Genomics Proteomics Bioinformatics 20 (2022) 814–835

Genomics Proteomics Bioinformatics

www.elsevier.com/locate/gpb
www.sciencedirect.com

REVIEW

Application of Deep Learning on Single-cell RNA Sequencing Data Analysis: A Review

Matthew Brendel [1,2,#], Chang Su [3,#,*], Zilong Bai [1], Hao Zhang [1], Olivier Elemento [2], Fei Wang [1,*]

[1] *Department of Population Health Sciences, Weill Cornell Medicine, Cornell University, New York, NY 10065, USA*
[2] *Institute for Computational Biomedicine, Caryl and Israel Englander Institute for Precision Medicine, Department of Physiology and Biophysics, Weill Cornell Medicine, Cornell University, New York, NY 10065, USA*
[3] *Department of Health Service Administration and Policy, Temple University, Philadelphia, PA 19122, USA*

Received 23 January 2022; revised 17 August 2022; accepted 24 November 2022
Available online 14 December 2022

Handled by Feng Gao

KEYWORDS

Single-cell RNA sequencing;
Single-cell sequencing;
Deep learning;
Deep neural network;
Artificial intelligence

Abstract **Single-cell RNA sequencing** (scRNA-seq) has become a routinely used technique to quantify the gene expression profile of thousands of single cells simultaneously. Analysis of scRNA-seq data plays an important role in the study of cell states and phenotypes, and has helped elucidate biological processes, such as those occurring during the development of complex organisms, and improved our understanding of disease states, such as cancer, diabetes, and coronavirus disease 2019 (COVID-19). **Deep learning**, a recent advance of **artificial intelligence** that has been used to address many problems involving large datasets, has also emerged as a promising tool for scRNA-seq data analysis, as it has a capacity to extract informative and compact features from noisy, heterogeneous, and high-dimensional scRNA-seq data to improve downstream analysis. The present review aims at surveying recently developed deep learning techniques in scRNA-seq data analysis, identifying key steps within the scRNA-seq data analysis pipeline that have been advanced by deep learning, and explaining the benefits of deep learning over more conventional analytic tools. Finally, we summarize the challenges in current deep learning approaches faced within scRNA-seq data and discuss potential directions for improvements in deep learning algorithms for scRNA-seq data analysis.

Introduction

Since the first single-cell RNA sequencing (scRNA-seq) paper in 2009 [1] and subsequent designation of "method of the year" a few years after [2–6], there has been a considerable amount of effort to advance both the experimental and computational techniques used for the study of single-cell transcriptomes.

* Corresponding authors.
E-mail: su.chang@temple.edu (Su C), few2001@med.cornell.edu (Wang F).
# Equal contribution.

Peer review under responsibility of Beijing Institute of Genomics, Chinese Academy of Sciences / China National Center for Bioinformation and Genetics Society of China

https://doi.org/10.1016/j.gpb.2022.11.011

The benefit of scRNA-seq, compared to bulk RNA sequencing (RNA-seq), is the ability to interrogate thousands of individual cells simultaneously, thus revealing previously hidden heterogeneous cellular populations. scRNA-seq can then be used to answer biological questions related to developmental processes, understand complex and heterogeneous cellular or genetic changes based on treatment conditions or disease states, or identify novel cell types within a cellular population. Many popular packages, such as Seurat [2], Scanpy [3], Monocle [4], and Orchestrating Single-Cell Analysis (OSCA) with Bioconductor [5], have been developed for a streamlined and reproducible analysis of scRNA-seq data. A pipeline for scRNA-seq analysis typically contains three steps (**Figure 1**): 1) scRNA-seq data collection that produces a gene by cell matrix, of which elements are the raw gene expression read counts or unique molecular identifiers (UMIs), normalized to account for total genes captured for a particular cell either using standard approaches such as log or square root normalization, or more advanced approaches such as SCTransform [6]; 2) data preprocessing including representation learning and dimensionality reduction, as well as optional doublet removal, cell cycle variance removal, data imputation and denoising, and batch effect removal; and 3) downstream analyses, such as cell clustering, cell type annotation, and trajectory inference for discovery of cellular dynamic process along the development of cells [7]. The result of this process can be used to answer biological questions of interest or determine unique features about the cellular populations that have been discovered.

Machine learning, a branch of artificial intelligence relying on mathematical and statistical principles, uses sets of data to build models that can perform specific tasks of interest and help accelerate or improve human decision making. In recent years, machine learning has successfully been used to analyze high-throughput omics data to improve upon the understanding of biological mechanisms of human health conditions [8,9]. Conventional machine learning approaches usually require a significant amount of effort to develop a feature engineering strategy designed by domain experts, especially in the analysis of uncertain, heterogenous, and high-dimensional data like scRNA-seq data. As one of the latest and most popular advanced sub-categories of machine learning, deep learning provides a methodology that is more powerful in discovering latent and informative patterns from complex data and has achieved extraordinary improvements in computer vision and natural language processing tasks. Importantly, compared to conventional machine learning, deep learning models can have thousands to millions of trainable parameters, which allow these models to uncover complex and non-linear patterns within the data in an end-to-end manner for improved analysis, specifically in the context of biology. In addition, deep learning models have a flexible architecture, which can be easily adjusted or assembled to adapt to solving different problems. Early evidence has demonstrated tremendous ability of deep learning in identifying underlying and informative patterns from scRNA-seq data, accounting for the heterogeneity presented between scRNA-seq experiments, and noise and sparsity associated with scRNA-seq [10–12].

This review focuses on the use of deep learning in advancing key steps in the scRNA-seq data analysis. Extending on previous work [10–12], this review provides a comprehensive survey of deep learning in scRNA-seq data analysis. This review first provides an overview of deep learning, then introduces the most comprehensive list of deep learning models that have been used for various aspects of scRNA-seq data analysis, and finally, discusses limitations of these approaches and potential future directions in the field for improved scRNA-seq data analysis.

To narrow the scope of the paper, some aspects of scRNA-seq analysis have been excluded. Firstly, any discussion about sequencing read quality checks, read alignment, or quality checks for the alignment have been excluded, as deep learning is not involved in these procedures. Secondly, there is no discussion of RNA velocity-based downstream analyses, which involve identifying developmental transitions between cell types, including approaches such as DeepCycle [13] and VeloAE [14]. Since the input to the RNA velocity differs from that of standard scRNA-seq data analysis, which requires splicing information, this topic has been excluded. In addition, techniques such as Cobolt [15], scMM [16], and Schema [17], that combine information from multiple types of single-cell omics data have been excluded; this is to avoid providing extensive background on all different types of sequencing and antibody-based signal recognition approaches. Finally, studies that focused on simulating scRNA-seq data using deep learning, such as ESCO [18] and ACTIVA [19], are also excluded as they are not strictly necessary for scRNA-seq data analysis. More details of article inclusion and exclusion criteria can be found in Figure S1.

Deep learning architecture in scRNA-seq data analysis

To differentiate machine learning from deep learning, we can refer to deep learning as the use of deep neural networks (DNNs) where "deep" describes the multilayer network structure. A deep feed-forward neural network (DFNN) is the most basic deep architecture by simply stacking layers of "neurons" (**Figure 2**A). An artificial neuron is the basic computational unit of the DNNs, which takes the weighted summation of all inputs and feeds the result to a non-linear activation function, such as sigmoid, rectifier [*i.e.*, rectified linear unit (ReLU)], and hyperbolic tangent (Figure 2B), inspired by how human neurons work. A layer consists of a set of neurons and a DNN is built by stacking layers (Figure 2A). In the basic design, a neuron receives information from all neurons of the previous layer with trainable weights while sending its output to the successor layer. Mimicking information flow in a human brain, the input information (*i.e.*, gene expression profiles of the cells in scRNA-seq) flows from the input layer through the hidden layers and then the model generates an output at the last layer, *i.e.*, the output layer. The large set of trainable weights of the neurons and the non-linear transformations enable the DNNs to capture underlying complex patterns of the data. Training of a DNN is the procedure of determination of these trainable weights that optimize model performance. In deep learning, the model training is typically done based on backpropagation, which mathematically transmits model prediction error in the reverse order of information flow from the output layer to the input layer to update model parameters or weights [20].

Based on the task of interest and the manner of model training, machine learning, and subsequently deep learning,

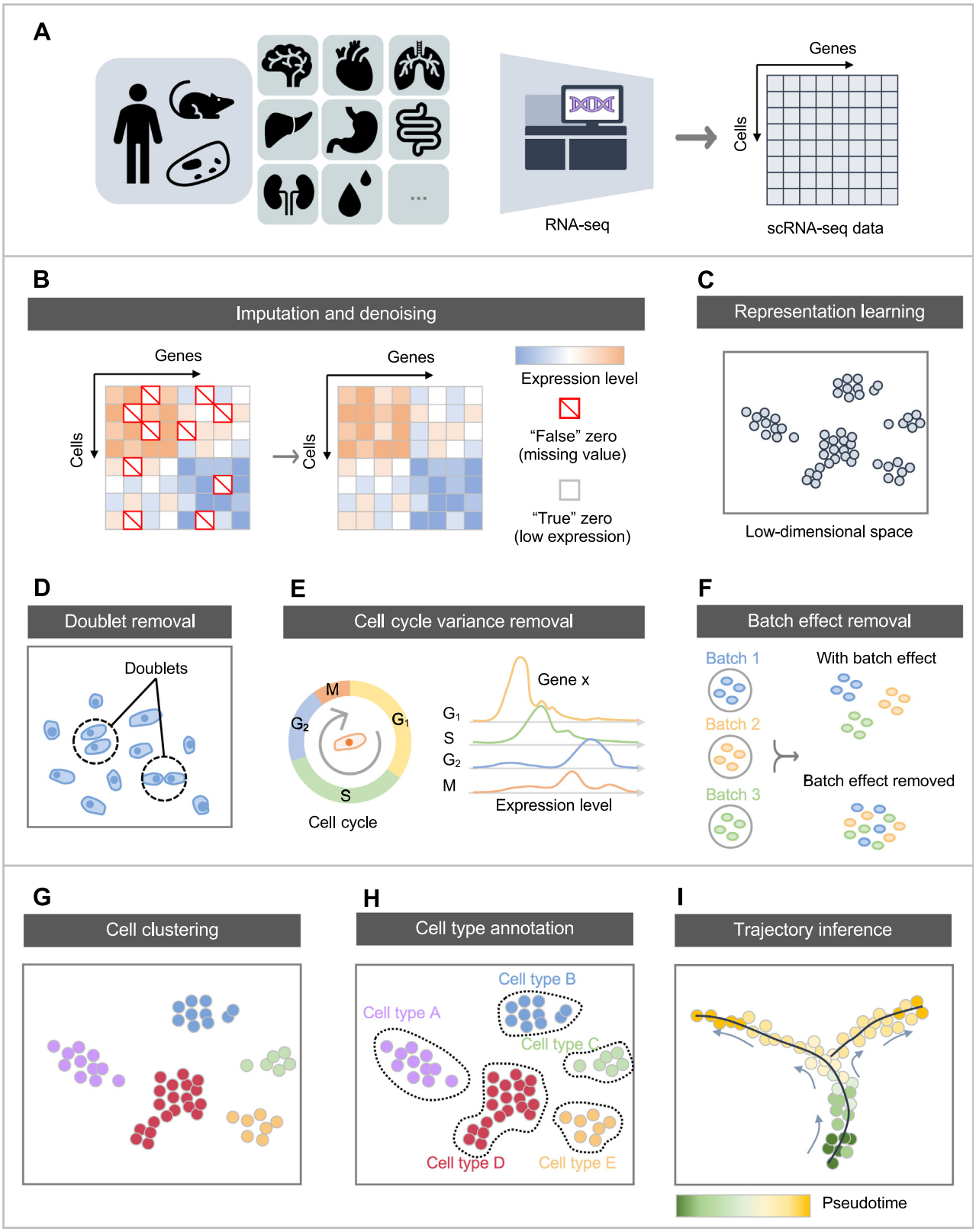

Figure 1 Schematic of the common pipeline in scRNA-seq analysis

A. scRNA-seq data collection. **B.** scRNA-seq data preprocessing: imputation and denoising. **C.** scRNA-seq data preprocessing: representation learning for dimensionality reduction. **D.** scRNA-seq data preprocessing: doublet removal. **E.** scRNA-seq data preprocessing: cell cycle variance removal. **F.** scRNA-seq data preprocessing: batch effect removal. **G.** Downstream analysis of scRNA-seq data: cell clustering. **H.** Downstream analysis of scRNA-seq data: cell type annotation. **I.** Downstream analysis of scRNA-seq data: trajectory inference. scRNA-seq, single-cell RNA sequencing; M, mitotic phase, *i.e.*, nuclear division of the cell (including prophase, metaphase, anaphase, and telophase); S, synthesis phase for the replication of the chromosomes (belonging to interphase); G1, gap 1 phase, representing the beginning of interphase; G2, gap 2 phase, representing the end of interphase, prior to entering the mitotic phase.

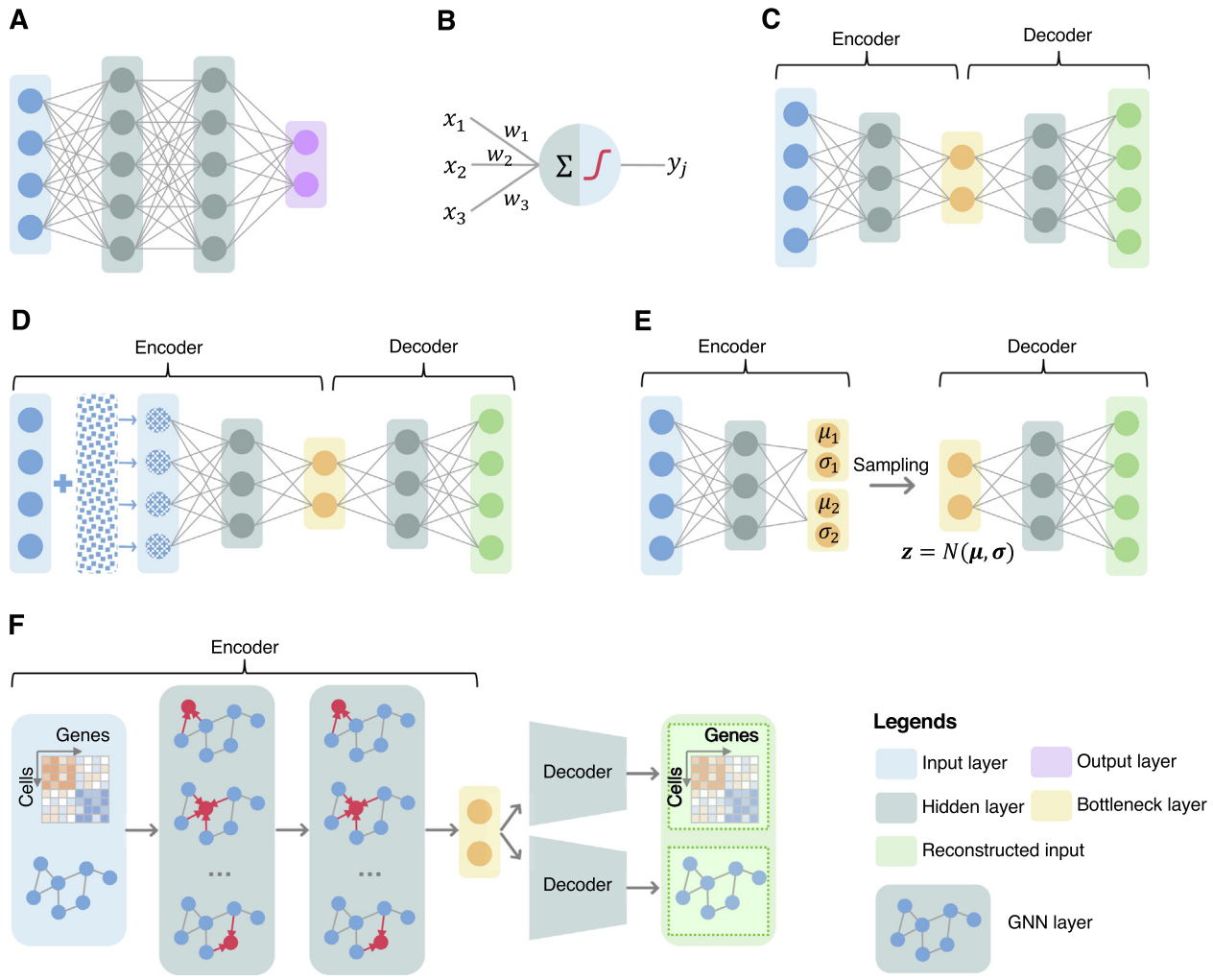

Figure 2 Illustration of deep learning architectures that have been used in scRNA-seq analysis

A. Basic design of a feed-forward neural network. **B.** A neural network is composed of "neurons" organized into layers. Each neuron combines a set of weights from the prior layer, and passes the weighted summed value through a non-linear activation function, such as sigmoid, rectifier (*i.e.*, ReLU), and hyperbolic tangent, to produce a transformed output. **C.** Autoencoder, a special variant of the feed-forward neural network aiming at learning low-dimensional representations of data while preserving data information. **D.** DAE, a variant of autoencoder, which was developed to address overfitting problems of autoencoders. DAE forces the input data to be partially corrupted and tries to reconstruct the raw un-corrupted data. **E.** VAE, a variant of autoencoder, aiming at compressing input data into a constrained multivariate latent distribution space in the encoder, which is regular enough and can be used to generate new content in the decoder. **F.** GAE. Benefiting from the advanced deep learning architecture GNN, GAE has been developed and used in scRNA-seq analysis. The encoder of GAE considers both sample features (*e.g.*, the gene expression profiles/counts of cells) and samples' neighborhood information (*e.g.*, topological structure of cellular interaction network) to produce low-dimensional representations while preserving topology in data. The decoder unpacks the low-dimensional representations to reconstruct the input network structure and/or sample features. ReLU, rectified linear unit; DAE, denoising autoencoder; VAE, variational autoencoder; GAE, graph autoencoder; GNN, graph neural network.

can be grouped into three main categories: supervised learning, unsupervised learning, and semi-supervised learning. The standard DFNN is an architecture mainly used for supervised learning (Figure 1A). In this scenario, the information available consists of a set of training data and the labels associated with each observation within the training set. The goal is to map the input data to a representation that can be used for tasks such as classification (for categorical labels) or regression (continuous labels). Semi-supervised learning, works when few data points have labels, using the limited labels to help inform the representation and label of the unlabeled data. Several scRNA-seq studies in this review use such technique, although it is not frequent.

There are several deep learning architectures suitable for unsupervised learning, which model data without any supervision, focus on identifying underlying patterns from the data, and are widely used in scRNA-seq data analysis, such as scRNA-seq data dimensionality reduction and cell clustering. The deep autoencoder (or autoencoder for simplicity) is a variant of the DFNN for unsupervised learning, which aims at

learning compact representations of data while attempting to maximally preserve input data information (*e.g.*, raw input gene expression in scRNA-seq) [21,22]. A autoencoder typically consists of two components: an encoder and a decoder (Figure 2C). The encoder is a DFNN that compacts data into a low-dimensional feature space at the so-called bottleneck layer. Then the decoder, with a mirror structure of the encoder, reconstructs the data in the original space from the low-dimensional representations derived by the encoder. Parameters of the autoencoder can be learned through minimizing such reconstruction errors using backpropagation. The learned low-dimensional representations of samples (*i.e.*, cells in scRNA-seq data) are also called embeddings. Compared to those non-deep learning models like principal component analysis (PCA) that are components of well-established scRNA-seq data analysis software like Seurat [2], an autoencoder is capable of finding a non-linear manifold where the data lie [20].

To overcome pitfalls of autoencoders like overfitting, several modifications to the autoencoder structure have been proposed that contain specific benefits for scRNA-seq data (Figure 2D–F). For instance, the denoising autoencoder (DAE) corrupts the input data slightly, by adding noise to a certain percentage of inputs, and then tries to rebuild the original input (Figure 2D). In this way, model robustness in overcoming data noise is enhanced, and hence quality of the low-dimensional representation of samples (*i.e.*, cells) learned from the scRNA-seq data [23,24] is improved. This can be added on top of standard regularization strategies such as L1 and L2 regularizations of model weights.

Variational autoencoders (VAEs) are a type of generative model, as opposed to a discriminative model like the standard autoencoder. A VAE learns a latent representation distribution (such as Gaussian distribution), instead of a specific vector, which can be used to generate examples of the latent representations of cells (Figure 2E). Compared to the standard autoencoders, VAEs allow for reduced dimensionality, but also the quantification of uncertainty of the latent representation [25]. In addition, VAEs allow for a smoother latent representation of the data, which is beneficial when trying to understand relationships between cells at lower dimensions. For example, the smoothed low-dimensional representations can help improve accuracy in measuring distance between cells, when using metrics like Euclidean distance. The variational component of the optimization process acts as a regularization term for the autoencoder to improve generalizability to other data sources [26]. Typically, training of a VAE is based on the loss function composed of the reconstruction error (such as mean-squared error) and the Kullback–Leibler (KL) divergence between the latent distribution and an assumed prior distribution. In this context, VAEs can suffer from KL vanishing, or loss of informativeness for the latent representation (latent space exactly matches prior distribution). Modifications, such as the β-VAE and other variations on it [27], have been developed to address these issues and adapted for single-cell analysis. In addition, depending on the value of β, these models also have been shown to improve the disentanglement, or the independence of the latent dimensions, which can advance scRNA-seq data analysis. In addition, by involving an adversarial loss function, popularized by generative adversarial networks (GANs) [28] that have been proven to be useful in synthetic data generation in other contexts, the VAEs can be described as an adversarial autoencoder [29].

Graph neural networks (GNNs) have successfully been applied to graph or network structured data analysis [30]. Typically, in each GNN layer, each node aggregates information from its local neighbors in the graph to update its representation (Figure 2F). The graph autoencoder (GAE) is a novel modification of autoencoders by using GNN layers (Figure 2F). In scRNA-seq data analysis, a cellular graph is usually built from the k-nearest neighbor (KNN) or shared nearest neighbor (SNN) strategies based on the gene expression profiles of cells [2]. In this context, the GAE can be used to learn cell (*i.e.*, node in the cellular graph) representations by incorporating cellular graph structure to decrease the noise of an individual cell. Figure 2F illustrates an example of GAE architecture for scRNA-seq analysis. Specifically, the encoder takes as input both gene expression read count matrix and cellular graph to generate cell representations, whereas the decoder(s) reconstructs the cellular graph structure (or both cellular graph structure and gene expression profile). There have also been more recent graph structures, using known protein–protein interaction (PPI) and cell–gene graphs, as prior knowledge, to improve scRNA-seq data analysis [31].

Applications of deep learning in scRNA-seq data analysis

This section describes how deep learning is currently being used to improve key steps in scRNA-seq data analysis (**Table 1**).

scRNA-seq data imputation and denoising

An intrinsic pitfall of scRNA-seq is that as little as 6%–30% of all transcripts are captured, based on the version of the chemistry used during sequencing and limited sequencing depth per cell [32]. Therefore, stochastically, cells will have what is known as "dropout" or the loss of all transcripts for a given gene [33], which is not biologically meaningful or accurate. From the data perspective, zero expression levels can be observed in the single-cell gene expression matrix; however, some of them are "true" zeros, indicating the lack of expression of genes in specific cells, while unfortunately some others could be "false" zeros observed from genes that are expressed, *i.e.*, dropout events, due to the low RNA capture rate (Figure 1B). Therefore, when imputing missing values in scRNA-seq data, one must distinguish the "true" zeros and "false" zeros (Figure 1B). This makes scRNA-seq data imputation more difficult than that of other biomedical data (such as clinical data), where missing values can be identified easily. Hence people also refer to the imputation procedure as scRNA-seq data denoising. It is important to note that denoising is not used in all deep learning-based approaches and therefore can be considered a potential component of the model, and benchmarking studies should be performed to see if it provides substantial benefits.

To account for the issue, conventional approaches [34–36] were proposed mainly focusing on imputing missing values based on correlated or similar genes or cells. However, they are usually computationally intensive and limited in capturing non-linearity in scRNA-seq data. To better address this issue, deep learning approaches have been developed for scRNA-seq data imputation and denoising [37–49]. Based on an idea similar to regression imputation [50], *i.e.*, predicting missing values

Table 1 A summary of the selected studies in this review

Category	Model name	Model type	Code availability	Technical advancement	Year	Ref.
Imputation and denoising	DeepImpute	AE	https://github.com/lanagarmire/deepimpute (Python)	Using correlated genes to impute missing values using AE	2019	[37]
	scIGAN	GAN	https://github.com/xuyungang/scIGANs	Using KNN of a set of boundary equilibrium GAN-generated cells for a certain cell type to perform imputation	2020	[38]
	scGMAI	AE	https://github.com/QUST-AIBBDRC/scGMAI	Using output of AE with Softplus activation functions as imputed representation for further dimensionality reduction with FastICA and clustering with GMM	2021	[39]
	SAVER-X	AE	https://github.com/jingshuw/SAVERX	Using novel empirical Bayesian shrinkage approach to predicting imputed values from autoencoder output based on gene–gene relationships	2019	[40]
	DCA	AE	https://github.com/theislab/dca	Using zero-inflated negative binomial loss for denoising	2019	[41]
	ZINBAE	AE	https://github.com/ttgump/ZINBAE	Using a Gumbel SoftMax applied to dropout matrix of decoder output and zero-inflated negative binomial loss for denoised data representation	2021	[42]
	scSDAE	DAE	https://github.com/klovbe/scSDAE	Stacked DAE with L1 penalty only for values with 0 to induce sparsity into output	2020	[43]
	GraphSCI	AE/GAE	https://github.com/biomed-AI/GraphSCI	Using gene–gene network derived from a thresholded Pearson correlation calculation for improved imputation	2021	[44]
	SAVERCAT	VAE	-	Using highly variable genes to train conditional VAE, then use the learned parameters to denoise retrain the decoder using the entire set of genes for downstream analysis	2020 (preprint)	[45]
	SEDIM	AE/DFNN	https://github.com/li-shaochuan/SEDIM	Using learning algorithm to find optimal hyperparameters for model generation to perform imputation	2021	[46]
	AdImpute	AE	-	Using MSE on AE output and imputed values from DrImpute in addition to standard autoencoder training	2021	[47]
	GNNImpute	GAE	https://github.com/Lav-i/GNNImpute	Using GAE to perform imputation	2021	[48]
	scGAIN	GAN	https://github.com/mgunady/scGAIN	Concatenating mask of dropout values and original count matrix with randomly initialized values, using hint generator to perturb original mask, and using adversarial training to predict which values in imputed cell representation are real or fake	2019	[38]
	LATE/TRANSLATE	AE	https://github.com/audreyqyfu/LATE	Using AE with MSE for non-zero input values and transfer or learned weights to other datasets	2020	[49]
Doublet removal	Solo	VAE	https://github.com/calico/solo	Using scVI model for dimensionality reduction with for doublet *vs.* singlet embedding and neural network for classification of doublets	2020	[54]
Cell cycle variance removal	Cyclum	AE	https://github.com/KChen-lab/Cyclum	Using circular activation functions in decoder to identify circular latent structures and subsequently cell cycle structure	2020	[58]
Dimensionality reduction	scScope	AE	https://github.com/AltschulerWu-Lab/scScope	Introducing the autoencoder output recurrently to impute missing values and improve latent representation	2019	[59]
	VASC	VAE	https://github.com/wang-research/VASC	Modeling the data as zero-inflated (Gumbel distribution) in decoder using VAE	2018	[60]
	net-SNE	DFNN	https://github.com/hhcho/netsne	Applying t-SNE loss function to neural network	2018	[61]
	scVI	VAE	https://github.com/YosefLab/scvi-tools	Using cell specific scaling of counts based on size factor for cell that is modeled into VAE	2018	[55]
	scDHA	AE/VAE	https://github.com/duct317/scDHA	Using non-negative weights for non-negative kernel autoencoder for feature selection and multiple decoders in VAE for the stacked Bayesian autoencoder for feature representation	2021	[62]

(*continued on next page*)

Table 1 (*continued*)

Category	Model name	Model type	Code availability	Technical advancement	Year	Ref.
	scGSLC	GCN	https://github.com/sharpwei/GCN_sc_cluster	Using protein–protein interaction network to perform dimensionality reduction for improved clustering	2021	[31]
	scVAE	VAE	https://github.com/scvae/scvae	Using a Gaussian mixture prior for the VAE training	2020	[63]
	scPhere	VAE	https://github.com/klarman-cell-observatory/scPhere	Using spherical or hyperbolic embedding to improve clustering and latent representation of single cells	2021	[64]
	DiffVAE/GraphVAE	VAE	https://github.com/ioanabica/DiffVAE	Using VAE and GraphVAE framework for scRNA-seq analysis with InfoVAE model	2020	[65]
	MMD-VAE	VAE	https://mmd-vae.hi-it.org/	Replacing Kullback–Leibler divergence term with MMD for VAE training	2019 (preprint)	[66]
	DR-A	AAE	https://github.com/eugenelin1/DRA	Using adversarial loss on reconstructed output and latent space of the variational autoencoder	2020	[67]
	scRAE	AAE	https://github.com/arnabkmondal/scRAE	Using a neural network to reduce the bias of the regularization term for the AE latent representation in VAE or AAE framework	2021	[68]
		VAE/β-VAE	-	Using β-VAE for disentangled representation of single cells generating more interpretable latent representations	2020	[69]
	scGAE	GAE	https://github.com/ZixiangLuo1161/scGAE	Using GAE for dimensionality reduction	2021	[70]
	SCA	AE	https://github.com/kendomaniac/SCAtutorial	Using known relationships of genes with transcription factors, kinases, and miRNA to model network connections for autoencoder	2021	[71]
	GOAE	AE	-	Using prior knowledge gene ontology terms to impact the connection between layers for the autoencoder	2019	[72]
	DeepAE	AE	https://github.com/sourcescodes/DeepAE	Using weights from neural network to generate gene ontology terms for hidden representation dimensions	2020	[73]
	pmVAE	VAE	https://github.com/ratschlab/pmvae	Using ensemble of VAEs each with a pathway specific set of genes for more interpretable single-cell representation	2021 (preprint)	[74]
	VEGA	VAE	https://github.com/LucasESBS/vega-reproducibility	Using mask on linear decoder weights to improve interpretation based on gene database	2021	[75]
	Interpretable Autoencoder	AE	https://github.com/theislab/intercode	Using pathway databases, such as MSigDB, to induce regularization into model for improved interpretability	2020 (preprint)	[76]
	LDVAE	VAE	https://github.com/YosefLab/scvi-tools	Restricting decoder of scVI to linear layer for improved interpretability	2020	[77]
	SCDRHA	GAE	https://github.com/WHY-17/SCDRHA	Using output of DCA as input for graph attention autoencoder	2021	[78]
	scCDG	DAE/GAE	https://github.com/WHY-17/scCDG	Using GAE on latent representation from DAE	2021	[79]
	CellVGAE	GAE	https://github.com/davidbuterez/CellVGAE	Using variational graph attention autoencoder for dimensionality reduction	2022	[80]
	graph-sc	GAE	https://github.com/ciortanmadalina/graph-sc	Inputting cell–gene graph into GAE for dimensionality reduction	2021	[81]
	contrastive-sc	DFNN	https://github.com/ciortanmadalina/contrastive-sc	Using SimCLR loss based on two different dropout representations of the same cell for self-supervised contrastive learning	2021	[82]
	resVAE	VAE	https://github.com/lab-conrad/resVAE	Masking out latent representation based on known cell type or other meta data	2020	[83]
	HD Spot	AE	-	Using genetic algorithm to optimize AE hyperparameters and converting the encoder to a classifier to perform SHAP for improved interpretability of gene importance for different classes	2020	[84]
	KPNN	DFNN	https://github.com/epigen/KPNN	Controlling node connections in neural network based on known biological pathways	2020	[85]

(*continued on next page*)

Table 1 (*continued*)

Category	Model name	Model type	Code availability	Technical advancement	Year	Ref.
	SSCA/SSCVA	AE/VAE	-	Using known gene sets to control node connections in autoencoder	2019	[86]
	MichiGAN	VAE/GAN	https://github.com/welch-lab/MichiGAN	Using β-TCVAE for disentangled representation of single cells generating more interpretable latent representations	2021	[87]
Batch effect removal	SMILE	DFNN	https://github.com/rpmccordlab/SMILE	Using contrastive learning loss, *i.e.*, NCE for the integration of multiple datasets	2021	[95]
	DAVAE	VAE	https://github.com/jhu99/davae_paper	Using gradient reversal layer for adversarial training to perform data integration	2021	[96]
	SCALEX	VAE	https://github.com/jsxlei/SCALEX	Using decoder-based domain-specific batch normalization for multi-source data integration	2021	[97]
	AD-AE	AE	https://gitlab.cs.washington.edu/abdincer/ad-ae	Using adversarial training of AE for multiple different confounders including age and batch to learn de-confounded latent representation	2020	[98]
	scGAN	VAE	https://github.com/li-lab-mcgill/singlecell-deepfeature	Using adversarial training of VAE with categorical (batch) or continuous (age) variables for data integration	2021	[99]
	iMAP	AE/GAN	https://github.com/Svvord/iMAP	Using two step integration including (1) content loss and (2) random walk MNN-based GAN model	2021	[100]
	BERMUDA	AE	https://github.com/txWang/BERMUDA	Using MetaNeighbor with MMD regularization for the integration of cluster pairs between batches identified	2019	[101]
	trVAE	VAE	https://github.com/theislab/trVAE	Using conditional VAE with MMD regularization in latent space	2020	[102]
	scDGN	DFNN	https://github.com/SongweiGe/scDGN	Using semi-supervised learning with domain adaptation using gradient reversal layer	2021	[103]
	scETM	VAE	https://github.com/hui2000ji/scETM	Using interpretable decoder based on matrix tri-factorization (topic modeling)	2021	[104]
	-	BERT Transformer	-	Using transformers for encoder and decoder	2021	[105]
	deepMNN	DFNN	https://github.com/zoubin-ai/deepMNN	Using residual network to perform batch correction on predetermined MNN pairs of cells using highly variable genes	2020	[106]
	HDMC	AE	https://github.com/zhanglabNKU/HDMC	Using contrastive loss with MetaNeighbor-identified similar clusters between batches for improved batch correction	2021	[107]
	CBA	AE	https://github.com/GEOBIOywb/CBA	Integrating pre-defined matching cell clusters from two domains using a two-stream AE network, which uses concatenation of latent representation within and between streams	2021	[108]
Cell clustering	scAIDE	AE/DFNN	https://github.com/tinglabs/scAIDE	Using MDS encoder for improved AE dimensionality reduction and K-means for improved clustering of different sized clusters	2020	[113]
	scDMFK	AE	https://github.com/xuebaliang/scDMFK	Using simultaneous dimensionality reduction and clustering with an adaptive fuzzy K-means loss function	2020	[114]
	scCCESS	AE	https://github.com/gedcom/scCCESS	Consensus clustering of latent representation clustering from ensemble of random projection or random subset of gene AE	2019	[115]
	DESC	AE	https://github.com/eleozzr/desc	Pretraining stacked AE, then performing simultaneous clustering and dimensionality reduction using deep embedding clustering	2020	[116]
	CarDEC	AE	https://github.com/jlakkis/CarDEC	Separate encoder for high and low expressing genes with separate loss functions to improve single-cell representation	2021	[117]
	scziDesk	AE	https://github.com/xuebaliang/scziDesk	Using weighted soft K-means clustering of latent space during AE training	2020	[118]
	scGNN	AE/GAE	https://github.com/juexinwang/scGNN	Using a combination of several AE structures, including a graph autoencoder to perform entire pipeline of single-cell analysis after pre-processing	2021	[119]

(*continued on next page*)

Table 1 (*continued*)

Category	Model name	Model type	Code availability	Technical advancement	Year	Ref.
	DUSC	DAE	https://github.com/KorkinLab/DUSC	Using DAE for dimensionality reduction	2020	[23]
	GraphSCC	GCN/DAE	https://github.com/GeniusYx/GraphSCC	Jointing residual GCN and DAE with simultaneous clustering for improved latent representation and clustering	2021	[24]
	SAUCIE	AE	https://github.com/KrishnaswamyLab/SAUCIE	Using information dimension regularization and cluster distance regularization for improved clustering	2019	[120]
	EMDEC	AE	-	Using optimization procedure for hyperparameters and architecture for deep embedded clustering with scRNA-seq data	2021	[121]
	MoE-Sim-VAE	VAE	https://github.com/andkopf/MoESimVAE	Using mixture of Gaussians prior for VAE, define separate decoders for each Gaussian for reconstruction, and using similarity + DEPICT loss function for clustering	2020	[122]
	scvis	VAE	https://bitbucket.org/jerry00/scvis-dev	Using probabilistic generative model with asymmetric t-SNE objective for improved clustering with dimensionality reduction	2018	[25]
Cell type annotation	scAnCluster	AE	https://github.com/xuebaliang/scAnCluster	Inclusion of soft K-means clustering with entropy regularization and a self-supervised cell similarity loss for improved clustering	2020	[126]
	JIND	DFNN	https://github.com/mohit1997/JIND	Using adversarial training to match latent representation coming from source and target domains for downstream cell annotation	2022	[127]
	ItClust	DAE	https://github.com/jianhuupenn/ItClust	Pretraining model on source dataset and then finetuning on target dataset	2020	[128]
	scDeepSort	GAE	https://github.com/ZJUFanLab/scDeepSort	Using graph neural network on cell-gene graph to predict pre-defined cell types	2021	[129]
	AutoClass	AE	https://github.com/datapplab/AutoClass	Pseudo-labels from K-means clustering or known cell types during training to improve AE-based imputation	2022	[130]
	scANVI	VAE	https://github.com/YosefLab/scvi-tools	Developing a semi-supervised extension of scVI	2021	[131]
	scSemiCluster	AE	https://github.com/xuebaliang/scSemiCluster	Using cluster compactness loss for labeled data to improve transfer learning	2020	[132]
	scAdapt	GAN	https://github.com/zhoux85/scAdapt	Using virtual adversarial training loss and semantic alignment loss to improve training in a semi-supervised setting	2021	[133]
	scArches	VAE	https://github.com/theislab/scarches	Concatenation of new dataset to pretrained AE (“architectural surgery”) for improved mapping of query to reference dataset	2021	[134]
	MARS	AE	https://github.com/snap-stanford/mars	Using the meta-learning approach to allow for identification of new clusters during transfer learning in new datasets	2020	[135]
	MAT^2	AE	https://github.com/Zhang-Jinglong/MAT2	Generating triplets using either known cell labels, or pseudo-labels based on Seurat for contrastive learning using triplet loss and use triplet loss in batch correction	2021	[136]
	scNym	DFNN	https://github.com/calico/scnym	Using MixMatch for semi-supervised learning	2021	[137]
	scGCN	GCN	https://github.com/QSong-github/scGCN	Development of multiple mutual nearest neighbor graphs based on CCA using reference and query datasets for transfer learning	2021	[138]
	scMRA	AE | GCN	https://github.com/ddb-qiwang/scMRA-torch	Development of cell type prototype knowledge graph based on multiple different source domains for improved transfer learning to unlabeled dataset	2021	[139]
	MapCell	DFNN	https://github.com/lianchye/mapcell	Using Siamese network with contrastive loss for pairs of cells identified as the same type. Using learned distance metric for label transfer and new cell discovery	2021	[140]
	sigGCN	GAE/DFNN	https://github.com/NabaviLab/sigGCN	Concatenating latent representation learned from FFNN and GAE to predict cell type	2021	[141]

(*continued on next page*)

Table 1 (*continued*)

Category	Model name	Model type	Code availability	Technical advancement	Year	Ref.
	scIAE	AE	https://github.com/JGuan-lab/scIAE	Using ensemble of autoencoders with random projections to perform dimensionality reduction. Using the learned representations to train downstream classifiers for new data	2021	[142]
	mtSC	DFNN	https://github.com/bm2-lab/mtSC	Using N-pair loss for deep metric learning across all reference datasets separately for trained model and using a consensus score from each reference dataset for cell annotation of query cell	2021	[143]
	ImmClassifier	DFNN	https://github.com/xliu-uth/ImmClassifier	Using probability of coarse cell predictions into fine-grain predictions using the coarse grain probability distribution as input of a DFNN	2021	[144]
	netAE	VAE	https://github.com/LeoZDong/netAE	Introduction of cell classification on latent representation for labeled cells and modularity loss based on cell–cell similarity matrix of latent representation	2021	[145]
	Cell BLAST	VAE	https://github.com/gao-lab/Cell_BLAST	Using of improved distance-metric for mapping query cell to reference latent-representation and includes Poisson distribution as method for data augmentation of input scRNA-seq data	2020	[147]
	MultiCapsNet	CapsNet [187]	https://github.com/bojone/Capsule	Using CapsNet for scRNA-seq data analysis	2021	[146]
Trajectory analysis	VITAE	VAE	https://github.com/jaydu1/VITAE	Using hierarchical mixture model based on latent representation from VAE to predict cell pseudotime	2020	[150]
Complete analysis framework	scAEspy	-	https://gitlab.com/cvejic-group/scaespy	Single-cell analysis package containing several different AE architectures for analysis	2021	[185]
	sfaira	-	https://github.com/theislab/sfaira	Single-cell package containing pipeline and pretrained models	2021	[186]

Note: AAE, adversarial autoencoder; AE, autoencoder; CapsNet, capsule neural network; CCA, canonical correlation analysis; DAE, denoising autoencoder; DCA, deep count autoencoder; DFNN, deep feed-forward neural network; FFNN, feed-forward neural network; GAN, generative adversarial networks; GAE, graph autoencoder; GCN, graph convolutional network; GMM, Gaussian mixture model; KNN, k-nearest neighbors; MDS, multidimensional scaling; MNN, maximum mean discrepancy; MSE, mean squared error; NCE, noise-contrastive estimation; TCVAE, total correlation variational autoencoder; t-SNE, t-distributed stochastic neighbor embedding; VAE, variational autoencoder; MMD, maximum mean discrepancy; SHAP, SHapley Additive exPlanations.

of target features (genes) using other features as predictors, DeepImpute (DNN imputation) [37] has been shown to be an effective approach for scRNA-seq data imputation using deep learning. Since DeepImpute only focuses on a subset of genes to impute (default 512), it can take advantage of the strength of the DNN but also reduces model parameters to make itself efficient and scalable. scIGAN (GAN for single-cell imputation) [38] leveraged a novel deep learning model, GAN. Specifically, scIGAN generates cells to impute dropout events, instead of using observed cells.

Other efforts that aimed at solving the scRNA-seq data imputation task use autoencoders. Intuitively, the reconstructed values by an autoencoder can be used to fill missing values in the original single-cell gene expression data. Based on such idea, a recent scRNA-seq analysis pipeline, scGMAI [39], has used an autoencoder for data imputation. Their experimental results on seventeen public scRNA-seq datasets demonstrated improvements of the autoencoder-based imputation in cell clustering task. SAVER-X [40] also used a standard autoencoder to denoise data. What makes SAVER-X unique is that the autoencoder was used to model the portion of expression of each gene that is predictable by other genes. Another innovation of SAVER-X is the incorporation of transfer learning framework. Particularly, the autoencoder can be pretrained using public cross-species (human and mouse) datasets, making it capable to transfer knowledge learned from mouse data to improve human data analysis.

In addition, some other studies combined the autoencoder architecture with parametric functions to facilitate imputation. Deep count autoencoder (DCA) [41] used the zero-inflated negative binomial distribution (ZINB) noise model, which is effective at characterizing discrete, overdispersed, and highly sparse count data, into the autoencoder architecture. Instead of directly reconstructing input data, DCA can produce three gene-specific parameters of ZINB, including mean, dispersion, and dropout probability, at the last layer of the autoencoder. After model training, the mean matrix from the output of the decoder can be used as a "denoised" version or imputed version of the original count matrix for downstream analysis. Yet, ZINB has its inherent shortcomings. As allowing three parameters for describing each data point, ZINB may be over-permissive to give a too high degree of freedom which may make the results unstable. To overcome this, ZINB model-based autoencoder (ZINBAE) [42] developed a ZINB autoencoder by introducing a differentiable function [51] to approximate the categorical data and a regularization term to control the ZINB. Sparsity-penalized stacked denoising autoencoder (scSDAE) [43] leveraged a stacked DAE for scRNA-seq imputation with L1 loss to prevent overfitting. GraphSCI [44] combined the graph convolutional network (GCN), a type of GNN, with the standard autoencoder to model gene–gene co-expression relations and single-cell gene expression matrix, respectively. The incorporation of gene–gene co-expression relations as prior knowledge helps to alleviate bias in imputation and reduce impact of technical variations in sequencing.

It is worth noting that a notable benefit of deep learning in scRNA-seq data imputation and denoising is that there could be some non-linear relationships between certain genes. The deep architecture would allow for a more informed imputation strategy as compared to standard linear approaches. In addition, whether or not the ZINB model is appropriate has been debated [52]. Finally, additional information, such as mapping relationships between genes in a graph structure, has been used for improved imputation.

Doublet removal

The two main technologies used in single-cell isolation for downstream sequencing are microfluidic approaches, where cells are individually placed into oil droplets using microfluidic devices, and nanowell-based approaches, where tiny and patterned wells are created and individual cells are placed within each well [32,53]. Although these technologies have been improved and even commercialized over the past decade, errors can occur, in which more than one cell is captured within a droplet or well, *i.e.*, so-called a "doublet". This can lead to improper interpretation of gene expression for a particular cell as the expression is a combination of multiple, and possibly different types of cells. This can happen if cells are not completely disassociated from one another after collection of the biological specimen.

To address this, single-cell doublet detection techniques have been developed. Typically, a doublet detection technique can be broken down into 3 main stages: doublet simulation, cell representation learning, and classifier training [54]. Solo [54] is a single-cell doublet detection model that leveraged the deep learning technique. For stage one, *i.e.*, doublet simulation, Solo repeatedly took a random subset of cells (assumed to be singlets or single cells) and summed their UMIs, to generate N different simulated doublets. For stage two, an unsupervised scRNA-seq data representation learning is engaged to embed these cells, singlets, and simulated doublets into a low-dimensional space. Specifically, Solo used the VAE-based representation learning model, single-cell variational inference (scVI) [55], to achieve the informative and robust cell representations. For stage three, Solo removed the decoder region and froze the weights for the encoder region. A set of fully connected layers were added to the end of the encoder, and then the model was trained to distinguish "singlet" and "doublet". Interestingly, scVI accounted for sequencing depth, which the authors state was a critical feature to include when running their model.

Traditional machine learning approaches for doublet detection, including Scrublet [56] and DoubletFinder [57], differ in the representation learning approach (usually PCA), as compared to a VAE, and in the way the authors identify doublets, relying on nearest neighbor approaches, compared to a neural network used in Solo. Interestingly, for Solo, the authors tested using both a VAE with KNN classifier and PCA with a neural network classifier, both of which performed worse in identifying doublets. This may highlight the need for both non-linear dimensionality reduction, to model the non-linear relationship between combinations of cells, and the need for a non-linear classifier, as the latent space can still have non-linear relationships between singlets and doublets.

Cell cycle variance annotation

Gene expression can change as the cell moves along its normal cell cycle. The frequency by which cell types move between phases of the cell cycle varies due to many different factors [58], and can impact the expression of certain genes as a

function of cycle. This change may add additional noise to downstream gene expression analysis and such uninformative variation between cells should be removed, or these changes may be useful information for downstream interpretation of sequencing data. Typically, Seurat [2] used a cell scoring package, which can be used to regress out or subtract out the influence of cell cycle in the PCA latent space or explain variation among cells based on cell stage. Our literature search did find one study, Cyclum [58], which utilized the deep learning technique to account for cell cycle regression. Cyclum aimed at finding a non-linear periodic function that encodes the gene expression profiles of cells to low-dimensional space and are sensitive to circular trajectories. To this end, it used a modified asymmetric autoencoder, which was composed of a standard encoder for representation learning and a decoder that uses a combination of cosine and sine as activation functions in the first layer and followed by a second layer for linear transformations. As a direct comparison to other linear methods (such as PCA), Cyclum showed superior performance in all datasets, using Hoechst staining of cells to identify cell cycle as ground truth labels. The test sets have a somewhat homogeneous cell population, so benchmarking on other datasets, with several different cell types, may be interesting for identifying model performance, and improvement in subsequent downstream analyses.

scRNA-seq data representation learning for dimensionality reduction

scRNA-seq data typically contains genome wide expression profiles of cells and hence has a very high feature space, making data analysis challenging due to the curse of dimensionality. The emerging term, scRNA-seq data representation learning, refers to the process of learning meaningful (information preserved) and compressed (low-dimensional) representations of cells, or so-called embeddings, based on their gene expression profiles and has been an essential intermediate step in single-cell analysis. It can not only advance other scRNA-seq data preprocessing procedures, such as doublet detection and cell cycle variance annotation, but also benefit downstream analyses such as cell clustering, cell type annotation, and trajectory inference.

Early efforts in scRNA-seq data dimensionality reduction aimed at identifying a set of highly variable genes [2]. In addition, PCA, which aims at determining principal components that can largely describe variance of the original data, has also been widely used to reduce dimensionality of scRNA-seq data. Though PCA is used in well-established software like Seurat [2], it cannot capture non-linear patterns in data and hence may harbor limitations when it comes to accurately reflecting the nature of cells. Due to their intrinsic ability to learn underlying, meaningful, and non-linear patterns from raw data [20], deep learning approaches [31,55,59–87], especially the autoencoder and its extensions, have been effective techniques for scRNA-seq data representation learning and dimensionality reduction.

scScope [59] used an autoencoder to learn improved low-dimensional representations of scRNA-seq data while simultaneously addressing dropout events. To this end, scScope introduced an imputer layer to generate a corrected input data based on the output of the decoder and re-sent it back to the encoder to re-learn an updated latent representation in an end-to-end manner. VAEs, which have shown the ability to disentangle latent representations or improve independence of latent dimensions [88], have demonstrated notable achievements in scRNA-seq representation learning. VASC (VAE for scRNA-seq data) [60] is an early effort that used VAE architecture with a zero-inflated layer to account for dropout for scRNA-seq data dimensionality reduction. Compared to the traditional approaches, VASC resulted in better representations for very rare cell populations and performed well on data with more cells and higher dropout rate. scVI [55] also used a VAE for scRNA-seq representation learning. It aggregated information across similar cells and genes to approximate latent distribution of the raw expression data but also accounted for batch effects. Single-cell decomposition using hierarchical autoencoder (scDHA) [62] leveraged an autoencoder combined with an ensemble of VAEs for learning informative representations of cells while preventing overfitting.

In VAEs, the modification of the prior distribution can be used to enhance the learned latent representation. scVAE (VAE for single-cell data) [63] utilized a Gaussian mixture model to model the latent representation instead of a standard normal. The Gaussian mixture enables the model to learn robust representations but also discover latent cluster structure simultaneously. scPhere [64] used the von Mises–Fisher (vMF) distribution to project data points onto the surface of a unit hypersphere and tested model variants that use hyperbolic space as the latent embedding [89]. In this way, scPhere decreased the crowding of points associated with normal VAE training and improved temporal information of data. In addition, there are modifications to loss function of VAEs to improve the disentangled representation. The basic VAEs, which typically use a KL loss, may suffer from the issue of less informative representation, *i.e.*, the learned representations are insufficient to represent the original data [90]. To overcome such issue, the DiffVAE [65] and maximum mean discrepancy VAE (MMD-VAE) [66] utilized a MMD loss instead of the traditional KL loss in the VAEs. Dimensionality reduction with adversarial VAE (DR-A) [67] is a model that utilized a modified VAE, where the KL divergence component is replaced with two adversarial losses, one for latent representation and another for reconstruction. scRAE [68] builds upon this by modifying the adversarial autoencoder structure. Instead of sampling from a prior distribution and feeding that directly into the adversarial arm of the model, which is done in DR-A, the authors add a neural network after sampling from the prior distribution to be matched with the latent distribution generated from scRNA-seq autoencoder part of the model. The authors argue that this form of regularization allows for a reduction in the bias associated with an assumed normal distribution, such as in DR-A, and shows that this model outperforms several other approaches including DR-A. Kimmel [69] introduced a β-VAE to learn a disentangled representation of scRNA-seq data. Although the author did see improvements in some downstream analyses such as identifying different cell conditions from the representation, others, such as cell type clustering, had a decreased performance.

GAEs have also been used to model topology structure of relationships between cells in addition to the features (gene expression profiles) themselves, toward achieving better representations. Graph-DiffVAE [65] and single-cell GAE (scGAE) [70] are existing efforts in this context. Typically, they first

constructed a cell graph by connecting each cell to its KNNs based on gene expression profiles. Then it models and reconstructs the cell graph and the gene expression matrix to learn low-dimensional representations of cells.

Model interpretability is a concern in deep learning. For scRNA-seq data, a common way for interpretable deep representation learning has been the use of prior domain knowledge, *i.e.*, known relationships between molecules, like RNA and transcription factors, to modify standard neural networks. Sparsely connected autoencoder (SCA) [71] used various forms of the autoencoder where the connections are related to genes, transcription factors, miRNA targets, cancer-related immune signatures, and kinase specific protein targets. Additionally, other methods have leveraged similar known relationships, which allow for the construction of gene regulatory networks (GRNs). In the case of knowledge-primed neural networks (KPNNs) [85], the dimensionality reduction from the input, genes, to the output, phenotype, can be done by connecting nodes in one layer to the next that represent true relationships previously identified from large scale databases, such as the SIGnaling Network Open Resource (SIGNOR) [91] and Transcriptional Regulatory Relationships Unraveled by Sentence-based Text mining (TRUST) [92]. GRNs can be reconstructed by analyzing the node weights across layers. Similarly, methods have utilized other forms of data representation using specific gene–gene correlations, to generate GRNs using more complex deep learning models, such as convolutional and recurrent neural networks [93]. In these settings, the supervised learning model can be thought of as a feature extraction method, that reduces the input feature space to a lower dimensional representation that can be used to predict whether there are specific interactions between genes.

More general pathway information is also useful to generate a more interpretable deep learning model. Gene Ontology AutoEncoder (GOAE) [72] used Gene Ontology (GO) [94] to determine the connections within an autoencoder. DeepAE [73] used an autoencoder and weights associated with each hidden unit to identify GO terms that are associated with high weighted genes. Pathway module VAE (pmVAE) [74] encoded gene–pathway memberships for interpretable representation learning. Specifically, pmVAE contains a series of VAE subnetworks, each of which refers to a specific pathway module and only includes genes associated to this pathway. All pathway modules are combined to achieve global reconstruction of the input scRNA-seq data. VAE enhanced by gene annotations (VEGA) [75] performed a similar approach by masking genes such that genes within a certain gene module have similar contributions to a single latent dimension for the decoder. In addition, incorporating domain knowledge as a regularization term in the loss function to guide model training is another way to enhance interpretability. Rybakov et al. [76] injected GO into the loss function as a regularization term, such that genes associated with a certain pathway will be the only weights that contribute to the sum of a certain latent dimension. In LDVAE [77], the authors tried to improve interpretability of scVI by converting the decoder into a single linear layer, such that each gene can have a weight associated with each hidden unit in the latent space. Although interpretability increases, there can be a decrease in performance, as now models are built based on known relationships and there could be some unknown relationships that are not modeled due to gaps in biological knowledge.

Batch effect removal

Due to the stochastic nature of single-cell sequencing, experiments done at different times, in different locations, using different reagents, using different technologies, or using different technicians, may have specific biases associated with that experiment that may influence sequencing results. To combat this, deep learning models [95–108] have been developed to learn a shared latent representation for these different experiments, that removes technical noise but keeps biological variation.

A common way to address this task is based on domain adaptation, which usually relies on GANs, an advanced branch of deep learning. Typically, a latent representation is generated using the autoencoder or its extensions, and then an adversarial training step is used in a discriminator module outside of the autoencoder to reduce difference in latent representations between batches. Following such an idea, iMAP [100] is a well-designed batch effect removal framework based on a autoencoder and GAN. Specifically, it used an encoder to produce batch ignorant representation of cells and two generators to reconstruct the expression profile. Applied to tumor microenvironment datasets from two platforms, iMAP showed the capacity in taking advantage of powers of both platforms and identified novel cell–cell interactions using a non-deep learning approach, CellPhoneDB. In domain-adversarial and variational approximation (DAVAE) [96], a gradient reversal layer (GRL) was designed for domain adaptation to remove the batch effect. The single-cell domain generalization network (scDGN) framework [103] also used GRL. In contrast to other models, scDGN is trained in a supervised manner, aiming at maximizing the accuracy of cell type prediction while minimizing the differences between batches. Single-cell generative adversarial network (scGAN) [99] utilized a VAE architecture. The authors incorporated a discriminator module to predict batch from the data using an adversarial training. Adversarial deconfounding autoencoder (AD-AE) [98] aimed to learn a confounder-free representation of data. The authors performed an adversarial optimization by adding an adversarial arm to the model to predict "confounders", such as batch and age. By alternating training by freezing the adversary arm weights and optimizing the loss by minimizing the reconstruction loss and maximizing the confounder loss and then freezing the autoencoder weights and minimizing the confounder prediction, the authors "remove" confounder information from the latent space. Pang and Tegnér [105] used BERT Transformer [109], an advanced attention-based neural network, as the encoder and an adversarial GAN based approach for batch alignment. SCALEX [97] incorporated a domain-specific batch normalization layer in the decoder of the VAE model to account for technical variations based on batches.

In addition to adversarial based approaches, there are also methods based on distribution matching, such as methods using different regularization terms like MMD. Batch effect removal using deep autoencoders (BERMUDA) aimed to match the latent representations learned by autoencoders between two batches [101]. Specifically, the autoencoder was performed on two batches separately. To overcome batch effects, the autoencoder was trained by optimizing a loss containing two components: a standard reconstruction loss and an

MMD-based transfer loss between the latent representations of similar clusters from the two batches. Transfer VAE (trVAE) [102] targeted at matching distributions across conditions. In the case of two conditions, the authors feed one condition into the encoder with the appropriate conditions associated with it. Then for the decoder, the authors attach the opposite condition in the latent representation to transform the original condition feature matrix into the same space as the second condition. The MMD loss between the two conditions on the decoder region of the model was engaged to match distributions between different batches.

In addition, there are alternative ways to do batch correction. For example, the scScope pipeline [59] used a built-in batch correction layer in the DNN to performance batch correction. SMILE [95] utilized a contrastive learning framework [110], which forces each cell to be like itself plus a Gaussian noise while dissimilar to any other cells. Single-cell embedded topic model (scETM) [104] used topic modeling to account for different batches and allow for some correction associated with batch-specific differences between cells. Specifically, it contains an encoder to infer cell type mixture and a linear decoder based on matrix tri-factorization.

Cell clustering

One major goal of scRNA-seq analysis is to group the heterogeneous cell population into homogeneous sub-populations, such that cells within a sub-population are likely to have the same cell type or status. Clustering, an unsupervised learning approach, is a good fit to address this task. Typically, a clustering algorithm aims at identifying clusters, by minimizing dissimilarity within a given cluster while maximizing that between clusters. The well-established single-cell pipelines, such as Seurat [2] or Scanpy [3], use graph-based clustering methods such as Louvain [111] and Leiden [112] algorithms. Generally, they first build a cell–cell network using strategy like KNN based on gene expression profiles of cells, and then identified clusters by optimizing a measure such as ''modularity'' in Louvain [111], which measures cluster structure in the network (graph). In addition, the well-known K-means, which greedily adjusts clusters' centroids to optimize cluster structure, has also been widely used in scRNA-seq data analysis. Typically, the clustering algorithms take low-dimensional representations of cells as input, instead of raw gene expression profiles. In the deep learning setting [23–25,113–122], the two steps, representation learning and clustering, can be done sequentially or simultaneously.

For the sequential modeling approaches, deep learning-based representation learning was performed first and followed by the classical clustering algorithms performed on the learned low-dimensional representations. The single-cell autoencoder-imputation network with a distance-preserved embedding network (scAIDE) [113] first provided a hybrid deep architecture for representation learning. Specifically, an autoencoder is used for imputation of the original input matrix, meanwhile a multidimensional scaling (MDS) encoder was used for dimensionality reduction. After that, scAIDE proposed a variant of K-means, called RPH-Kmeans, which utilized the locality sensitive hashing (LSH) technique [123] to tackle the data imbalance for clusters problem (*i.e.*, different sized clusters) [113]. In addition, deep unsupervised single-cell clustering (DUSC) [23] made an extension to DAE for representation learning, *i.e.*, denoising autoencoder with neuronal approximator (DAWN), which enables the model to automatically determine the number of latent features that are sufficient to represent the original gene expression data efficiently. The learned low-dimensional representations were then used to identify clusters using an expectation–maximization (EM) algorithm [124]. scDMFK [114] also used DAE and combined with the fuzzy K-means algorithm to identify cell clusters. scCCESS [115] sampled the input data randomly to obtain multiple subsets. Then it learned low-dimensional representations in each subset using autoencoders and performed clustering subsequently. An ensemble clustering method was used to integrate clustering results in each subset to get the final one.

For the simultaneous modeling approaches, the models were designed in an end-to-end manner. Taking raw gene expression profiles as input, the data representation learning and clustering modules can be done automatically and these two modules can even improve each other in some advanced models. To achieve this, transfer learning is an intuitive option, which generally first pretrains a representation learning model, usually by an autoencoder or its extensions, and then removes decoder and adds the pretrained encoder to another neural network for clustering. For instance, DESC [116] engaged a stacked autoencoder and pretrained it to learn low-dimensional representations of cells. After pretraining, the encoder was added to the neural network for cell clustering, in which batch effect can be removed over iterations in model training. Count-adapted regularized deep embedded clustering (CarDEC) [117] is an advanced deep architecture that enables simultaneous batch effect correction, denoising, and clustering of scRNA-seq data. An innovation of CarDEC is that it treats the highly variable genes (HVGs) and lowly variably genes (LVGs) as different feature blocks. Specifically, it pretrained an autoencoder using HVGs, which were combined with LVG features for representation learning and clustering.

In addition, some authors designed hybrid deep architectures for joint representation learning and clustering. For instance, single-cell zero-inflated deep soft K-means (scziDesk) [118] learned data representation using ZINB autoencoder while capturing non-linear dependencies between genes, and fed the learned representations to soft K-means clustering. The ZINB autoencoder and clustering module were trained jointly. GraphSCC [24] is a deep graph-based model for cell clustering. It contains three components: a DAE that encodes input gene expression profiles for preserving local structure, a GCN encodes structural information of the cell–cell network, and a dual self-supervised module that connects the above two modules to learn informative latent representations of data and discover cluster structures. The low-dimensional representations learned by GraphSCC showed superior intra-cluster compactness and inter-cluster separability. Single-cell GNN (scGNN) [119] is a hypothesis-free deep learning framework that integrates autoencoder, GNN, and left truncated mixed Gaussian modeling for scRNA-seq data analysis. scGNN performs imputation, representation learning, and clustering simultaneously, but also can produce a learned cell–cell interaction network.

All in all, both the sequential modeling approaches and simultaneous modeling approaches have shown improvement in cell clustering based on scRNA-seq data compared to the traditional non-deep clustering approaches. However, there

has not been a direct comparison to show that performing the tasks sequentially or simultaneously has a strong impact on downstream analysis. This may be a future area of discussion and could be helpful when identifying which approach to use. In addition, tuning of the number of clusters based on the number of different cell types, and similarity between those cell types is something that is not fully investigated.

Cell annotation

After cell clustering analysis, there is the need of interpreting or annotating the cell sub-populations, which is the so-called cell annotation. Traditionally, cell annotation can be done by identifying gene markers or gene signatures which are differentially expressed in the specific cell cluster and interpreting it manually [125]. However, such approaches are both labor- and resource-consuming. To address this, researchers are seeking deep learning approaches [126–146] that can handle this task with limited human supervision.

The supervised classification model, which can predict types or states of unlabeled cells based on labeled cells, is a good fit to address this task. For instance, scAnCluster [126] designed a hybrid deep model, which combined a cell type classifier with autoencoder for representation learning and clustering. Joint integration and discrimination (JIND) [127] used a GAN style deep architecture, where an encoder is pretrained on classification tasks instead of using an autoencoder framework. The model is also able to account for batch effects. ItClust [128] engaged a transfer learning framework that pretrained model in source data to capture cell-type-specific gene expression information and then transferred model to identify and annotate clusters in the target data. scDeepSort [129] used an advanced GNN, GraphSAGE, to perform supervised classification for cell type annotation, accounting for cell interactions. AutoClass [130] used an autoencoder, where the output reconstruction loss is combined with a classification loss, for cell annotation with data imputation.

It is not uncommon to have only a subset of cells available for analysis with some level of annotation. In this context, semi-supervised learning, which can take full advantage of both labeled and unlabeled data to train a model, has been used in computational cell annotation. Single-cell annotation using variational inference (scANVI) [131] is an extension of scVI [55] by incorporating semi-supervised learning to address cell type annotation with partial label information. scSemiCluster [132] learned cell labels using the combination of unlabeled data and labeled data with an additional cluster compactness loss based on similarity matrix generation. scAdapt [133] used an adversarial training approach to perform semi-supervised cell type annotation. Specifically, it introduced the domain adaptation in DNN to include both adversary-based global distribution alignment and class-level alignment to preserve discriminations between cell clusters in the latent space. scAdapt has shown significance in cell annotation in simulated, cross-platforms, cross-species, and spatial transcriptomic datasets. scArches [134] used an architecture by concatenating nodes for new batches or datasets to existing autoencoder frameworks, to leverage information from other data sources. Moreover, in order for the utilization of the existing annotations to accelerate curation of newly sequenced cells, deep learning-based cell-querying approach has been proposed. Cell BLAST uses large scale reference databases with an autoencoder-based generative model to build low-dimensional representations of cells, and uses a developed cell similarity metric, normalized projection distance, to map query cells to a specific cell type and allow for novel cell types to be identified [147].

Lastly, there is the situation where cell label information is very limited. To address this, there has been a study based on meta-learning to identify previously uncharacterized cell types. The meta-learning can train model to learn from models of known cell type classification to predict never-before-seen cell types. An existing effort in this context is the MARS [135], which used a DFNN as an embedding function to encode gene expression profiles. Under the meta-learning framework, the DFNN was shared by all experiments in the meta-dataset, which enables MARS to generalize to an unannotated experiment to address never-before-seen cell types.

Trajectory inference

Biological questions can be answered by analyzing how cells change as they move from one cell type to another or one cell stage to another. Trajectory analysis in scRNA-seq is an approach to interrogate this type of question [7]. A "pseudotime" or developmental ranking of cells is established, such that the analysis seeks for how gene expression changes as a function of this time. The key process that is used for many approaches is transforming a latent representation of the model into a graph structure. Next, the model usually requires a start cell, which in developmental analyses is usually-one with some "stem-like" marker. The algorithms developed the graph traversal, usually the novel component of most algorithms, to find a path from the start cell to several terminal states. Standard scRNA-seq data analysis tools that provide trajectory inference include Scanpy [3], Monocle [4], VIA [148], Palantir [149], *etc*. To date, these tools have been using traditional methods like PCA for data dimensionality reduction for inferring trajectories. Although approaches like VIA claimed that dimensionality reduction is not a necessary step for their algorithm, there remains the comparison between linear and non-linear approaches for dimensionality reduction in this task. Variational inference for trajectory by autoencoder (VITAE) [150] is an existing effort that uses deep learning to advance trajectory inference. Specifically, VITAE combined a VAE for latent representation learning with a hierarchical mixture model to represent the trajectory. The use of a deep learning model, VAE, enables VITAE to recognize non-linear patterns in data and adjust for confounding covariates to integrate multiple datasets at scale.

Open issues and future directions

In this review, we have investigated how deep learning has been incorporated to advance different elements of scRNA-seq data analysis. Despite the promising results obtained using the deep learning techniques, there remain challenges in the field that need to be solved.

Need of benchmarking studies

One of the most pressing needs, especially for the deep learning approaches developed for scRNA-seq analysis, are benchmarking studies. Most of the papers published using deep learning approaches compared performance to other standard methods but didn't go into great depth when comparing across different types of deep learning models. Single-cell experiments can be vastly different, with tissue samples that contain known cell types, such as in the pancreas (alpha cells, beta cells, delta cells, *etc.*) or from much more complex tissues, such as in diseases such as cancer or coronavirus disease (COVID), where there are many different cell types, and variations of cell types present within the tissue sample. However, most methods claimed superior performance only based on a set of example datasets from specific single-cell experiments. What is more, it is difficult to assess, with the vast number of approaches that have been developed, whether a certain regularization term or added preprocessing step is essential for a particular scRNA-seq data analysis. Therefore, to overcome the above issues, one potential way would be to better understand when these deep learning models fail or what the limitations are for these approaches. Understanding the types of deep learning approaches and model structures that can be beneficial in some cases as compared to others would be very important for 1) developing new approaches to handle these shortcomings and 2) guiding the field as to what methods perform better under specific conditions. In addition, another major improvement in the field would be the human cell atlas, *i.e.*, the aggregation of many different human single-cell expression data across many institutions to cover all major organ systems within the body. This will allow for large amounts of annotated scRNA-seq data, from multiple institutions. This collection of data can allow for more comprehensive benchmarking studies, as a dataset for standardized model evaluation, similar to that of ImageNet or CIFAR10 for computer vision algorithm developers. Fortunately, recent work is moving in this direction, as a group has just tested several batch correction approaches using an atlas level amount of single-cell data and another group has tested 45 different single-cell trajectory inference approaches on 110 different single-cell datasets and proposed guidelines for method selection [7,151].

Integrative analysis of multiple datasets

Although deep learning has been involved in continuously increased scRNA-seq data analysis studies, they usually suffer from limited available information of single dataset, on the order of several tens of thousands of single cells, for the analyses. At this point, it may be difficult to identify substantial amounts of rare cell populations and characterize how these rare cell populations change under varying disease states. These datasets are orders of magnitude smaller than datasets in computer vision tasks where deep learning has achieved notable improvements. For example, most deep learning models in computer vision are pretrained on ImageNet, which contains 1.2 million images split between 1000 different classes. With the increasing availability of scRNA-seq data, the use of these smaller datasets for computational analyses may be changing. Recent work by Sikkema et al. [152] uses a combination of 46 different datasets with 2.2 million cells to analyze lung tissue across healthy and diseased patients. The authors specifically did a benchmarking step to identify the appropriate single-cell integration approach to use for their dataset, and found that the deep learning method, scANVI, outperformed all other methods, including the standard pipeline approach of Seurat. In addition, this was further validated in a large-scale benchmarking dataset [151], showing that two out of the three top performing methods were deep learning approaches. The authors suggest that standard approaches for data integration, such as Harmony, work best when biological complexity is small, but are outperformed by deep learning approaches in more complex settings [151]. Additionally, the deep learning use of transfer learning, similar to approaches such as scArches, can be used to save the information gained from large-scale training sets, to additional researchers that do not have access to such large and diverse datasets. This idea of large-scale model training and transfer learning to fine-tune the model is the key aspect of deep learning and a potential future direction in the field of scRNA-seq computational analysis. The field of scRNA-seq is continuing to embrace the concept of open-source data sharing, and new toolkits, such as scverse (https://scverse.org), look to provide a unified framework for doing these large-scale scRNA-seq analyses. Information gathered in these analyses, on top of other large scale data collection efforts such as TCGA, can be utilized to better understand how cellular changes correlate with disease [153]. In addition, for datasets where patient scRNA-seq and additional disease-related information, such as the Human Pancreas Analysis Program (HPAP) PANC-DB dataset [154], information beyond transcriptome data can be used to identify how distinct cellular changes affect clinical phenotypes [155].

Knowledge-enhanced deep modeling

As the field of deep learning has advanced, the "deep" architectures being developed have become more complex and more "black-box" like. In other words, it is difficult to understand and interpret how the models work. To make deep learning useful for clinicians and biology in general, interpreting deep learning models has been an active area of research. In addition, the "deep" architectures may result in the overfitting issue if the developed models are too complex and hence focus on limited details of the data. Meanwhile, the heterogeneous cell populations and the high dimensionality of gene expression profiles challenge the modeling training, potentially leading to underfitting, such that the developed models are not capable of sufficiently capturing patterns within the data. In this context, incorporating biomedical domain knowledge has been a desirable option to account for those issues in data analysis. To date, there have been several existing studies [71,74,76,77] that developed knowledge-enhanced deep learning models for scRNA-seq analysis. Though these models have gained notable improvement in specific application areas, there remains considerable room for improvement as the knowledge used is limited to specific resources like the GO knowledge base. In addition, today's biomedical knowledge graphs (BKGs) [156–158] have been an important biomedical resource that store comprehensive knowledge in biology and medicine and have been engaged to improve omics data analysis [158–161]. Generally, a BKG is a type of biomedical knowledge base with a graph/network structure where nodes are a set of

biomedical entities (*e.g.*, diseases, drugs, genes, and biological processes) and edges between nodes/entities are relations linking the biomedical entities (*e.g.*, drug–treats–disease, disease–associates–gene, and drug–interacts–drug) [156,157,162]. The BKGs have been used to interpret findings from omics data analysis through BKG query. For instance, Santos et al. [158] developed a clinical knowledge graph (CKG) platform, which enables clinically meaningful queries for automated proteomics data analysis, knowledge mining, and visualization. Doddahonnaiah et al. [160] used a BKG derived from literature to augment the annotation and interpretation of scRNA-seq data. The gene–cell type associations in their BKG were used to categorize cell clusters identified by scRNA-seq data. In addition, researchers have been seeking new methods to develop BKG-guided machine learning and deep learning models to improve scRNA-seq data analysis. In their recent work, Cao and Gao [161] developed a deep learning model for multi-omics single-cell data integration and regulatory inference. Specifically, a graph VAE was used to learn feature embeddings from a prior knowledge-based guidance graph (a specific BKG), which were then fed to the omics VAE to reconstruct omics data via inner product with cell embeddings. In this way, unpaired multi-omics single-cell data such as scRNA-seq, single-cell sequencing assay for transposase-accessible chromatin (scATAC-seq), and single-nucleus methylcytosine sequencing (*i.e.*, snmC-seq, with non-overlapped samples and features) can be projected to the shared cell embedding space.

Integrative modeling with multi-omics data

The ever-improving single-cell isolation and barcoding techniques have been producing diverse omics data at single-cell level, such as genetics, genomics, transcriptomics, and proteomics [163]. On the other hand, integrative analyses of multi-omics data at the bulk level [164–166] have shown the promise to provide a comprehensive understanding of molecular mechanisms to accelerate biology and medicine, as it provides the route to study molecular processes from multiple angles. Compared to traditional machine learning methods, deep learning has demonstrated its superiority in bulk multi-omics data analysis [167–169], due to the capacity in capturing informative latent features from the high-dimensional heterogeneous multi-omics feature space, and the flexible architecture that can model each modality separately using small DNNs (*e.g.*, autoencoders) and combine them later to aggregate information extracted from each modality appropriately to learn a joint representation [170]. Drawing on the success in bulk multi-omics data, integrating scRNA-seq data with other single-cell omics data as well as multi-omics data at bulk level using deep learning may help provide a better and deeper understanding of the biological mechanisms. Although there have been many successes in multi-omics data integration [171–175], there remains specific and distinct challenges, for both joint-modality single-cell sequencing, such as Cellular Indexing of Transcriptomes and Epitopes by Sequencing (CITE-seq) [176], and the integration of single-modality single-cell omics sequencing data. For joint-modality sequencing, to leverage both datasets simultaneously, most methods employ a method of joint representation learning, or finding a shared latent representation of the data for all modalities. One challenge with this type of joint sequencing is that there can be an increase in noise and sparsity in the data, compared to scRNA-seq data using one modality [177]. In addition, it is difficult for the balance of both modalities during the embedding process, and it is possible that some modalities can dominate the downstream embedding tasks leading to the reduction of biological variability that exists within one modality. Finally, there are inherent biases [178] between different institutions, making joint learning more challenging when generalizing across institutions. Additionally, joint sequencing models are much less frequent than single-modality sequencing methods, so an important direction for analysis is to develop methods to integrate two different modalities with unique cell populations. In this setting some goals would be to predict the expression of one modality from another or identify cells in the same cellular state across different modalities. This remains a big computational challenge, as highlighted by the 2021 NeurIPS single-cell challenge. Several methods were developed in this challenge as well as outside, but more work can be done to improve overall performance and more work can be done to improve multi-omics analysis when unique or rare cell populations are in one technology, but not present within another.

Spatially resolved transcriptomics (SRT) is a new approach to single-cell analysis that preserves the spatial relationship of RNA-seq within a tissue. Although SRT has the advantage of spatial resolution, the major technology currently on the market, the 10x Genomics Visium platform, currently generates 50 micron spots that are pooled for analysis, losing the ability to identify the transcriptome of a single cell. There are other approaches that aim to improve the resolution, such as Slide-seqv2 [179], but these too have drawbacks such as limited ability to detect low-expression genes compared with scRNA-seq methods [180]. It is therefore important to realize that scRNA-seq can act as the complementation for the SRT technology. Firstly, SRT will require unique computational and deep learning algorithms, separate from scRNA-seq. For example, a method PASTE [181], shows that scRNA-seq methods are insufficient to properly analyze SRT data. In addition, cell–cell communication networks can be elucidated using newly developed algorithms [182]. However, scRNA-seq currently can provide unique gene information that has been leveraged during SRT analysis. For example, DestVI uses a reference scRNA-seq dataset to deconvolve or attempt to identify unique cell types within a given SRT spot [183]. In addition, work has been done to jointly embed sequential fluorescence in situ hybridization (seqFISH) data and an scRNA-seq atlas, to annotate specific cell types in the seqFISH dataset [184]. Therefore, with current SRT spatial resolution constraints and detection limitations, SRT and scRNA-seq can act synergistically. Additionally, the autoencoder structures used in the context of scRNA-seq and in this review can also be components used within SRT analysis.

Golden standard pipeline

We have discussed deep learning applications in steps in scRNA-seq data preprocessing, including data imputation, representation learning, doublet removal, batch effect removal, and cell cycle regression, and scRNA-seq data downstream

analyses, such as cell clustering, cell annotation, and trajectory inference. However, there are several steps in the pipeline that we have discussed, such as doublet detection and imputation, not always used for analysis. The well-established software like Seurat and Scanpy do allow users to customize the analysis pipeline according to the application scenarios. Efforts like scAEspy [185] and sfaira [186] also built deep learning-based scRNA-seq data analysis pipelines. It will be important to perform thorough comparisons to validate 1) the need for each of these steps, 2) the better way to arrange them in the analysis pipeline, and 3) how deep learning impacts these steps to advance the whole analysis pipeline. There should be systematic effort to determine critical steps in the scRNA-seq analysis pipeline to assure that methods are being developed for critical steps in the analysis.

Conclusion

scRNA-seq has been a critical technique to study cell-level gene expression. Deep learning, a powerful artificial intelligence technique that has shown high capacity in big data mining and outperforms the conventional machine learning, has now firmly been introduced in scRNA-seq data analysis. Specifically, deep learning has been involved in key steps to advance scRNA-seq data analysis. Notable achievements have been gained through the use of deep learning techniques compared to the traditional data analysis methods. By carefully reviewing and comparing existing applications of deep learning in scRNA-seq data analysis, we summarize the challenges that the current deep learning applications are faced with and discuss potential future directions in this field.

CRediT author statement

Matthew Brendel: Conceptualization, Investigation, Writing - original draft. **Chang Su:** Writing - original draft, Writing - review & editing, Visualization. **Zilong Bai:** Writing - review & editing. **Hao Zhang:** Writing - review & editing. **Olivier Elemento:** Writing - review & editing. **Fei Wang:** Supervision, Conceptualization, Writing - review & editing. All authors have read and approved the final manuscript.

Competing interests

The authors have declared no competing interests.

Acknowledgments

Fei Wang would like to acknowledge the support from the National Science Foundation, USA (Grant No. 1750326) and the National Institutes of Health, USA (Grant Nos. R01MH124740 and RF1AG072449).

Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.gpb.2022.11.011.

ORCID

ORCID 0000-0003-3417-4597 (Matthew Brendel)
ORCID 0000-0003-4019-6389 (Chang Su)
ORCID 0000-0002-3891-8015 (Zilong Bai)
ORCID 0000-0002-2928-2692 (Hao Zhang)
ORCID 0000-0002-8061-9617 (Olivier Elemento)
ORCID 0000-0001-9459-9461 (Fei Wang)

References

[1] Tang F, Barbacioru C, Wang Y, Nordman E, Lee C, Xu N, et al. mRNA-seq whole-transcriptome analysis of a single cell. Nat Methods 2009;6:377–82.
[2] Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck 3rd WM, et al. Comprehensive integration of single-cell data. Cell 2019;177:1888–902.
[3] Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol 2018;19:15.
[4] Trapnell C, Cacchiarelli D, Grimsby J, Pokharel P, Li S, Morse M, et al. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nat Biotechnol 2014;32:381–6.
[5] Amezquita RA, Lun ATL, Becht E, Carey VJ, Carpp LN, Geistlinger L, et al. Orchestrating single-cell analysis with Bioconductor. Nat Methods 2020;17:137–45.
[6] Hafemeister C, Satija R. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. Genome Biol 2019;20:296.
[7] Saelens W, Cannoodt R, Todorov H, Saeys Y. A comparison of single-cell trajectory inference methods. Nat Biotechnol 2019;37:547–54.
[8] Libbrecht MW, Noble WS. Machine learning applications in genetics and genomics. Nat Rev Genet 2015;16:321–32.
[9] Reel PS, Reel S, Pearson E, Trucco E, Jefferson E. Using machine learning approaches for multi-omics data analysis: a review. Biotechnol Adv 2021;49:107739.
[10] Ma Q, Xu D. Deep learning shapes single-cell data analysis. Nat Rev Mol Cell Biol 2022;23:303–4.
[11] Flores M, Liu Z, Zhang T, Hasib MM, Chiu YC, Ye Z, et al. Deep learning tackles single-cell analysis — a survey of deep learning for scRNA-seq analysis. Brief Bioinform 2022;23:bbab531.
[12] Bao S, Li K, Yan C, Zhang Z, Qu J, Zhou M. Deep learning-based advances and applications for single-cell RNA sequencing data analysis. Brief Bioinform 2022;23:bbab473.
[13] Riba A, Oravecz A, Durik M, Jiménez S, Alunni V, Cerciat M, et al. Cell cycle gene regulation dynamics revealed by RNA velocity and deep learning. Nat Commun 2021;13:2865.
[14] Qiao C, Huang Y. Representation learning of RNA velocity reveals robust cell transitions. Proc Natl Acad Sci U S A 2021;118:e2105859118.
[15] Gong B, Zhou Y, Purdom E. Cobolt: integrative analysis of multimodal single-cell sequencing data. Genome Biol 2021;22:351.
[16] Minoura K, Abe K, Nam H, Nishikawa H, Shimamura T. A mixture-of-experts deep generative model for integrated analysis of single-cell multiomics data. Cell Rep Methods 2021;1:100071.
[17] Singh R, Hie BL, Narayan A, Berger B. Schema: metric learning enables interpretable synthesis of heterogeneous single-cell modalities. Genome Biol 2021;22:131.
[18] Tian J, Wang J, Roeder K. ESCO: single cell expression simulation incorporating gene co-expression. Bioinformatics 2021;37:2374–81.
[19] Heydari AA, Davalos OA, Zhao L, Hoyer KK, Sindi SS. ACTIVA: realistic single-cell RNA-seq generation with auto-

matic cell-type identification using introspective variational autoencoders. Bioinformatics 2022;38:2194–201.
[20] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436–44.
[21] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 2010;11:3371–408.
[22] Liou CY, Cheng WC, Liou JW, Liou DR. Autoencoder for words. Neurocomputing 2014;139:84–96.
[23] Srinivasan S, Leshchyk A, Johnson NT, Korkin D. A hybrid deep clustering approach for robust cell type profiling using single-cell RNA-seq data. RNA 2020;26:1303–19.
[24] Zeng Y, Lin J, Zhou X, Lu Y, Yang Y. Graph convolutional network-based method for clustering single-cell RNA-seq data. bioRxiv 2021;278804.
[25] Ding J, Condon A, Shah SP. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. Nat Commun 2018;9:2002.
[26] Mitra R, MacLean AL. RVAgene: generative modeling of gene expression time series data. Bioinformatics 2021;37:3252–62.
[27] Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, et al. β-VAE: learning basic visual concepts with a constrained variational framework. 5th International Conference on Learning Representations 2017:1–13.
[28] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. Proceedings of the 27th International Conference on Neural Information Processing Systems 2014:2672–80.
[29] Makhzani A, Shlens J, Jaitly N, Goodfellow I, Frey B. Adversarial autoencoders. arXiv 2015;1511.05644.
[30] Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, et al. Graph neural networks: a review of methods and applications. AI Open 2020;1:57–81.
[31] Li J, Jiang W, Han H, Liu J, Liu B, Wang Y. ScGSLC: an unsupervised graph similarity learning framework for single-cell RNA-seq data clustering. Comput Biol Chem 2021;90:107415.
[32] Zheng GXY, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, et al. Massively parallel digital transcriptional profiling of single cells. Nat Commun 2017;8:14049.
[33] Patruno L, Maspero D, Craighero F, Angaroni F, Antoniotti M, Graudenzi A. A review of computational strategies for denoising and imputation of single-cell transcriptomic data. Brief Bioinform 2020;22:bbaa222.
[34] van Dijk D, Sharma R, Nainys J, Yim K, Kathail P, Carr AJ, et al. Recovering gene interactions from single-cell data using data diffusion. Cell 2018;174:716–29.
[35] Ronen J, Akalin A. netSmooth: network-smoothing based imputation for single cell RNA-seq. F1000Res 2018;7:8.
[36] Huang M, Wang J, Torre E, Dueck H, Shaffer S, Bonasio R, et al. SAVER: gene expression recovery for single-cell RNA sequencing. Nat Methods 2018;15:539–42.
[37] Arisdakessian C, Poirion O, Yunits B, Zhu X, Garmire LX. DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. Genome Biol 2019;20:211.
[38] Xu Y, Zhang Z, You L, Liu J, Fan Z, Zhou X. scIGANs: single-cell RNA-seq imputation using generative adversarial networks. Nucleic Acids Res 2020;48:e85.
[39] Yu B, Chen C, Qi R, Zheng R, Skillman-Lawrence PJ, Wang X, et al. scGMAI: a Gaussian mixture model for clustering single-cell RNA-seq data based on deep autoencoder. Brief Bioinform 2021;22:bbaa316.
[40] Wang J, Agarwal D, Huang M, Hu G, Zhou Z, Ye C, et al. Data denoising with transfer learning in single-cell transcriptomics. Nat Methods 2019;16:875–8.
[41] Eraslan G, Simon LM, Mircea M, Mueller NS, Theis FJ. Single-cell RNA-seq denoising using a deep count autoencoder. Nat Commun 2019;10:390.
[42] Tian T, Min MR, Wei Z. Model-based autoencoders for imputing discrete single-cell RNA-seq data. Methods 2021;192:112–9.
[43] Chi W, Deng M. Sparsity-penalized stacked denoising autoencoders for imputing single-cell RNA-seq data. Genes 2020;11:532.
[44] Rao J, Zhou X, Lu Y, Zhao H, Yang Y. Imputing single-cell RNA-seq data by combining graph convolution and autoencoder neural networks. iScience 2021;24:102393.
[45] Huang M, Zhang Z, Zhang NR. Dimension reduction and denoising of single-cell RNA sequencing data in the presence of observed confounding variables. bioRxiv 2020;234765.
[46] Li X, Li S, Huang L, Zhang S, Wong KC. High-throughput single-cell RNA-seq data imputation and characterization with surrogate-assisted automated deep learning. Brief Bioinform 2022;23:bbab368.
[47] Xu L, Xu Y, Xue T, Zhang X, Li J. AdImpute: an imputation method for single-cell RNA-seq data based on semi-supervised autoencoders. Front Genet 2021;12:739677.
[48] Xu C, Cai L, Gao J. An efficient scRNA-seq dropout imputation method using graph attention network. BMC Bioinformatics 2021;22:582.
[49] Badsha MB, Li R, Liu B, Li YI, Xian M, Banovich NE, et al. Imputation of single-cell gene expression with an autoencoder neural network. Quant Biol 2020;8:78–94.
[50] Enders CK. Applied missing data analysis. New York: Guilford press; 2010.
[51] Jang E, Gu S, Poole B. Categorical reparameterization with Gumbel-Softmax. arXiv 2017;1611.01144.
[52] Svensson V. Droplet scRNA-seq is not zero-inflated. Nat Biotechnol 2020;38:147–50.
[53] Gierahn TM, Wadsworth 2nd MH, Hughes TK, Bryson BD, Butler A, Satija R, et al. Seq-Well: portable, low-cost RNA sequencing of single cells at high throughput. Nat Methods 2017;14:395–8.
[54] Bernstein NJ, Fong NL, Lam I, Roy MA, Hendrickson DG, Kelley DR. Solo: doublet identification in single-cell RNA-seq via semi-supervised deep learning. Cell Syst 2020;11:95–101.
[55] Lopez R, Regier J, Cole MB, Jordan MI, Yosef N. Deep generative modeling for single-cell transcriptomics. Nat Methods 2018;15:1053–8.
[56] Wolock SL, Lopez R, Klein AM. Scrublet: computational identification of cell doublets in single-cell transcriptomic data. Cell Syst 2019;8:281–91.
[57] McGinnis CS, Murrow LM, Gartner ZJ. DoubletFinder: doublet detection in single-cell RNA sequencing data using artificial nearest neighbors. Cell Syst 2019;8:329–37.
[58] Tzur A, Kafri R, LeBleu VS, Lahav G, Kirschner MW. Cell growth and size homeostasis in proliferating animal cells. Science 2009;325:167–71.
[59] Deng Y, Bao F, Dai Q, Wu LF, Altschuler SJ. Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning. Nat Methods 2019;16:311–4.
[60] Wang D, Gu J. VASC: dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder. Genomics Proteomics Bioinformatics 2018;16:320–31.
[61] Cho H, Berger B, Peng J. Generalizable and scalable visualization of single-cell data using neural networks. Cell Syst 2018;7:185–91.
[62] Tran D, Nguyen H, Tran B, La Vecchia C, Luu HN, Nguyen T. Fast and precise single-cell data analysis using a hierarchical autoencoder. Nat Commun 2021;12:1029.

[63] Grønbech CH, Vording MF, Timshel PN, Sønderby CK, Pers TH, Winther O. scVAE: variational auto-encoders for single-cell gene expression data. Bioinformatics 2020;36:4415–22.
[64] Ding J, Regev A. Deep generative model embedding of single-cell RNA-seq profiles on hyperspheres and hyperbolic spaces. Nat Commun 2021;12:2554.
[65] Bica I, Andrés-Terré H, Cvejic A, Liò P. Unsupervised generative and graph representation learning for modelling cell differentiation. Sci Rep 2020;10:9790.
[66] Zhang C. Single-cell data analysis using MMD variational autoencoder for a more informative latent representation. bioRxiv 2019;613414.
[67] Lin E, Mukherjee S, Kannan S. A deep adversarial variational autoencoder model for dimensionality reduction in single-cell RNA sequencing analysis. BMC Bioinformatics 2020;21:64.
[68] Mondal AK, Asnani H, Singla P, Ap P. scRAE: deterministic regularized autoencoders with flexible priors for clustering single-cell gene expression data. IEEE/ACM Trans Comput Biol Bioinform 2022;19:2996–3007.
[69] Kimmel JC. Disentangling latent representations of single cell RNA-seq experiments. bioRxiv 2020;972166.
[70] Luo Z, Xu C, Zhang Z, Jin W. A topology-preserving dimensionality reduction method for single-cell RNA-seq data using graph autoencoder. Sci Rep 2021;11:20028.
[71] Alessandri L, Cordero F, Beccuti M, Licheri N, Arigoni M, Olivero M, et al. Sparsely-connected autoencoder (SCA) for single cell RNA-seq data mining. NPJ Syst Biol Appl 2021;7:1.
[72] Peng J, Wang X, Shang X. Combining gene ontology with deep neural networks to enhance the clustering of single cell RNA-seq data. BMC Bioinformatics 2019;20:284.
[73] Zhang S, Li X, Lin Q, Lin J, Wong KC. Uncovering the key dimensions of high-throughput biomolecular data using deep learning. Nucleic Acids Res 2020;48:e56.
[74] Gut G, Stark SG, Rätsch G, Davidson NR. pmVAE: learning interpretable single-cell representations with pathway modules. bioRxiv 2021;428664.
[75] Seninge L, Anastopoulos I, Ding H, Stuart J. Biological network-inspired interpretable variational autoencoder. bioRxiv 2020;423310.
[76] Rybakov S, Lotfollahi M, Theis FJ, Wolf FA. Learning interpretable latent autoencoder representations with annotations of feature sets. bioRxiv 2020;401182.
[77] Svensson V, Gayoso A, Yosef N, Pachter L. Interpretable factor models of single-cell RNA-seq via variational autoencoders. Bioinformatics 2020;36:3418–21.
[78] Zhao J, Wang N, Wang H, Zheng C, Su Y. SCDRHA: a scRNA-seq data dimensionality reduction algorithm based on hierarchical autoencoder. Front Genet 2021;12:733906.
[79] Wang H, Zhao J, Su Y, Zheng CH. scCDG: a method based on DAE and GCN for scRNA-seq data analysis. IEEE/ACM Trans Comput Biol Bioinform 2022;19:3685–94.
[80] Buterez D, Bica I, Tariq I, Andrés-Terré H, Liò P. CellVGAE: an unsupervised scRNA-seq analysis workflow with graph attention networks. Bioinformatics 2022;38:1277–86.
[81] Ciortan M, Defrance M. GNN-based embedding for clustering scRNA-seq data. Bioinformatics 2022;38:1037–44.
[82] Ciortan M, Defrance M. Contrastive self-supervised clustering of scRNA-seq data. BMC Bioinformatics 2021;22:280.
[83] Lukassen S, Ten FW, Adam L, Eils R, Conrad C. Gene set inference from single-cell sequencing data using a hybrid of matrix factorization and variational autoencoders. Nat Mach Intell 2020;2:800–9.
[84] Prince E, Hankinson TC. HD Spot: interpretable deep learning classification of single cell transcript data. bioRxiv 2019;822759.
[85] Fortelny N, Bock C. Knowledge-primed neural networks enable biologically interpretable deep learning on single-cell sequencing data. Genome Biol 2020;21:190.
[86] Gold MP, LeNail A, Fraenkel E. Shallow sparsely-connected autoencoders for gene set projection. Pac Symp Biocomput 2019;24:374–85.
[87] Yu H, Welch JD. MichiGAN: sampling from disentangled representations of single-cell data using generative adversarial networks. Genome Biol 2021;22:158.
[88] Kingma DP, Welling M. An introduction to variational autoencoders. Found Trends Mach Learn 2019;12:307–92.
[89] Davidson TR, Falorsi L, De Cao N, Kipf T, Tomczak JM. Hyperspherical variational auto-encoders. Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence 2018:856–65.
[90] Zhao S, Song J, Ermon S. InfoVAE: information maximizing variational autoencoders. arXiv 2017;1706.02262.
[91] Licata L, Lo Surdo P, Iannuccelli M, Palma A, Micarelli E, Perfetto L, et al. SIGNOR 2.0, the SIGnaling Network Open Resource 2.0: 2019 update. Nucleic Acids Res 2020(48):D504–10.
[92] Han H, Shim H, Shin D, Shim JE, Ko Y, Shin J, et al. TRRUST: a reference database of human transcriptional regulatory interactions. Sci Rep 2015;5:11432.
[93] Zhao M, He W, Tang J, Zou Q, Guo F. A hybrid deep learning framework for gene regulatory network inference from single-cell transcriptomic data. Brief Bioinform 2022;23:bbab568.
[94] Gene Ontology Consortium. The Gene Ontology (GO) database and informatics resource. Nucleic Acids Res 2004;32:D258–61.
[95] Xu Y, Das P, McCord RP. SMILE: mutual information learning for integration of single-cell omics data. Bioinformatics 2021;38:476–86.
[96] Hu J, Zhong Y, Shang X. Efficient and scalable integration of single-cell data using domain-adversarial and variational approximation. bioRxiv 2021;438733.
[97] Xiong L, Tian K, Li Y, Ning W, Gao X, Zhang QC. Online single-cell data integration through projecting heterogeneous datasets into a common cell-embedding space. Nat Commun 2022;13:6118.
[98] Dincer AB, Janizek JD, Lee SI. Adversarial deconfounding autoencoder for learning robust gene expression embeddings. Bioinformatics 2020;36:i573–82.
[99] Bahrami M, Maitra M, Nagy C, Turecki G, Rabiee HR, Li Y. Deep feature extraction of single-cell transcriptomes by generative adversarial network. Bioinformatics 2021;37:1345–51.
[100] Wang D, Hou S, Zhang L, Wang X, Liu B, Zhang Z. iMAP: integration of multiple single-cell datasets by adversarial paired transfer networks. Genome Biol 2021;22:63.
[101] Wang T, Johnson TS, Shao W, Lu Z, Helm BR, Zhang J, et al. BERMUDA: a novel deep transfer learning method for single-cell RNA sequencing batch correction reveals hidden high-resolution cellular subtypes. Genome Biol 2019;20:165.
[102] Lotfollahi M, Naghipourfar M, Theis FJ, Wolf FA. Conditional out-of-distribution generation for unpaired data using transfer VAE. Bioinformatics 2020;36:i610–7.
[103] Ge S, Wang H, Alavi A, Xing E, Bar-joseph Z. Supervised adversarial alignment of single-cell RNA-seq data. J Comput Biol 2021;28:501–13.
[104] Zhao Y, Cai H, Zhang Z, Tang J, Li Y. Learning interpretable cellular and gene signature embeddings from single-cell transcriptomic data. Nat Commun 2021;12:5261.
[105] Pang M, Tegnér J. Multitask learning for transformers with application to large-scale single-cell transcriptomes. bioRxiv 2020;935239.
[106] Zou B, Zhang T, Zhou R, Jiang X, Yang H, Jin X, et al. deepMNN: deep learning-based single-cell RNA sequencing data batch correction using mutual nearest neighbors. Front Genet 2021;12:708981.
[107] Wang X, Wang J, Zhang H, Huang S, Yin Y. HDMC: a novel deep learning-based framework for removing batch effects in single-cell RNA-seq data. Bioinformatics 2022;38:1295–303.

[108] Yu W, Mahfouz A, Reinders MJT. CBA: cluster-guided batch alignment for single cell RNA-seq. Front Genet 2021;12:644211.
[109] Devlin J, Chang MW, Lee K, Toutanova K. Bert: pre-training of deep bidirectional transformers for language understanding. Proc 2019 Conf North Am Chapter Assoc Comput Linguist Hum Lang Technol 2019:4171–86.
[110] Wu Z, Xiong Y, Yu SX, Lin D. Unsupervised feature learning via non-parametric instance discrimination. IEEE Conf Comput Vis Pattern Recognit 2018:3733–42.
[111] Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008;2008:P10008.
[112] Traag VA, Waltman L, van Eck NJ. From Louvain to Leiden: guaranteeing well-connected communities. Sci Rep 2019;9:5233.
[113] Xie K, Huang Y, Zeng F, Liu Z, Chen T. scAIDE: clustering of large-scale single-cell RNA-seq data reveals putative and rare cell types. NAR Genom Bioinform 2020;2:lqaa082.
[114] Chen L, Wang W, Zhai Y, Deng M. Single-cell transcriptome data clustering via multinomial modeling and adaptive fuzzy k-means algorithm. Front Genet 2020;11:295.
[115] Geddes TA, Kim T, Nan L, Burchfield JG, Yang JYH, Tao D, et al. Autoencoder-based cluster ensembles for single-cell RNA-seq data analysis. BMC Bioinformatics 2019;20:660.
[116] Li X, Wang K, Lyu Y, Pan H, Zhang J, Stambolian D, et al. Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis. Nat Commun 2020;11:2338.
[117] Lakkis J, Wang D, Zhang Y, Hu G, Wang K, Pan H, et al. A joint deep learning model enables simultaneous batch effect correction, denoising, and clustering in single-cell transcriptomics. Genome Res 2021;31:1753–66.
[118] Chen L, Wang W, Zhai Y, Deng M. Deep soft *K*-means clustering with self-training for single-cell RNA sequence data. NAR Genom Bioinform 2020;2:lqaa039.
[119] Wang J, Ma A, Chang Y, Gong J, Jiang Y, Qi R, et al. scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses. Nat Commun 2021;12:1882.
[120] Amodio M, van Dijk D, Srinivasan K, Chen WS, Mohsen H, Moon KR, et al. Exploring single-cell data with deep multitasking neural networks. Nat Methods 2019;16:1139–45.
[121] Li X, Zhang S, Wong KC. Deep embedded clustering with multiple objectives on scRNA-seq data. Brief Bioinform 2021;22:bbab090.
[122] Kopf A, Fortuin V, Somnath VR, Claassen M. Mixture-of-Experts Variational Autoencoder for clustering and generating from similarity-based representations. PLoS Comput Biol 2021;17:e1009086.
[123] Gionis A, Indyk P, Motwani R. Similarity search in high dimensions via hashing. Proc 25th Int Conf Very Large Data Bases 1999:518–29.
[124] Do CB, Batzoglou S. What is the expectation maximization algorithm? Nat Biotechnol 2008;26:897–9.
[125] Clarke ZA, Andrews TS, Atif J, Pouyabahar D, Innes BT, MacParland SA, et al. Tutorial: guidelines for annotating single-cell transcriptomic maps using automated and manual methods. Nat Protoc 2021;16:2749–64.
[126] Chen L, Zhai Y, He Q, Wang W, Deng M. Integrating deep supervised, self-supervised and unsupervised learning for single-cell RNA-seq clustering and annotation. Genes 2020;11:792.
[127] Goyal M, Serrano G, Shomorony I, Hernaez M, Ochoa I. JIND: joint integration and discrimination for automated single-cell annotation. Bioinformatics 2022;38:2488–95.
[128] Hu J, Li X, Hu G, Lyu Y, Susztak K, Li M. Iterative transfer learning with neural network for clustering and cell type classification in single-cell RNA-seq analysis. Nat Mach Intell 2020;2:607–18.
[129] Shao X, Yang H, Zhuang X, Liao J, Yang P, Cheng J, et al. scDeepSort: a pre-trained cell-type annotation method for single-cell transcriptomics using deep learning with a weighted graph neural network. Nucleic Acids Res 2021;49:e122.
[130] Li H, Brouwer CR, Luo W. A universal deep neural network for in-depth cleaning of single-cell RNA-seq data. Nat Commun 2022;13:1901.
[131] Xu C, Lopez R, Mehlman E, Regier J, Jordan MI, Yosef N. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. Mol Syst Biol 2021;17:e9620.
[132] Chen L, He Q, Zhai Y, Deng M. Single-cell RNA-seq data semi-supervised clustering and annotation via structural regularized domain adaptation. Bioinformatics 2021;37:775–84.
[133] Zhou X, Chai H, Zeng Y, Zhao H, Yang Y. scAdapt: virtual adversarial domain adaptation network for single cell RNA-seq data classification across platforms and species. Brief Bioinform 2021;22:bbab281.
[134] Lotfollahi M, Naghipourfar M, Luecken MD, Khajavi M, Büttner M, Wagenstetter M, et al. Mapping single-cell data to reference atlases by transfer learning. Nat Biotechnol 2022;40:121–30.
[135] Brbić M, Zitnik M, Wang S, Pisco AO, Altman RB, Darmanis S, et al. MARS: discovering novel cell types across heterogeneous single-cell experiments. Nat Methods 2020;17:1200–6.
[136] Zhang J, Zhang X, Wang Y, Zeng F, Zhao XM. MAT2: manifold alignment of single-cell transcriptomes with cell triplets. Bioinformatics 2021;37:3263–9.
[137] Kimmel JC, Kelley DR. Semisupervised adversarial neural networks for single-cell classification. Genome Res 2021;31:1781–93.
[138] Song Q, Su J, Zhang W. scGCN is a graph convolutional networks algorithm for knowledge transfer in single cell omics. Nat Commun 2021;12:3826.
[139] Yuan M, Chen L, Deng M. scMRA: a robust deep learning method to annotate scRNA-seq data with multiple reference datasets. Bioinformatics 2022;38:738–45.
[140] Koh W, Hoon S. MapCell: learning a comparative cell type distance metric with siamese neural nets with applications toward cell-type identification across experimental datasets. Front Cell Dev Biol 2021;9:767897.
[141] Wang T, Bai J, Nabavi S. Single-cell classification using graph convolutional networks. BMC Bioinformatics 2021;22:364.
[142] Yin Q, Wang Y, Guan J, Ji G. scIAE: an integrative autoencoder-based ensemble classification framework for single-cell RNA-seq data. Brief Bioinform 2021;23:bbab508.
[143] Duan B, Chen S, Chen X, Zhu C, Tang C, Wang S, et al. Integrating multiple references for single-cell assignment. Nucleic Acids Res 2021;49:e80.
[144] Liu X, Gosline SJC, Pflieger LT, Wallet P, Iyer A, Guinney J, et al. Knowledge-based classification of fine-grained immune cell types in single-cell RNA-seq data. Brief Bioinform 2021;22:bbab039.
[145] Dong Z, Alterovitz G. netAE: semi-supervised dimensionality reduction of single-cell RNA sequencing to facilitate cell labeling. Bioinformatics 2021;37:43–9.
[146] Wang L, Miao X, Nie R, Zhang Z, Zhang J, Cai J. Multi-CapsNet: a general framework for data integration and interpretable classification. Front Genet 2021;12:767602.
[147] Cao ZJ, Wei L, Lu S, Yang DC, Gao G. Searching large-scale scRNA-seq databases via unbiased cell embedding with Cell BLAST. Nat Commun 2020;11:3458.
[148] Stassen SV, Yip GGK, Wong KKY, Ho JWK, Tsia KK. Generalized and scalable trajectory inference in single-cell omics data with VIA. Nat Commun 2021;12:5528.
[149] Setty M, Kiseliovas V, Levine J, Gayoso A, Mazutis L, Pe'er D. Characterization of cell fate probabilities in single-cell data with Palantir. Nat Biotechnol 2019;37:451–60.

[150] Du JH, Gao M, Wang J. Model-based trajectory inference for single-cell RNA sequencing using deep learning with a mixture prior. bioRxiv 2020;424452.
[151] Luecken MD, Büttner M, Chaichoompu K, Danese A, Interlandi M, Mueller MF, et al. Benchmarking atlas-level data integration in single-cell genomics. Nat Methods 2022;19:41–50.
[152] Sikkema L, Strobl D, Zappia L, Madissoon E, Markov NS, Zaragosi L, et al. An integrated cell atlas of the human lung in health and disease. bioRxiv 2022;483747.
[153] Cheng S, Li Z, Gao R, Xing B, Gao Y, Yang Y, et al. A pan-cancer single-cell transcriptional atlas of tumor infiltrating myeloid cells. Cell 2021;184:792–809.
[154] Kaestner KH, Powers AC, Naji A, HPAP Consortium, Atkinson MA. NIH initiative to improve understanding of the pancreas, islet, and autoimmunity in type 1 diabetes: the Human Pancreas Analysis Program (HPAP). Diabetes 2019;68:1394–402.
[155] Dann E, Henderson NC, Teichmann SA, Morgan MD, Marioni JC. Differential abundance testing on single-cell data using k-nearest neighbor graphs. Nat Biotechnol 2022;40:245–53.
[156] Himmelstein DS, Lizee A, Hessler C, Brueggeman L, Chen SL, Hadley D, et al. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. Elife 2017;6:e26726.
[157] Su C, Hou Y, Guo W, Chaudhry F, Ghahramani G, Zhang H, et al. iBKH: the integrative Biomedical Knowledge Hub. medRxiv 2021;21253461.
[158] Santos A, Colaço AR, Nielsen AB, Niu L, Strauss M, Geyer PE, et al. A knowledge graph to interpret clinical proteomics data. Nat Biotechnol 2020;40:692–702.
[159] Blatti 3rd C, Emad A, Berry MJ, Gatzke L, Epstein M, Lanier D, et al. Knowledge-guided analysis of "omics" data using the KnowEnG cloud platform. PLoS Biol 2020;18:e3000583.
[160] Doddahonnaiah D, Lenehan PJ, Hughes TK, Zemmour D, Garcia-Rivera E, Venkatakrishnan AJ, et al. A literature-derived knowledge graph augments the interpretation of single cell RNA-seq datasets. Genes 2021;12:898.
[161] Cao ZJ, Gao G. Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. Nat Biotechnol 2022;40:1458–66.
[162] Nicholson DN, Greene CS. Constructing knowledge graphs and their biomedical applications. Comput Struct Biotechnol J 2020;18:1414–28.
[163] Lee J, Hyeon DY, Hwang D. Single-cell multiomics: technologies and data analysis methods. Exp Mol Med 2020;52:1428–42.
[164] Hasin Y, Seldin M, Lusis A. Multi-omics approaches to disease. Genome Biol 2017;18:83.
[165] Subramanian I, Verma S, Kumar S, Jere A, Anamika K. Multi-omics data integration, interpretation, and its application. Bioinform Biol Insights 2020;14:1177932219899051.
[166] Wörheide MA, Krumsiek J, Kastenmüller G, Arnold M. Multi-omics integration in biomedical research - a metabolomics-centric review. Anal Chim Acta 2021;1141:144–62.
[167] Kang M, Ko E, Mersha TB. A roadmap for multi-omics data integration using deep learning. Brief Bioinform 2022;23: bbab454.
[168] Zhang L, Lv C, Jin Y, Cheng G, Fu Y, Yuan D, et al. Deep learning-based multi-omics data integration reveals two prognostic subtypes in high-risk neuroblastoma. Front Genet 2018;9:477.
[169] Chaudhary K, Poirion OB, Lu L, Garmire LX. Deep learning-based multi-omics integration robustly predicts survival in liver cancer. Clin Cancer Res 2018;24:1248–59.
[170] Su C, Xu Z, Pathak J, Wang F. Deep learning in mental health outcome research: a scoping review. Transl Psychiatry 2020;10:116.
[171] Gayoso A, Steier Z, Lopez R, Regier J, Nazor KL, Streets A, et al. Joint probabilistic modeling of single-cell multi-omic data with totalVI. Nat Methods 2021;18:272–82.
[172] Wu KE, Yost KE, Chang HY, Zou J. BABEL enables cross-modality translation between multiomic profiles at single-cell resolution. Proc Natl Acad Sci U S A 2021;118:e2023070118.
[173] Zuo C, Chen L. Deep-joint-learning analysis model of single cell transcriptome and open chromatin accessibility data. Brief Bioinform 2021;22:bbaa287.
[174] Cao ZJ, Gao G. Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. Nat Biotechnol 2022;2:1–9.
[175] Zuo C, Dai H, Chen L. Deep cross-omics cycle attention model for joint analysis of single-cell multi-omics data. Bioinformatics 2021;37:4091–9.
[176] Stoeckius M, Hafemeister C, Stephenson W, Houck-Loomis B, Chattopadhyay PK, Swerdlow H, et al. Simultaneous epitope and transcriptome measurement in single cells. Nat Methods 2017;14:865–8.
[177] Stanojevic S, Li Y, Ristivojevic A, Garmire LX. Computational methods for single-cell multi-omics integration and alignment. Genomics Proteomics Bioinformatics 2022;20:836–49.
[178] Luecken MD, Burkhardt DB, Cannoodt R, Lance C, Agrawal A, Aliee H, et al. A sandbox for prediction and integration of DNA, RNA, and proteins in single cells. Proceeding of the 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track 2021:1–13.
[179] Stickels RR, Murray E, Kumar P, Li J, Marshall JL, Di Bella DJ, et al. Highly sensitive spatial transcriptomics at near-cellular resolution with Slide-seqV2. Nat Biotechnol 2021;39:313–9.
[180] Marshall JL, Noel T, Wang QS, Chen H, Murray E, Subramanian A, et al. High-resolution Slide-seqV2 spatial transcriptomics enables discovery of disease-specific cell neighborhoods and pathways. iScience 2022;25:104097.
[181] Zeira R, Land M, Strzalkowski A, Raphael BJ. Alignment and integration of spatial transcriptomics data. Nat Methods 2022;19:567–75.
[182] Fischer DS, Schaar AC, Theis FJ. Learning cell communication from spatial graphs of cells. bioRxiv 2021;451750.
[183] Lopez R, Li B, Keren-Shaul H, Boyeau P, Kedmi M, Pilzer D, et al. DestVI identifies continuums of cell types in spatial transcriptomics data. Nat Biotechnol 2022;40:1360–9.
[184] Lohoff T, Ghazanfar S, Missarova A, Koulena N, Pierson N, Griffiths JA, et al. Integration of spatial and single-cell transcriptomic data elucidates mouse organogenesis. Nat Biotechnol 2022;40:74–85.
[185] Tangherloni A, Ricciuti F, Besozzi D, Liò P, Cvejic A. Analysis of single-cell RNA sequencing data based on autoencoders. BMC Bioinformatics 2021;22:309.
[186] Fischer DS, Dony L, König M, Moeed A, Zappia L, Heumos L, et al. Sfaira accelerates data and model reuse in single cell genomics. Genome Biol 2021;22:248.
[187] Sabour S, Frosst N, Hinton GE. Dynamic routing between capsules. Proceedings of the 31st International Conference on Neural Information Processing Systems 2017:3859–69.