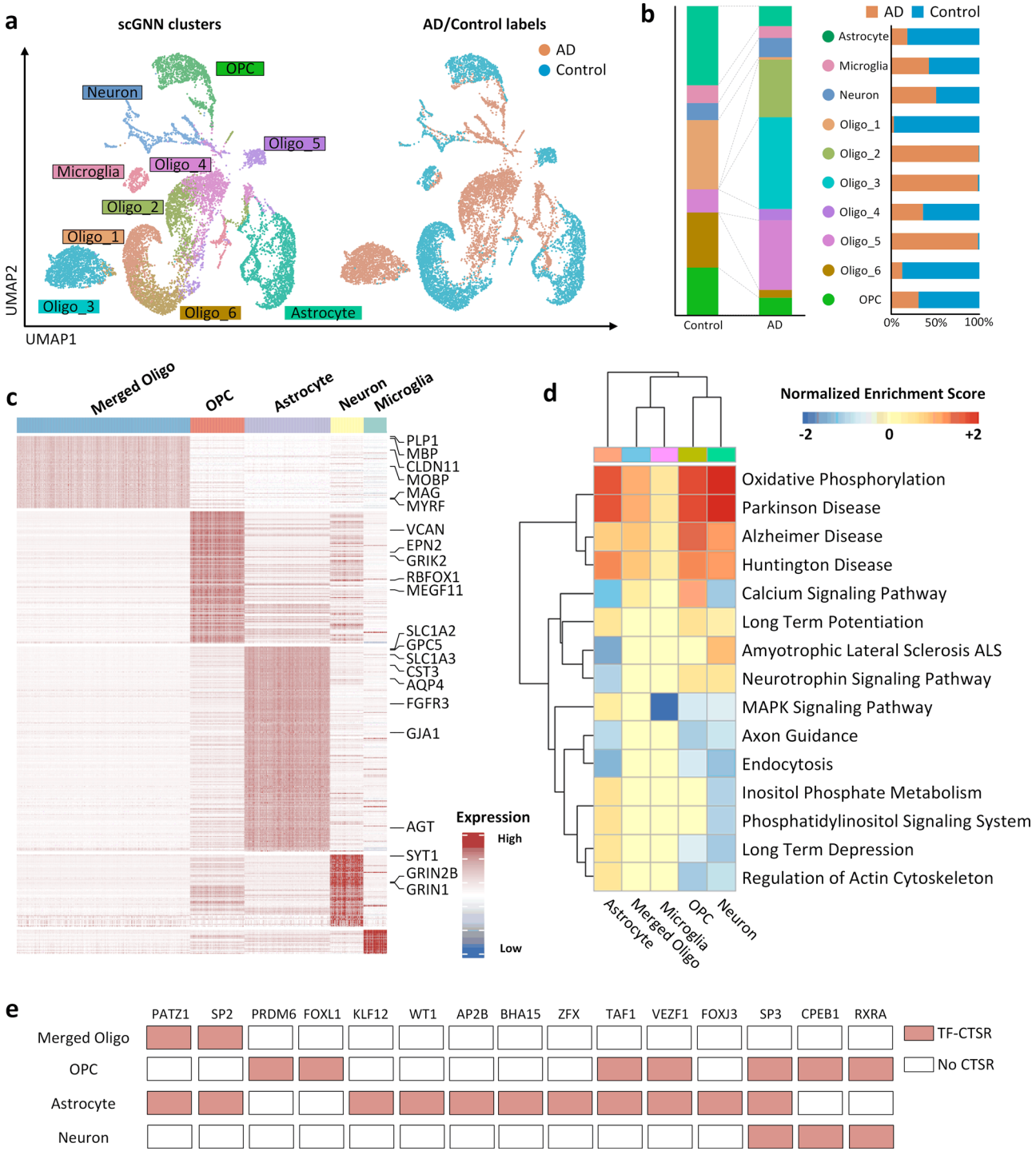


**Fig. 5 Alzheimer's disease data set (GSE138852) analysis based on scGNN. a** Cell clustering UMAP. Labeled with scGNN clusters (left) and AD/control samples (right). **b** Comparison of cell proportions in AD/control samples (left) and each cluster (right). **c** Heatmap of DEGs (logFC > 0.25) in each cluster. Six oligodendrocyte sub-clusters are merged as one to compare with other cell types. Marker genes identified in DEGs are listed on the right. **d** Selected AD-related enrichment pathways in each cell type in the comparison between AD and control cells. **e** Underlying TFs are responsible for the cell-type-specific gene regulations identified by IRIS3.

Methods 1–4, Supplementary Table 2, and Supplementary Data 3–8. We also extensively studied the parameter selection in Supplementary Data 9–12.

**scGNN illustrates AD-related neural development and the underlying regulatory mechanism**. To further demonstrate the applicative power of scGNN, we applied it to a scRNA-Seq data set (GEO accession number GSE138852) containing 13,214 single nuclei collected from six AD and six control brains[27]. scGNN identifies 10 cell clusters, including microglia, neurons, oligodendrocyte progenitor cells (OPCs), astrocytes, and six sub-clusters of oligodendrocytes (Fig. 5a). Specifically, the proportions of these six oligodendrocyte sub-clusters differ between AD patients (Oligos 2, 3, and 4) and healthy controls (Oligos 1, 5, and

6) (Fig. 5b). Moreover, the difference between AD and the control in the proportion of astrocyte and OPCs is observed, indicating the change of cell population in AD patients compared to healthy controls (Fig. 5b). We then combined these six oligodendrocyte sub-clusters into one to discover DEGs. Since scGNN can significantly increase true signals in the raw data set, DEG patterns are more explicit (Supplementary Fig. 10). Among all DEGs, we confirmed 22 genes as cell-type-specific markers for astrocytes, OPCs, oligodendrocytes, and neurons, in that order[43] (Fig. 5c). A biological pathway enrichment analysis shows several highly positive enrichments in AD cells compared to control cells among all five cell types. These enrichments include oxidative phosphorylation and pathways associated with AD, Parkinson's disease, and Huntington disease[44] (Fig. 5d and Supplementary Fig. 11). Interestingly, we observed a strong negative enrichment of the MAPK (mitogen-activated protein kinase) signaling pathway in the microglia cells, suggesting a relatively low MAPK regulation in microglia than other cells.

In order to investigate the regulatory mechanisms underlying the AD-related neural development, we applied the imputed matrix of scGNN to IRIS3 (an integrated cell-type-specific regulon inference server from single-cell RNA-Seq) and identified 21 cell-type-specific regulons (CTSR) in five cell types[45] (Fig. 5e and Supplementary Data 13; IRIS3 job ID: 20200626160833). Not surprisingly, we identified several AD-related transcription factors (TFs) and target genes that have been reported to be involved in the development of AD. SP2 is a common TF identified in both oligodendrocytes and astrocytes. It has been shown to regulate the *ABCA7* gene, which is an IGAP (International Genomics of Alzheimer's Project) gene that is highly associated with late-onset AD[46]. We also observed an SP2 CTSR in astrocytes that regulate *APOE*, *AQP4*, *SLC1A2*, *GJA1*, and *FGFR3*. All of these five targeted genes are marker genes of astrocytes, which have been reported to be associated with AD[47,48]. In addition, the SP3 TF, which can regulate the synaptic function in neurons is identified in all cell clusters, and it is highly activated in AD[49,50]. We identified CTSRs regulated by SP3 in OPCs, astrocytes, and neurons suggesting significant SP3-related regulation shifts in these three clusters. We observed 26, 60, and 22 genes that were uniquely regulated in OPCs, astrocytes, and neurons, as well as 60 genes shared among the three clusters (Supplementary Data 14). Such findings provide a direction for the discovery of SP3 function in AD studies.

## Discussion

It is still a fundamental challenge to explore cellular heterogeneity in high-volume, high-sparsity, and noisy scRNA-Seq data, where the high-order topological relationships of the whole-cell graph are still not well explored and formulated. The key innovations of scGNN are incorporating global propagated topological features of the cells through GNNs, together with integrating gene regulatory signals in an iterative process for scRNA-Seq data analysis. The benefits of GNN are its intrinsic learnable properties of propagating and aggregating attributes to capture relationships across the whole cell–cell graph. Hence, the learned graph embedding can be treated as the high-order representations of cell–cell relationships in scRNA-Seq data in the context of graph topology. Unlike the previous autoencoder applications in scRNA-Seq data analysis, which only captures the top-down distributions of the overall cells, scGNN can effectively aggregate detailed relationships between similar cells using a bottom-up approach. We also observed that the imputation of scGNN can decrease batch effects introduced by different sequencing technologies (Supplementary Fig. 12), which makes scGNN a good choice for data imputation prior to multiple scRNA-Seq data integration[51]. Furthermore, scGNN integrates gene regulatory signals efficiently by representing them discretely in LTMG in the feature autoencoder regularization. These gene regulatory signals can help identify biologically meaningful gene–gene relationships as they apply to our framework and eventually, they are proven capable of enhancing performance. Technically, scGNN adopts multi-modal autoencoders in an iterative manner to recover gene expression values and cell-type prediction simultaneously. Notably, scGNN is a hypothesis-free deep learning framework on a data-driven cell graph model, and it is flexible to incorporate different statistical models (e.g., LTMG) to analyze complex scRNA-Seq data sets.

Some limitations can still be found in scGNN. (i) It is prone to achieve better results with large data sets, compared to relatively small data sets (e.g., <1000 cells), as it is designed to learn better representations with many cells from scRNA-Seq data, as shown in the benchmark results, and (ii) Compared with statistical model-based methods, the iterative autoencoder framework needs more computational resources, which is more time-consuming (Supplementary Data 15). In the future, we will investigate creating a more efficient scGNN model with a lighter and more compressed architecture.

In the future, we will continue to enhance scGNN by implementing heterogeneous graphs to support the integration of single-cell multi-omics data (e.g., the intra-modality of Smart-Seq2 and Droplet scRNA-Seq data; and the inter-modality integration of scRNA-Seq and scATAC-Seq data). We will also incorporate attention mechanisms and graph transformer models[52] to make the analyses more explainable. Specifically, by allowing the integration of scRNA-Seq and scATAC-Seq data, scGNN has the potential to elucidate cell-type-specific gene regulatory mechanisms[53]. On the other hand, T cell receptor repertoires are considered as unique identifiers of T cell ancestries that can improve both the accuracy and robustness of predictions regarding cell–cell interactions[54]. scGNN can also facilitate batch effects and build connections across diverse sequencing technologies, experiments, and modalities. Moreover, scGNN can be applied to analyze spatial transcription data sets regarding spatial coordinates as additional regularizers to infer the cell neighborhood representation and better prune the cell graph. We plan to develop a more user-friendly software system from our scGNN model, together with modularized analytical functions in support of standardizing the data format, quality control, data integration, multi-functional scMulti-seq analyses, performance evaluations, and interactive visualizations.

## Methods

**Data set preprocessing**. scGNN takes the scRNA-Seq gene expression profile as the input. Data filtering and quality control are the first steps of data preprocessing. Due to the high dropout rate of scRNA-seq expression data, only genes expressed as non-zero in more than 1% of cells, and cells expressed as non-zero in more than 1% of genes are kept. Then, genes are ranked by standard deviation, i.e., the top 2000 genes in variances are used for the study. All the data are log-transformed.

**Left-truncated mixed Gaussian (LTMG) modeling**. A mixed Gaussian model with left truncation assumption is used to explore the regulatory signals from gene expression[14]. The normalized expression values of gene $X$ over $N$ cells are denoted as $X = \{x_1,\ldots x_N\}$, where $x_j \in X$ is assumed to follow a mixture of $k$ Gaussian distributions, corresponding to $k$ possible gene regulatory signals (TRSs). The density function of $X$ is:

$$p(X;\Theta) = \prod_{j=1}^{N} p(x_j;\Theta) = \prod_{j=1}^{N} \sum_{i=1}^{k} \alpha_i p\left(x_j;\theta_i\right) = \prod_{j=1}^{N} \sum_{i=1}^{k} \alpha_i \frac{1}{\sqrt{2\pi}\sigma_i} e^{\frac{-\left(x_j-\mu_i\right)^2}{2\sigma_i^2}} = L(\Theta;X) \quad (1)$$

where $\alpha_i$ is the mixing weight, $\mu_i$ and $\sigma_i$ are the mean and standard deviation of the $i$th Gaussian distribution, which can be estimated by: $\Theta^* = \underset{\Theta}{\arg\max}\, L(\Theta;X)$ to model the errors at zero and the low expression values. With the left truncation

assumption, the gene expression profile is split into $M$, which is a truly measured expression of values, and $N - M$ representing left-censored gene expressions for $N$ conditions. The parameter $\Theta$ maximizes the likelihood function and can be estimated by an expectation-maximization algorithm. The number of Gaussian components is selected by the Bayesian Information Criterion; then, the original gene expression values are labeled to the most likely distribution under each cell. In detail, the probability that $x_j$ belongs to distribution $i$ is formulated by:

$$p\left(x_j \in \mathrm{TRS}\, i|K,\Theta^*\right) \propto \frac{\alpha_i}{\sqrt{2\pi\sigma_j^2}} e^{\frac{-(x_j-\mu_i)^2}{2\sigma_i^2}} \quad (2)$$

where $x_j$ is labeled by TRS $i$ if $p\left(x_j \in \mathrm{TRS}\, i|K,\Theta^*\right) = \max_{i=1,\ldots,K}(p\left(x_j \in \mathrm{TRS}\, i|K,\Theta^*\right))$. Thus, the discrete values (1,2, …, $K$) for each gene are generated.

**Feature autoencoder**. The feature autoencoder is proposed to learn the representative embedding of the scRNA expression through stacked two layers of dense networks in both the encoder and decoder. The encoder constructs the low-dimensional embedding of $X'$ from the input gene expression $X$, and the encoder reconstructs the expression $\hat{X}$ from the embedding; thus, $X, \hat{X} \in \mathbb{R}^{N\times M}$ and $X' \in \mathbb{R}^{N\times M'}$, where $M$ is the number of input genes, $M'$ is the dimension of the learned embedding, and $M' < M$. The objective of training the feature autoencoder is to achieve a maximum similarity between the original and reconstructed through minimizing the loss function, in which $\sum(X-\hat{X})^2$ is the main term serving as the mean squared error (MSE) between the original and the reconstructed expressions.

**Regularization**. Regularization is adopted to integrate gene regulation information during the feature autoencoder training process. The aim of this regularization is to treat each gene differently based on their individual gene regulation role through penalizing it in the loss function. The MSE is defined as:

$$\alpha\sum\left((X-\hat{X})^2 \circ \mathrm{TRS}\right) \quad (3)$$

where $\mathrm{TRS} \in \mathbb{R}^{N\times M}$; $\alpha$ is a parameter used to control the strength of gene regulation regularization; $\alpha \in [0,1]$. $\circ$ denotes element-wise multiplication. Thus, the loss function of the feature autoencoder is shown as Eq.(4).

$$\mathrm{Loss} = (1-\alpha)\sum(X-\hat{X})^2 + \alpha\sum\left((X-\hat{X})^2 \circ \mathrm{TRS}\right) \quad (4)$$

In the encoder, the output dimensions of the first and second layers are set as 512 and 128, respectively. Each layer is followed by the ReLU activation function. In the decoder, the output dimensions of the first and second layers are 128 and 512, respectively. Each layer is followed by a sigmoid activation function. The learning rate is set as 0.001. The cluster autoencoder has the same architecture as the feature autoencoder, but without gene regulation regularization in the loss function.

**Cell graph and pruning**. The cell graph formulates the cell–cell relationships using embedding learned from the feature autoencoder. As done in the previous works[4,55], the cell graph is built from a KNN graph, where nodes are individual single cells, and the edges are relationships between cells. $K$ is the predefined parameter used to control the scale of the captured interaction between cells. Each node finds its neighbors within the $K$ shortest distances and creates edges between them and itself. Euclidian distance is calculated as the weights of the edges on the learned embedding vectors. The pruning process selects an adaptive number of neighbors for each node on the original KNN graph and keeps a more biologically meaningful cell graph. Here, Isolation Forest is applied to prune the graph to detect the outliner in the $K$-neighbors of each node[56]. Isolation Forest builds individual random forest to check distances from the node to all $K$-neighbors and only disconnects the outliners.

**Graph autoencoder**. The graph autoencoder learns to embed and represent the topological information from the pruned cell graph. For the input pruned cell graph, $G = (V, E)$ with $N = |V|$ nodes denoting the cells and $E$ representing the edges. $A$ is its adjacency matrix and $D$ is its degree matrix. The node feature matrix of the graph autoencoder is the learned embedding $X'$ from the feature autoencoder.

The graph convolution network (GCN) is defined as $\mathrm{GCN}(X', A) = \mathrm{ReLU}(\tilde{A}X'W)$, and $W$ is a weight matrix learned from the training. $\tilde{A} = D^{-1/2}AD^{-1/2}$ is the symmetrically normalized adjacency matrix and activation function $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$. The encoder of the graph autoencoder is composed of two layers of GCN, and $Z$ is the graph embedding learned through the encoder in Eq.(5). $W_1$ and $W_2$ are learned weight matrices in the first and second layers, and the output dimensions of the first and second layers are set at 32 and 16, respectively. The learning rate is set at 0.001.

$$Z = \mathrm{ReLU}(\tilde{A}ReLU\left(\tilde{A}X'W_1\right)W_2) \quad (5)$$

The decoder of the graph autoencoder is defined as an inner product between the embedding:

$$\hat{A} = \mathrm{sigmoid}(ZZ^T) \quad (6)$$

where $\hat{A}$ is the reconstructed adjacency matrix of $A$. $\mathrm{sigmoid}(\cdot) = 1/(1+e^{-\cdot})$ is the sigmoid activation function.

The goal of learning the graph autoencoder is to minimize the cross-entropy $L$ between the input adjacency matrix $A$ and the reconstructed matrix $\hat{A}$:

$$L(A,\hat{A}) = -\frac{1}{N\times N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(a_{ij} * \log\left(\hat{a}_{ij}\right) + \left(1-a_{ij}\right) * \log(1-\hat{a}_{ij})\right) \quad (7)$$

where $a_{ij}$ and $\hat{a}_{ij}$ are the elements of the adjacency matrix $A$ and $\hat{A}$ in the $i$th row and the $j$th column. As there are $N$ nodes as the cell number in the graph, $N \times N$ is the total number of elements in the adjacency matrix.

**Iterative process**. The iterative process aims to build the single-cell graph iteratively until converging. The iterative process of the cell graph can be defined as:

$$\tilde{A} = \lambda L_0 + (1-\lambda)\frac{A_{ij}}{\sum_j A_{ij}} \quad (8)$$

where $L_0$ is the normalized adjacency matrix of the initial pruned graph, and $L_0 = D_0^{-1/2}A_0D_0^{-1/2}$, where $D_0$ is the degree matrix. $\lambda$ is the parameter to control the converging speed, $\lambda \in [0,1]$. Each time in iteration $t$, two criteria are checked to determine whether to stop the iteration: (1) that is, to determine whether the adjacency matrix converges, i.e., $\tilde{A}_t - \tilde{A}_{t-1} < \gamma_1\tilde{A}_0$, or (2) whether the inferred cell types are similar enough, i.e., ARI < $\gamma_2$. ARI is the similarity measurement, which is detailed in the next section. In our setting, $\lambda = 0.5$ and $\gamma_1$, $\gamma_2 = 0.99$. The cell-type clustering results obtained in the last iteration are chosen as the final cell-type results.

**Imputation autoencoder**. After the iterative process stops, the imputation autoencoder imputes and denoises the raw expression matrix within the inferred cell–cell relationship. The imputation autoencoder shares the same architecture as the feature autoencoder, but it also uses three additional regularizers from the cell graph in Eq. (9), cell types in Eq. (10), and the L1 regularizer in Eq. (11):

$$\gamma_1\sum(A\cdot(X-\hat{X})^2) \quad (9)$$

where $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix from the pruned cell graph in the last iteration. $\cdot$ denotes dot product. Cells within an edge in the pruned graph will be penalized in the training:

$$B_{ij} = \begin{cases} \gamma_2\sum(B\cdot(X-\hat{X})^2) \\ 1 \quad \text{where } i \text{ and } j \text{ in same cell type} \\ 0 \quad \text{else} \end{cases} \quad (10)$$

where $B \in \mathbb{R}^{N\times N}$ is the relationship matrix between cells, and two cells in the same cell type have a $B_{ij}$ value of 1. Cells within the same inferred cell type will be penalized in the training. $\gamma_1$, $\gamma_2$ are the intensities of the regularizers and $\gamma_1$, $\gamma_2 \in [0,1]$. The L1 regularizer is defined as

$$\beta\sum|w| \quad (11)$$

which brings sparsity and increases the generalization performance of the autoencoder by reducing the number of non-zero $w$ terms in $\sum|w|$, where $\beta$ is a hyperparameter controlling the intensity of the L1 term ($\beta \in [0,1]$). Therefore, the loss function of the imputation autoencoder is

$$\begin{aligned}\mathrm{Loss} = (1-\alpha)\sum(X-\hat{X})^2 + \alpha\sum\left((X-\hat{X})^2 \circ \mathrm{TRS}\right) \\ + \beta\sum|w| + \gamma_1\sum\left(A\cdot(X-\hat{X})^2\right) + \gamma_2\sum(B\cdot(X-\hat{X})^2)\end{aligned} \quad (12)$$

**Benchmark evaluation compared to existing tools**

*Imputation evaluation*. For benchmarking imputation performance, we performed synthetic dropout simulation to randomly flip 10% of the non-zero entries to zeros. These synthetic dropouts still follow the zero-inflated negative binomial (ZINB) distribution with details shown in Supplementary Method 5 and Data 16. We evaluated median L1 distance, cosine similarity, and root-mean-squared error (RMSE) between the original data set and the imputed values for these corrupted entries. For all the flipped entries, $x$ is the row vector of the original expression, and $y$ is its corresponding row vector of the imputed expression. The L1 distance is the absolute deviation between the value of the original and imputed expression. A lower L1 distance means a higher similarity.

$$\text{L1 distance} = |x-y|, \quad \text{L1 distance} \in [0,+\infty) \quad (13)$$

The cosine similarity computes the dot products between original and imputed expression.

$$\text{Cosine similarity}(x, y) = \frac{xy^T}{\|x\| \, \|y\|}, \quad \text{Cosine similarity} \in [0, 1] \tag{14}$$

The RMSE computes the squared root of the quadratic mean of differences between original and imputed expression.

$$\text{RMSE}(x, y) = \sqrt{\frac{\sum_{i=1}^{N} (x_i - y_i)^2}{N}}, \ \text{RMSE} \in [0, +\infty) \tag{15}$$

The process is repeated three times, and the mean and standard deviation were selected as a comparison. The scores are compared between scGNN and nine imputation tools (i.e., MAGIC[4], SAUCIE[10], SAVER[19], scImpute[33], scVI[32], DCA[11], DeepImpute[34], scIGANs[35], and netNMF-sc[36]), using the default parameters.

*Clustering evaluation*. We compared the cell clustering results of scGNN, the same nine imputation tools, and four scRNA-Seq analytical frameworks, in terms of ten clustering evaluation scores. Noted that, we considered the default cell clustering method (i.e., Louvain method[31] in Seurat[5], Ward.D2[57] method in CIDR[58], Louvain method in Monocle[59], and *k*-means[60] method in RaceID[61]) in each of the analytical frameworks to compare the cell clustering performance with scGNN. The default parameters are applied in all test tools. ARI[37] is used to compute similarities by considering all pairs of the samples that are assigned in clusters in the current and previous clustering adjusted by random permutation:

$$\text{ARI} = \frac{\text{RI} - \text{E[RI]}}{\max(\text{RI}) - \text{E[RI]}} \tag{16}$$

where the unadjusted rand index (RI) is defined as:

$$\text{RI} = \frac{a + b}{C_n^2} \tag{17}$$

where $a$ is the number of pairs correctly labeled in the same sets, and b is the number of pairs correctly labeled as not in the same data set. $C_n^2$ is the total number of possible pairs. E[RI] is the expected RI of random labeling.

Different from ARI which requires known ground truth labels, the Silhouette coefficient score[38] defines how similar an object is to its own cluster compared to other clusters. It is defined as:

$$\text{Silhouette} = \frac{b - a}{\max(a, b)} \tag{18}$$

where $a$ is the mean distance between a sample and all other points in the same class, $b$ is the mean distance between a sample and all other points in the next nearest cluster. Silhouette $\in [-1,1]$, where 1 indicates the best clustering results and $-1$ indicates the worst. We calculated the average Silhouette score of all cells in each data set to compare the cell clustering results. More quantitative measurements are also used in Supplementary Method 4.

**Statistical validation of cell–cell graph topology based on LRPs**. We used CellChat[42] to predict potential interaction probability scores (ranging from 0 to 1; a higher score indicates the two cells are more likely to interact with each other) of ligand–receptor pairs (LRP) between any two cells. We built a fully connected cell–cell background graph (using all the cells) based on Pearson's correlation of the raw expression matrix and compared it with the cell–cell graph generated from scGNN. CellChat calculates an aggregated interaction probability for each linked cell pair based on the expression level of LRPs. For all linked cell pairs in the background graph and scGNN cell–cell graph, we performed a Wilcoxon test to evaluate the statistical significance between the corresponding aggregated interaction probability. Five scRNA-Seq data sets (i.e., Klein, Zeisel, Kolo, Chung, and AD) were used in this analysis.

**Case study of the AD database**. We applied scGNN on public Alzheimer's disease (AD) scRNA-Seq data with 13,214 cells[27]. The resolution of scGNN was set to 1.0, KI was set to 20, and the remaining parameters were kept as default. The AD patient and control labels were provided by the original paper and used to color the cells on the same UMAP coordinates generated from scGNN. We simply combined cells in six oligodendrocyte subpopulations into one cluster, referred to as merged oligo. The DEGs were identified in each cell cluster via the Wilcoxon rank-sum test implemented in the Seurat package along with adjusted *p*-values using the Benjamini-Hochberg procedure with a nominal level of 0.05. DEGs with logFC > 0.25 or <−0.25 were finally selected. We further identified the DEGs between AD and control cells in each cluster using the same strategy and applied GSEA for pathway enrichment analysis[62]. The imputed matrix, which resulted from scGNN was then sent to IRIS3 for CTSR prediction, using the predicted cell clustering labels with merged oligodendrocytes[45]. The default parameters were served in regulatory analysis in IRIS3.

**Software implementation**. Tools and packages used in this paper include: Python version 3.7.6, numpy version 1.18.1, torch version 1.4.0, networkx version 2.4, pandas version 0.25.3, rpy2 version 3.2.4, matplotlib version 3.1.2, seaborn version 0.9.0, umap-learn version 0.3.10, munkres version 1.1.2, R version 3.6.1, and igraph version 1.2.5. The IRIS3 website is at https://bmbl.bmi.osumc.edu/iris3/index.php.

**Reporting summary**. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

## Data availability

The scRNA-seq data sets analyzed during the current study are publicly available. Three benchmark and AD case data sets can be downloaded from Gene Expression Omnibus (GEO) databases with accession numbers of GSE75688 (the Chung data); GSE65525 (the Klein data); GSE60361 (the Zeisel data); and GSE138852 (the AD case). The Kolodziejczy data can be accessed from EMBL-EBI with an accession number of E-MTAB-2600.

## Code availability

Our tool is open source and publicly available at GitHub and Zenodo (https://github.com/juexinwang/scGNN)[63].

Received: 27 July 2020; Accepted: 24 February 2021;
Published online: 25 March 2021

## References

1. Hwang, B., Lee, J. H. & Bang, D. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp. Mol. Med.* **50**, 96 (2018).
2. Gawel, D. R. et al. A validated single-cell-based strategy to identify diagnostic and therapeutic targets in complex diseases. *Genome Med.* **11**, 47 (2019).
3. Levine, J. H. et al. Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. *Cell* **162**, 184–197 (2015).
4. van Dijk, D. et al. Recovering gene interactions from single-cell data using data diffusion. *Cell* **174**, 716–729.e727 (2018).
5. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. Biotechnol.* **36**, 411–420 (2018).
6. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. In *The International Conference on Learning Representations (ICLR)* (2017).
7. Wang, J., Ma, A., Ma, Q., Xu, D. & Joshi, T. Inductive inference of gene regulatory network using supervised and semi-supervised graph neural networks. *Comput. Struct. Biotechnol. J.* **18**, 3335–3343 (2020).
8. Fang, C., Xu, D., Su, J., Dry, J. R. & Linghu, B. DeePaN: deep patient graph convolutional network integrating clinico-genomic evidence to stratify lung cancers for immunotherapy. *npj Digit. Med.* **4**, 14 (2021).
9. Wang, W., Huang, Y., Wang, Y. & Wang, L. Generalized autoencoder: a neural network framework for dimensionality reduction. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops* 496–503, https://doi.org/10.1109/CVPRW.2014.79 (2014)
10. Amodio, M. et al. Exploring single-cell data with deep multitasking neural networks. *Nat. Methods* **16**, 1139–1145 (2019).
11. Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell RNA-seq denoising using a deep count autoencoder. *Nat. Commun.* **10**, 390 (2019).
12. Miao, Z. et al. Putative cell type discovery from single-cell gene expression data. *Nat. Methods* **17**, 621–628 (2020).
13. Kipf, T. N. & Welling, M. Variational graph auto-encoders. Preprint at https://arxiv.org/abs/1611.07308 (2016).
14. Wan, C. et al. LTMG: a novel statistical modeling of transcriptional expression states in single-cell RNA-Seq data. *Nucleic Acids Res.* **47**, e111 (2019).
15. Huang, M. et al. SAVER: gene expression recovery for single-cell RNA sequencing. *Nat. Methods* **15**, 539 (2018).
16. Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S. & Vert, J. P. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nat. Commun.* **9**, 284 (2018).
17. Svensson, V. et al. Power analysis of single-cell RNA-sequencing experiments. *Nat. Methods* **14**, 381–387 (2017).
18. Tian, L. et al. Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments. *Nat. Methods* **16**, 479–487 (2019).
19. Huang, M. et al. SAVER: gene expression recovery for single-cell RNA sequencing. *Nat. Methods* **15**, 539–542 (2018).
20. Wang, J. et al. Data denoising with transfer learning in single-cell transcriptomics. *Nat. Methods* **16**, 875–878 (2019).
21. Zhang, L. & Zhang, S. Comparison of computational methods for imputing single-cell RNA-sequencing data. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* **17**, 376–389 (2020).
22. Liu, B. et al. An entropy-based metric for assessing the purity of single cell populations. *Nat. Commun.* **11**, 3155 (2020).

23. Kolodziejczyk, A. A. et al. Single cell RNA-sequencing of pluripotent states unlocks modular transcriptional variation. *Cell Stem Cell* **17**, 471–485 (2015).
24. Klein, A. M. et al. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell* **161**, 1187–1201 (2015).
25. Zeisel, A. et al. Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* **347**, 1138–1142 (2015).
26. Chung, W. et al. Single-cell RNA-seq enables comprehensive tumour and immune cell profiling in primary breast cancer. *Nat. Commun.* **8**, 15081 (2017).
27. Grubman, A. et al. A single-cell atlas of entorhinal cortex from individuals with Alzheimer's disease reveals cell-type-specific gene expression regulation. *Nat. Neurosci.* **22**, 2087–2097 (2019).
28. Xie, J. et al. QUBIC2: a novel and robust biclustering algorithm for analyses and interpretation of large-scale RNA-Seq data. *Bioinformatics* **36**, 1143–1149 (2020).
29. Bendall, S. C. et al. Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development. *Cell* **157**, 714–725 (2014).
30. Wolf, F. A. et al. PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol.* **20**, 59 (2019).
31. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, P10008 (2008).
32. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat. Methods* **15**, 1053–1058 (2018).
33. Li, W. V. & Li, J. J. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nat. Commun.* **9**, 997 (2018).
34. Arisdakessian, C., Poirion, O., Yunits, B., Zhu, X. & Garmire, L. X. DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. *Genome Biol.* **20**, 211 (2019).
35. Xu, Y. et al. scIGANs: single-cell RNA-seq imputation using generative adversarial networks. *Nucleic Acids Res.* **48**, e85 (2020).
36. Elyanow, R., Dumitrascu, B., Engelhardt, B. E. & Raphael, B. J. netNMF-sc: leveraging gene-gene interactions for imputation and dimensionality reduction in single-cell expression analysis. *Genome Res.* **30**, 195–204 (2020).
37. Hubert, L. & Arabie, P. Comparing partitions. *J. Classif.* **2**, 193–218 (1985).
38. Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987).
39. McInnes, L., Healy, J. & Melville, J. Umap: uniform manifold approximation and projection for dimension reduction. Preprint at https://arxiv.org/abs/1802.03426 (2018).
40. Hinton, G. E. & Roweis, S. Stochastic neighbor embedding. *Adv. Neural Inf. Process Syst.* **15**, 857–864 (2002).
41. Moon, K. R. et al. Visualizing structure and transitions in high-dimensional biological data. *Nat. Biotechnol.* **37**, 1482–1492 (2019).
42. Armingol, E., Officer, A., Harismendy, O. & Lewis, N. E. Deciphering cell-cell interactions and communication from gene expression. *Nat. Rev. Genet.* 1–18, https://doi.org/10.1038/s41576-020-00292-x (2020).
43. Tanzi, R. E. The genetics of Alzheimer disease. *Cold Spring Harb. Perspect. Med.* **2**, a006296 (2012).
44. Su, B. et al. Oxidative stress signaling in Alzheimer's disease. *Curr. Alzheimer Res.* **5**, 525–532 (2008).
45. Ma, A. et al. IRIS3: integrated cell-type-specific regulon inference server from single-cell RNA-Seq. *Nucleic Acids Res.* https://doi.org/10.1093/nar/gkaa394 (2020).
46. Karch, C. M., Ezerskiy, L. A., Bertelsen, S., Goate, A. M. & Alzheimer's Disease Genetics Consortium. Alzheimer's disease risk polymorphisms regulate gene expression in the ZCWPW1 and the CELF1 loci. *PLoS ONE* **11**, e0148717 (2016).
47. Franzén, O., Gan, L.-M. & Björkegren, J. L. M. PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data. *Database* **2019**, https://doi.org/10.1093/database/baz046 (2019).
48. Mathys, H. et al. Single-cell transcriptomic analysis of Alzheimer's disease. *Nature* **570**, 332–337 (2019).
49. Yamakawa, H. et al. The transcription factor Sp3 cooperates with HDAC2 to regulate synaptic function and plasticity in neurons. *Cell Rep.* **20**, 1319–1334 (2017).
50. Boutillier, S. et al. Sp3 and sp4 transcription factor levels are increased in brains of patients with Alzheimer's disease. *Neuro-degen. Dis.* **4**, 413–423 (2007).
51. Büttner, M., Miao, Z., Wolf, F. A., Teichmann, S. A. & Theis, F. J. A test metric for assessing single-cell RNA-seq batch correction. *Nat. Methods* **16**, 43–49 (2019).
52. Hu, Z., Dong, Y., Wang, K. & Sun, Y. Heterogeneous Graph Transformer. In *Proc. Web Conference 2020* 2704–2710 (2020).
53. Ma, A., McDermaid, A., Xu, J., Chang, Y. & Ma, Q. Integrative methods and practical challenges for single-cell multi-omics. *Trends Biotechnol.* https://doi.org/10.1016/j.tibtech.2020.02.013 (2020).
54. Han, A., Glanville, J., Hansmann, L. & Davis, M. M. Linking T-cell receptor sequence to functional phenotype at the single-cell level. *Nat. Biotechnol.* **32**, 684–692 (2014).
55. Grün, D. Revealing dynamics of gene expression variability in cell state space. *Nat. Methods* **17**, 45–49 (2020).
56. Liu, F. T., Ting, K. M. & Zhou, Z. in *2008 Eighth IEEE International Conference on Data Mining* 413–422 (2008).
57. Murtagh, F. & Legendre, P. Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? *J. Classif.* **31**, 274–295 (2014).
58. Lin, P., Troup, M. & Ho, J. W. K. H. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol.* **18**, 59 (2017).
59. Qiu, X. et al. Reversed graph embedding resolves complex single-cell trajectories. *Nat. Methods* **14**, 979–982 (2017).
60. Hartigan, J. A. & Wong, M. A. Algorithm AS 136: a K-means clustering algorithm. *J. R. Stat. Soc. Ser. C* **28**, 100–108 (1979).
61. Lin, P., Troup, M. & Ho, J. W. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol.* **18**, 59 (2017). PMC5371246.
62. Subramanian, A. et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl Acad. Sci. USA* **102**, 15545–15550 (2005). PMC1239896.
63. Juexin Wang, A. M. et al. scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses. *juexinwang/scGNN* https://doi.org/10.5281/zenodo.4540635 (2021).


## Acknowledgements

This work was supported by awards R35-GM126985 and R01-GM131399 from the National Institute of General Medical Sciences of the National Institutes of Health. The work was also supported by award NSF1945971 from the National Science Foundation. We thank Ms. Carla Roberts for thoroughly proofreading this paper.

## Author contributions

Conceptualization: Q.M. and D.X.; methodology: J.W., A.M., Q.M., and D.X.; software coding: J.W. and Y.C.; data collection and investigation: J.W., A.M., and R.Q.; data analysis: A.M., J.W., J.G., Y.C., and Y.J.; software testing and tutorial: J.W., J.G., R.Q., Y.J., and C.W.; manuscript writing, review, and editing: J.W., A.M., H.F., Q.M., and D.X.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at https://doi.org/10.1038/s41467-021-22197-x.

**Correspondence** and requests for materials should be addressed to Q.M. or D.X.

**Peer review information** *Nature Communications* thanks Guy Wolf and Yuedong Yang for their contribution to the peer review of this work.

**Reprints and permission information** is available at http://www.nature.com/reprints

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.