**Deep learning tackles single-cell analysis –   
A survey of deep learning for scRNA-seq analysis**

**Mario Flores1**§**, Zhentao Liu1, Tinghe Zhang1, Md Musaddaqui Hasib1, Yu-Chiao Chiu2, Zhenqing Ye2,3, Karla Paniagua1, Sumin Jo1, Jianqiu Zhang1, Shou-Jiang Gao4,6, Yufang Jin1, Yidong Chen2,3**§**, and Yufei Huang5,6**§

1Department of Electrical and Computer Engineering, the University of Texas at San Antonio, San Antonio, TX 78249, USA

2Greehey Children’s Cancer Research Institute, University of Texas Health San Antonio, San Antonio, TX 78229, USA

3Department of Population Health Sciences, University of Texas Health San Antonio, San Antonio, TX 78229, USA

4Department of Microbiology and Molecular Genetics, University of Pittsburgh

, Pittsburgh, Pennsylvania, PA 15232, USA

5Department of Medicine, School of Medicine, University of Pittsburgh, PA 15232, USA

6UPMC Hillman Cancer Center, University of Pittsburgh, PA 15232, USA

§**Correspondence** should be addressed to Mario Flores ([mario.flores@utsa.edu](mailto:mario.flores@utsa.edu)); Yidong Chen ([cheny8@uthscsa.edu](mailto:cheny8@uthscsa.edu)); Yufei Huan (yuh119@pitt.edu)

**Running title**: Deep learning for single-cell RNA-seq analysis

**Mario Flores**, PhD, is an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Texas at San Antonio, and joint program Faculty of Biomedical Engineering at the University of Texas Health San Antonio. Before joined ECE, he was a postdoctoral fellow at the National Center for Biotechnology Information of the National Institutes of Health from 2015 to 2019. His research focuses on DNA and RNA sequence methods, transcriptomics analysis, epigenetics, comparative genomics, and deep learning to study mechanisms of gene regulation, single-cell RNA-seq, and Natural Language Processing.

**Zhentao Liu** is a PhD student in the Department of Electrical and Computer Engineering, the University of Texas at San Antonio. His research focuses on deep learning for cancer genomics and drug response prediction.

**Tinghe Zhang** is a PhD student in the Department of Electrical and Computer Engineering, the University of Texas at San Antonio. His research focuses on deep learning for cancer genomics and drug response prediction.

**Md Musaddaqui Hasib** is a PhD student in the Department of Electrical and Computer Engineering, the University of Texas at San Antonio. His research focuses on interpretable deep learning for cancer genomics.

**Zhenqing Ye**, PhD, is an assistant professor in the Department of Population Health Sciences and the director of Computational Biology and Bioinformatics at Greehey Children’s Cancer Research Institute at the University of Texas Health San Antonio. His research focuses on computational methods on next generation sequencing and single-cell RNA-seq data analysis.

**Sumin Jo** is a PhD student in the Department of Electrical and Computer Engineering, the University of Texas at San Antonio. Her research focuses on m6A mRNA methylation and deep learning for biomedical applications.

**Karla Paniagua** is a PhD student in the Department of Electrical and Computer Engineering, the University of Texas at San Antonio. Her research focuses on

**Yu-Chiao Chiu**, PhD, is a postdoctoral fellow at the Greehey Children’s Cancer Research Institute at the University of Texas Health San Antonio. His postdoctoral research is focused on developing deep learning models for pharmacogenomic studies.

**Jianqiu Zhang, PhD, is an Associate Professor in** the Department of Electrical and Computer Engineering at the University of Texas at San Antonio. Her current research focuses on deep learning for biomedical applications such as m6A mRNA methylation.

**Shou-Jiang Gao, PhD, is a Professor** in UPMC Hillman Cancer Center and Department of Microbiology and Molecular Genetics, University of Pittsburgh. His current research interests include Kaposi’s sarcoma-associate herpesvirus (KSHV), AIDS-related malignancies,

translational and cancer therapeutics, and systems biology.

**Yufang Jin**, PhD, is a Professor in the Department of Electrical and Computer Engineering at the University of Texas at San Antonio. Her research focuses on mathematical modeling of cellular responses in immune systems, data-driven modeling and analysis of macrophage activations, and deep learning applications.

**Yidong Chen**, PhD, is a Professor in the Department of Population Health Sciences and the director of Computational Biology and Bioinformatics at Greehey Children’s Cancer Research Institute at the University of Texas Health San Antonio. His research interests include bioinformatics methods in next-generation sequencing technologies, integrative genomic data analysis, genetic data visualization and management, and machine learning in translational cancer research

**Yufei Huang**, PhD, is a Professor in UPMC Hillman Cancer Center and Department of Medicine, School of Medicine, University of Pittsburgh. His current research interests include uncovering the functions of m6A mRNA methylation, cancer virology, and medical AI & deep learning.

**Abstract**

Since its selection as the method of the year in 2013, single-cell technologies have become mature enough to provide answers to complex research questions. However, together with the growth of single-cell profiling technologies, there has also been an increase of computational challenges to process the generated datasets. It's here that by effectively leveraging large data sets, Deep Learning (DL) is positioning as the first option for single-cell analyses. Here we provide a unified mathematical description of the DL methods used in single cell RNA sequencing (scRNA-Seq) followed with the survey of the most representative published DL algorithms for scRNA-Seq in the field.

**Key points**:

* Single cell RNA sequencing technology generate large collection of transcriptomic profiles of up to millions of cells, enabling biological investigation of hidden structures or cell types, predicting their effects or responses to treatment more precisely, or utilizing subpopulation to address unanswered hypotheses.
* Current Deep Learning-based analysis approaches for single cell RNA seq data is systematically reviewed in this paper according to the challenge they address and their roles in the analysis pipeline.
* A unified mathematical description of the surveyed DL models is presented and the specific model features were discussed when reviewing each approach.
* A comprehensive summary of the evaluation metrics, comparision algorithms, and datasets by each approaches is presented.

**Keywords**: deep learning; single-cell RNA-seq; imputation; dimention reduction; clustering; batch correction; cell type identification; functional prediction; visualization

# Introduction

Single cell sequencing technology has been a rapid developing area to study genomics, transcriptomics, proteomics, metabolomics, and cellular interactions at the single cell level for cell type identification, tissue composition and reprogramming [1][2]. Specifically, sequencing of the transcriptome of single cells, or single-cell RNA-sequencing (scRNA-seq), has become the dominant technology in many frontier research areas such as disease progression, drug discovery, and host-virus interactions[3-5]. One particular area that scRNA-seq has made a tangible impact is cancer where scRNA-seq is becoming a powerful tool for the understanding of invasion, intratumor heterogeneity, metastasis, epigenetic alterations and to detect rare cancer stem cells. Currently, scRNA-seq is applied to develop personalized therapeutic strategies that are potentially useful in cancer diagnosis, therapy resistance during cancer progression, and the survival of patients [6, 7]. The scRNA-seq has also been adopted in the combat of COVID-19 to elucidate how the innate and adaptive host immune system miscommunicate to worsen the immunopathology produced during this viral infection [8].

All these studies have led to a massive amount of scRNA-seq data deposited to public databases, waiting for further computational analysis. Add one to two sentences to describe the databases. On the other hand, due to biological and technical factors, scRNA-seq data presents several analytical challenges related to its complex characteristics like missing expression values, high technical and biological variance, noise and sparse gene coverage, and elusive cell identities [1]. These characteristics make it difficult to directly apply commonly used bulk RNA-seq data analysis techniques and have called for novel statistical approaches for scRNA-seq data cleaning and computational algorithms for data inteperatation. To this end, specialized scRNA-seq analysis pipelines such as Seurat [9] and Scanpy[10], along with a large collection of task-specific tools, have been developed to address intricate technical and biological complexity of scRNA-seq data.

Recently, deep learning has demonstrated its significant advantages in natural language processing, speech and facial recognition with massive amount of data. Such advantages have iniated the application of DL in scRNA-seq data analysis as a competitive alternatives to conventional machine learning approaches for uncovering cell clustering [11], [12], cell type identification [13, 14], gene imputation [15-17], and batch correction [18] in scRNA-seq analysis. Compared to conventional ML approaches, DL is more powerful in capturing complex features of high-dimensional scRNA-seq data. It is also more versatile, where a single model can be trained to address multiple tasks or adapted and transferred to different tasks. Moreover, the DL training scales more favorably with the number of cells in scRNA-seq data size, making it particularly attractive for handling the ever-increasing volume of single cell data. Indeed, the growing body of DL-based tools has demonstrated DL’s exciting potential as a learning paradigm to significantly advance the tools we use to interrogate scRNA-seq data.

In this paper, we present a review of the recent advances of DL methods for solving the present challenges in scRNA-seq data analysis (Table 1). For this, we have surveyed deep learning papers published up to April 2021. In order to maintain high quality for this review, we choose not to include any (bio)archival papers, although a proportion of these manuscripts contain important new findings that would be published after completing their peer-reviewed process. Previous efforts to review the recent advances in machine learning methods focused on efficient integration of single cell data [19, 20], instead of applications of deep learning. To help understand the relationship of each surveyed model with the overall scRNA-seq analysis pipeline, we organize the surveys according to the challenge they address and discuss them following the analysis pipeline. A unified mathematical description of the surveyed DL models is presented and the specific model features were discussed when reviewing each method. This will help the readers see the modeling connections among the surveyed DL methods and recognizes the uniqueness of each model. Besides the models, we also review the metrics that these methods adopt to evaluate their performances and compile the information of the datasets used by these methods to demonstrate their advantages and utility. We envision that this survey will serve as an important information portal for learning the application of DL for scRNA-seq analysis and inspire innovative use of DL to address a broader range of new challenges in emerging multi-omics and spatial single-cell sequencing.

# 2. Overview of scRNA-seq processing pipeline

Various scRNA-seq techniques (like SMART-seq, Drop-seq and 10X genomics sequencing [21, 22] are available nowadays with their sets of advantages and disadvantages. However, the data content and processing steps of different scRNA-seq techniques are quite standard and conventional. A typical scRNA-seq dataset consists of three files: genes quantified (gene IDs), cells quantified (cellular barcode) and a count matrix (number of cells × number of genes), irrespective of the technology or pipeline used. Conventionall, a series of essential steps are applied to these matrices for different analysis objectives as illustrated **Fig. 1**. These steps are:

**Quality control** is the first and the key step to filter out dead cells, double-cells, or cells with failed chemistry or other technical artifacts. It is commonly determined using three QC covariates: the number of counts (count depth) per barcode (or per cell), the number of genes per barcode, and the fraction of counts from mitochondrial genes per barcode [23]. QC ensures that barcode sequences that identify each cell and UMIs that identify and allow counting for each RNA are present before generating the matrix [24].

![Graphical user interface

Description automatically generated]()

**Figure 1. Conventional single cell data analysis steps.** Depending on the input data and analysis objectives, major analysis steps are illustrated in the middle track, along with optional analysis modules below each analysis module. Deep learning algorithms, either specific task oriented or multi-functional methods, some are reviewed in the survey, are also illustrated above each corresponding module.

**Normalization** is designed to eliminate imbalanced sampling, cell differentiation, viability, and many other factors. Approaches tailored for scRNA-seq have been developed including the Bayesian-based method coupled with spike-in, or BASiCS [25], deconvolution approach, scran [26], and sctransfrom in Seurat where regularized Negative Binomial regression was proposed [27].

**Batch effect** is a common source of technical variation in high-throughput sequencing experiments due to the distinct people handled at different time, or produced by multiple sites of a cell-atlas consortium, a major challenge in scRNA-seq data analysis. Many methods have been developed for batch effect correction, such as through detection of mutual nearest neighbors (MNNs) [28], canonical correlation analysis (CCA) with Seurat [29], and Hormony algorithm through cell-type representation [30].

**Imputation** step is necessary to handle high sparsity data matrix, due to missing value or dropout in scRNA-seq data analysis. Several tools have been developed to “impute” zero values in scRNA-seq data, such as SCRABBLE [31], SAVER [32] and scImpute [33].

**Dimensionality reduction and visualization** are essential tools to represent biological meaningful variation and high dimensionality with significant reduced computational cost. Dimensionality reduction methods, such as PCA, is widely used in scRNA-seq data analysis to achieve that purpose. More advanced nonlinear approaches that preserve the topological structure and avoid overcrowding in lower dimension representation, such as LLE [34] (used in SLICER [35]), tSNE [36], and UMAP [37] have also been developed and adopted as standard in single-cell data visualization.

**Clustering analysis** is a key step to identify cell subpopulations or distinct cell types to unravel the extent of heterogeneity, and their associated cell-type specific markers. Unsupervised clustering is frequently used here to categorize cells into clusters by their similarity often taken the aforementioned dimensionality-reduced representations as input, such as community detection algorithm Louvain [38] and Leiden [39], or data-driven dimensionality reduction followed with k-Means cluster by SIMLR [40].

**Feature selection** is another important step in single-cell RNA-seq analysis is to select a subset of genes, or features, for cell type identification and functional enrichment of each cluster. This step is achieved by differential expression analysis designed for scRNA-seq, such as MAST that used linear model fitting and likelihood ratio testing [41]; SCDE that adopted a Bayesian approach with a Negative Binomial model for gene expression and Poisson process for dropouts [42], or DEsingle that utilized a Zero-Inflated Negative Binomial model to estimate the dropouts [43].

Besides these key steps, downstream analysis can include cell type identification, coexpression analysis, prediction of perturbation response, where DL has also been applied. Other advanced analyses including trajectory inference and velocity and pseudotime analysis are not discussed here because most of the approaches on these topics are non-DL based.

# Overview of common deep learning models for scRNA-seq analysis

Unsupervised learning is the key step in the scRNA-Seq analysis, including batch correction, dimension reduction, imputation, and clustering, which lend themselves naturally to unsupervised DL models including the variational autoencoder (VAE), the autoencoder (AE), or generative adversarial networks (GAN). Also, adversarial transfer learning has been applied for cell-type classification. We started our review by introducing the general formulations of VAE, AE, and GAN for scRNA-seq together with their training strategies. These general formulations facilitate understanding the methodologies used by different papers in developing their specific algorithms, enabling us to focus on the different features of each method and bring attention to their uniqueness and novelty.

Diagram

Description automatically generated

**Figure 2. Graphical models of the surveyed DL models including A)** Variational Autoencoder (VAE); **B)** Autoencoder (AE);and **C)** Generative Adversarial Network (GAN).

* 1. **Variational Autoencoder (VAEs) for scRNA-Seq data**

Let represent a vector of gene expression (UMI counts or normalized, log-transformed expression) of genes in cell , where denotes gene ’s expression, which is assumed to follow some distribution (e.g., zero-inflated negative binomial (ZINB) or Gaussian), where and are parameters of the distribution (e.g., mean, variance, or dispersion) (**Fig. 2A**). We consider the first parameter to be of particular interest (e.g., the mean counts) for the scRNA-seq analysis and is thus further modeled as a function of a -dimension latent variable and an observed variable (e.g., the batch ID) by a decoder neural network (Fig. 2A) as

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

where the th element of is and is a vector of decoder weights, represents a low-dimensional latent representation of gene expression and is used in all the works for visualization and clustering. For VAE, is commonly assumed to follow a multivariate standard normal prior, i.e., with being a identity matrix. The second parameter of is a nuisance parameter, which is assumed with a prior distribution and can be either estimated or marginalized in variational inference. Now define as the collection of the unknown model parameters. Then, and (1) together define the likelihood .

The goal of training or inference is to compute the maximum likelihood estimate of

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

where is the marginal likelihood, which is in general analytically intractable but can be lower-bounded by the evidence lower bound (ELBO) , expressed as

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

where is an approximate to the intractable posterior distribution . To make the variational inference tractable is assumed in most cases as a qmultivariate Gaussian

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

whose means and variances are given by an encoder network applied to and (**Fig. 2A**) as

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

where is the vector of the unknown decoder weights. Because of the approximation by in eq. (2) and the introduction of the decoder network in eq. (4), the model parameters to be estimated become **.** Optimization of in (2) is computed efficiently by stochastic optimization, where the gradient is calculated by backpropagation.

All the surveyed papers that deploy VAE follow this general modeling process. However, an alternative and more general formulation treats it as optimization with a loss function expressed as

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

where are additional function-specific losses introduced to constrain the model for different functions (clustering, cell type prediction, etc), and s are the Lagrange multipliers. With this general formulation, for each paper, we focus on the survey the specific choice of data distribution that defines , different designed for specific functions, and how the decoder and encoder are applied to model different aspects of scRNA-seq data.

* 1. **Autoencoders (AEs) for scRNA-seq data**

AEs have been proposed to learn the low dimensional latent representation of expression . The AE includes an encoder and a decoder (**Fig. 2B**) such that

|  |  |  |
| --- | --- | --- |
|  | ; | (7) |

where like VAE is the -dimension latent variable, are encoder and decoder weight parameters, and defines the parameters (e.g. mean) of data distribution and thus the likelihood (Fig. 1B). Note that the mean of is often considered as the imputed and denoised expression of . For most common AEs, assumes a Gaussian distribution and becomes the mean of the Gaussian and can be directly used as imputed, normalized gene expression. Nevertheless, additional designs can be introduced to attend imputation specifically. can also be negative binomial (NB) or ZINB as in DCA [23] to model the reads count directly with their parameters defined as functions of . Additional design can be included in the AE model for batch correction, clustering, and other functions.

The training of the AE is generally carried out by stochastic gradient descent algorithms to minimize the loss with the general expression similar to that of VAE in eq. (6)

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

where is , and s are additional function-specific losses. When is the Gaussian, becomes one of the most commonly used mean square error (MSE) loss

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

Because different AE models differ in their AE architectures and the loss functions, we will discuss the specific architecture and the loss functions for each reviewed model.

* 1. **Generative adversarial networks (GANs) for scRNA-seq data**

GANs have been used for imputation, data generation and augmentation of the scRNA-seq analysis. Without loss of generality, the GAN, when applied to scRNA-seq, is designed to learn to generate gene expression profiles from , the distribution of , the count or normalized expression vectors of the scRNA data. The vanilla GAN consists of two deep neural networks [2]. The first network is the generator with parameter , which is essentially a decoder that takes a noise vector from the distribution and a class label (e.g. cell type) as input and is trained to generate , a "fake" sample of a gene expression profile (Fig. 2C). Note that including a class label at the input is optional and when it is included, the model is known as the conditional GAN. The second network is the discriminator network with parameters , which is a classifier trained to distinguish between the "real" sample and fake data (Fig. 1C). The generator and discriminator are trained to outplay each other, resulting in a minimax game, in which is forced by to produce better samples, which, when converge, can fool the discriminator , thus becoming samples from . The vanilla GAN suffers heavily from training instability and mode collapsing [44]. To that end, Wasserstein GAN (WGAN) was developed to effectively improve the training stability and convergence [45]. The WGAN loss computes the Wasserstein distance, also called the earth moving distance between the real and fake sample distributions [46]:

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

Additional terms can also be added to (10) to constrain the functions of the generator. Training based on the WGAN loss in Eq. (10) amounts to a min-max optimization, which iterates between the discriminator and generator, where each optimization is achieved by a stochastic gradient descent algorithm through backpropagation. The WGAN requires to be K-Lipschitz continuous [44], which can be satisfied by adding the gradient penalty to the WGAN loss [47]. Once the training is done, the generator can be used to generate gene expression profiles of new cells.

1. **Overview of datasets and evaluation metrics**

A variety of datasets and performance evaluation metrics have been used to demonstrate the performance of the surveyed DL models for different tasks. We summarize these datasets and evaluation metrics in **Table 2** & 3. We detail the mathatical definition of the evaluation metrics in the following.

* 1. **Evaluation methods**

An extensive list of evaluation methods has been proposed for different scRNA-seq analysis tasks [18, 48, 49]. We provide an overview here the methods adopted in the surveyed papers. We discuss them according to the key categories on which the surveyed papers are organized, namely, imputation, batch effect correction, dimension reduction and clustering, cell type identification, and functional analysis.

* + 1. **Imputation**

The evaluation of the performance of imputation methods considers their ability to recover biological signals and improve downstream analyses. For this two main methods have been used. First is the evaluation of similarity between bulk and imputed scRNA-seq data.

Second is the evaluation of imputation on unsupervised clustering.

The first approach consist in assessing the similarity between bulk and imputed scRNA-seq data. For a given scRNA-seq dataset, the “pseudobulk”, or the average of normalized (log2-transformed) scRNA-seq counts across cells, is calculated first, and the Spearman’s rank correlation coefficient (SCC) between the pseudobulk and the bulk RNA-seq profile of the same cell type is evaluated. The statistical significance is assessed whether SCCs bewteen the bulk and pseudobulks from two imputation methods are equal.

The second approach consist in measuring the accuracy of several clustering assignments methods using four metrics:

* Entropy of accuracy (*Hacc*) and entropy of purity (*Hpur*). *Hacc* (*Hpur*) measures the diversity of the ground-truth labels (predicted cluster labels) within each predicted cluster group (ground-truth group), respectively.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

where *M* is the total number of predicted clusters from the clustering algorithm, *N* is the number of ground-truth clusters, *Mi,* (or *Ni*) is the number of predicted clusters (or ground-truth clusters) in the *i*th ground -truth cluster (or predicted cluster), respectively. *pi*(*xj*) (or *qi*(*xj*)) are the proportions of cells in the *j*th ground-truth cluster (or predicted cluster) relative to the total number of cells in the *i*th predicted cluster (or ground-truth clusters), respectively. A smaller value of *Hacc* means the cells in a predicted cluster are constantly labeled from the same ground-truth group, while A smaller value of *Hpur* means the cells in the ground-truth groups are homogeneous with the same predicted cluster labels [18]. However, smaller *Hacc* (or *Hpur*) can lead to over-clustering (or under-cluster), when each predicted cluster contains 1 cell (*Hacc* = 0) or all cells in one predicted cluster (*Hpur* = 0).

* Adjusted Rand index (ARI). Rand index (RI) is another measure of constancy between two clustering outcomes. If *a* (or *b*) is the count of number of pairs of cells in one cluster (or different clusters) from one clustering algorithm but also fall in the same cluster (or different clusters) from the other clustering algorithm, then, , where is the total number of pairs when given *n* cells. The *RI* has a value between 0 and 1, with 0 indicating that the two clustering algorithms do not agree on any pair of cells and 1 indicating that the two clustering algorithms are exactly the same. ARI is a corrected-for-chance version of *RI*, or

|  |  |  |
| --- | --- | --- |
|  |  |  |

where *E*[*RI*] is the expected Rand Index [50].

* Median Silhouette index. The Silhouette index is defined as

|  |  |  |
| --- | --- | --- |
|  |  |  |

where *a*(*i*) is the average dissimilarity of *i*th cell to all other cells in the same cluster, and *b*(*i*) is the average dissimilarity of *i*th cell to all cells in the closest cluster. The range of *s*(*i*) is [−1,1], with 1 to be well-clustered with appropriate labels, and -1 to be completely misclassified. *s*(*i*) = 0 indiates the cell could be assigned to nearest clusters (or overlapping clusters).

A good imputation method should allow perform downstream (clustering) analyses without introducing any artifacts or false signals.

* + 1. **Batch effect correction**

When evaluating the performance of a batch correction method, we need to consider how well it mixes the shared cell types between different batches and at the same time identifies batch-specific cells. The existing metrics can be classified as cluster-level and cell-level metrics. Cluster level metrics are those used for evaluating clustering performance and include adjusted rand index (ARI), normalized mutual information (NMI), and silhouette coefficients. They are easy to compute but do not measure local mixture of cells from different batches. This drawback is addressed by the cell-level metrics, which includes k-nearest neighbor batch-effect test (kBET), local inverse Simpson’s index (LISI), and classifier-based. Because the cluster-level metrics will be discussed in detail in Section 4.2.3, we focus on discussing cell-level metrics in this section.

**Entropy of mixing.** This metric evaluates the mixing of cells from different batches in the neighborhood of each cell [28]. It first randomly sample *N* cells and then for each cell it calculates the regional entropy of mixing as

|  |  |  |
| --- | --- | --- |
|  |  |  |

where is the number of batches and is the proportion of cells from batch among *N* nearest cells (e.g. *N* = 100). The total entropy is the sum of reginal entropies. The computation repeats *K* times to obtain an empirical distribution of the entropy of mixing.

**Maximum Mean Discrepancy** (MMD) is a non-parametric distance between distributions based on the reproducing kernel Hilbert space (RKHS) [51], or, MMD is a distance-based measure between two distribution *p* and *q* based on the mean embeddings *mp* and *mq* in a reproducing kernel Hilbert space *F*,

|  |  |  |
| --- | --- | --- |
|  |  |  |

MMD will vanish for most finite samples *xk* and *yk* only if two distributions are the same.

**k-Nearest neighbor batch-effect test (kBET).** kBET assesses the batch mixing by comparing the batch-specific distribution within *k*-nearest neighbors (kNNs) of a cell with the global distribution of batches [52]. It uses a -based test for random neighborhoods of fixed size to determine whether they are well mixed. Given a dataset of cells from batches with denoting the number of cells in batch . Under the null hypothesis that cells are ‘well mixed’, that is the absence of batch effect, we have the test statistics as

|  |  |  |
| --- | --- | --- |
|  |  |  |

where is the number of cells from batch in the *k*-nearest neighbors of cell , *fl* is the global fraction of cells in batch , or , and denotes the distribution with degrees of freedom. The averaged rejection rate of the test for all cells is used to define the performance of a batch correction method.

**Local Inverse Simpson’s Index (LISI).**  Like kBET, LISI also compares the batch mixing at local level with global batch distribution. However, unlike kBET, which is agnostic of cell types, LISI requires well mixing of cells from the same cell type but not of those from different types [30]. LISI evaluates cell-type-specific mixing using an inverse Simpson’s Index in a local neighborhood of each cell. LISI builds Gaussian kernel-based distributions of local neighborhoods sensitive to local diversity. It calculates inverse Simpson’s Index in the *k*-nearest neighbors of cell for all batches as , where denotes the proportion of batch in the *k*-nearest neighbors. The score reports the effective number of batches in the *k*-nearest neighbors of cell . Inverse Simpson’s Index in the *k*-nearest neighbors of cell can also be calculated to evaluate the diversity of different cell types. However, in an ideal case, LISI score should be 1, reflecting a separation of unique cell types.

**Classifier-based.**  Although LISI addresses the issue of cell-type proportion of different batches but it is hard to summarize all single cell-level LISI scores into a simple statistic for comparing across different methods [15]. The classifier-based approach addresses this issue by using two distinct local classifiers for each single cell. The first classifier classifies every single cell as positive and negative cells. A single cell is positive if at least 50% cells of its k-nearest neighbor (KNN) cells are from the same cell-type label, otherwise ‘negative’. The positive cells are further classified into true and false positive cells, where true positive cells are those surrounded by appropriate portions of cells with batches. In other words, if we sample cells from this cell-type cluster, the expected number of cells from batch will be *k⋅ fl*, where *fl* is the global fraction of cells in batch *l*. A positive cell in this cluster is a true positive when the observed cell numbers for each batch among its neighbors are within 3 standard deviations around the expected numbers. The proportions of positive cells and true positive cells are used as the summary metrics to evaluate the overall performance of batch effect removal. The higher the proportions, the better the algorithm.

* + 1. **Clustering**

Evaluating the performance of clustering algorithms is not as trivial as counting the number of errors like supervised learning. In general, the clustering performance evaluation metric should not just take absolute corrected labelled cells into account but also consider if the clustering defines a good similarity or separation in the dataset compared to ground truth. When ground truth is not available, evaluation must be performed using model itself such as clustering distance, dispersion, etc. Similar measures, such as Adjusted Rand Index (ARI) and Silhouette Index discussed in Section 4.2.1 can also be employed here to measure the agreement between predicted assignment to the ground-truth assignment.

**Normalized Mutual Information (NMI).** The mutual information(MI) [47] is a measure of mutual dependency between two cluster assignments *U* and *V*. It quantifies the amount of information we could have about one assignment by observing the other assignment. For samples, we have the entropy for cluster assignments *U* and *V* as

|  |  |  |
| --- | --- | --- |
|  |  |  |

where and . Also, define the joint distribution probability is . Then, the mutual information of *U* and *V* is defined as

|  |  |  |
| --- | --- | --- |
|  |  |  |

The NMI is a normalization of the MI score between 0 and 1. For example, the average NMI is defined as [53]

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Homogeneity, Completeness, and V-Measure.** The homogeneity score (HS) measures the extent to which its clusters contain only samples that belong to one cell type, or , where *H*() is the entropy, and *U* is the ground-truth assignment and *V* is the predicted assignment. The *HS* range from 0 to 1, where 1 indicates perfectly homogeneous labelling. Similarly, the completeness score (CS) is defined as , Its values range from 0 to 1, where 1 indicates all member from a ground-truth label are assigned to a single cluster.

The V-Measure [54] is the harmonic mean between *HS* and *CS*, defined as , where *b* indicates the weight of *HS*. V-Measure is a more symmetric, i.e. switching the true and predicted cluster labels does not change V-Measure.

1. **Survey of deep learning models for scRNA-Seq analysis**

In this section, we survey the DL models for scRNA-seq analysis. To help the reader understand the relationship between the problems that each surveyed work addresses and the key challenges in the general scRNA-seq processing pipeline, we divide the survey into sections according to general functions in the scRNA-seq processing pipeline illustrated in **Fig. 1**. For each paper, we describe the model details under the general model framework introduced in Section 3 and discuss the specific loss function. We also survey the evaluation metrics and summarize results. To facilitate cross-reference of the information, w and tabulate the datasets and evaluation metrics used in each paper in **Table 2 &3.**

* 1. **Imputation**

The goal of imputation is to estimate the missing gene expression values due to dropout, or the failure to amplify the original RNA transcripts. These missing expression values can affect downstream bioinformatics analysis significantly as it decreases the power of the studies and introduces biases in gene expression [17]. VAE, AE, and GAN have been applied for imputation and we review their specific model designs in this section.

* + 1. **DCA:** **deep count autoencoder**

DCA [15] is an AE designed for imputation (**Fig. 2B**). DCA is implemented in Python as a command line and also integrated into the Scanpy framework.

*Model*. DCA models UMI counts of a cell with missing values using the ZINB distribution as

|  |  |
| --- | --- |
| , for ; | (11) |

where is a Dirac delta function, denotes the negative binomial distribution and are dropout rate, mean, and dispersion, which are functions of DCA decoder output as

|  |  |
| --- | --- |
|  | (12) |

where the th element of ,, and are and , respectively and **, ,** and areadditional weights to be estimated. The DCA encoder and decoder follow the general AE formulation as in Eq. (7) but the encoder takes the size factor normalized, log-transformed expression as input. The encoder and decoder architecture are the conventional deep neural networks. The parameters to be trained are .To train the model, DCA uses a constrained log-likelihood as the loss function as

|  |  |  |
| --- | --- | --- |
|  |  | (13) |

where the second term functions as a ridge prior on the dropout probabilities. Once the DCA is trained, the mean counts are used as the denoised and imputed counts for cell .

*Evaluation metrics.*A Density Resampled Estimate of Mutual Information (DREMI) measure was adapted for the higher dimensionality and sparsity of scRNA-seq data.

*Results*. For evaluation, DCA is compared to other methods using simulated data (using Splatter R package), and a real bulk transcriptomics data from a developmental *C. elegans* time-course experiment was used with added simulating single-cell specific noise. For this. gene expression was measured from 206 developmentally synchronized young adults over a twelve-hour period (C elegans). Single-cell specific noise was added in silico by genewise subtracting values drawn from the exponential distribution such that 80% of values were zeros. The paper analyzed the Bulk contains less noise than single-cell transcriptomics data and can thus aid in evaluating single-cell denoising methods by providing a good ground truth model. The authors also did a comparison of other methods like SAVER [32], scImpute [33], and MAGIC[55]. DCA denoising recovered original time-course gene expression pattern while removing single-cell specific noise. Overall, DCA demonstrates the strongest recovery of the top 500 genes most strongly associated with development in the original data without noise; is shown to outperform other existing methods in capturing cell population structure in real data (using PBMC, CITE-seq, runtime scales linearly with the number of cells.

* + 1. **SAVER-X: single-cell analysis via expression recovery harnessing external data**

SAVER-X [56] is an AE model developed to denoise and impute scRNA-seq data with transfer learning from other data resources. SAVER-X is implemented in R with the support of Python package sctransfer.

*Model*. SAVER-X decomposes the variation in the observed counts with missing values into three components: i) predictable structured component that represents the shared variation across genes, ii) unpredictable cell-level biological variation and gene-specific dispersions, and iii) technical noise. Specifically, is modeled as a Poisson-Gamma hierarchical model,

|  |  |
| --- | --- |
| , | (14) |

where is the sequencing depth of cell *n*, is the mean, and is the dispersion. This Poisson-Gamma mixture is an equivalent expression to the NB distribution. As a result, the ZINB distribution as Eq. (11) in DCA is also adopted to model missing values.

To train the model, a similar log-likelihood function as Eq. (13) in DCA is used as the loss function. However, is initially learned by an AE pre-trained using external datasets from an identical or similar tissue and then transferred to and updated using data to be denoised. Such transfer learning can be applied to data between species (e.g., human and mouse in the study), cell types, batches, and single-cell profiling technologies. After is inferred, SAVER-X generates the final denoised data by an empirical Bayesian shrinkage.

*Evaluation metrics.* *t*-SNE visualization and ARI were used to evaluate the clustering performance after imputation. Pearson correlation coefficients between protein and corresponding gene expression levels after denoising CITE-seq data were computed.

*Results.* SAVER-X was applied to multiple human single-cell datasets of different scenarios: i) T-cell subtypes, ii) a cell type (CD4+ tegulatory T cells) that was absent from the pretraining dataset, iii) gene-protein correlations of CITE-seq data, and iv) immune cells of primary breast cancer samples with a pretraining on normal immune cells. SAVER-X with pretraining on HCA and/or PBMCs outperformed the same model without pretraining and other denoising methods, including DCA [23], scVI[13], scImpute [33], and MAGIC [55]. The model achieved promising results even for genes with very low UMI counts. SAVER-X was also applied for a cross-species study in which the model was pretrained on a human or mouse dataset and transferred to denoise another. The results demonstrated the merit of transferring public data resources to denoise in-house scRNA-seq data even when the study species, cell types, or single-cell profiling technologies are different.

* + 1. **DeepImpute (Deep neural network Imputation**)

DeepImpute [17] is a deep neural network model that imputes genes in a divide-and-conquer approach. DeepImpute implemented in Keras framework/TenorFlow environment.

*Model.* For each dataset, DeepImpute selects to impute a list of genes or highly variable genes (variance over mean ratio, default = 0.5). Each sub-neural network aims to understand the relationship between the input genes (input layer) and a subset of target genes (output layer). Genes are first divided into random subsets of size 512 called target genes. For each subset, a neural network of four layers (input, dense, dropout and output layers) is trained where the input layer includes genes (predictor genes) who are among top 5 best correlated genes to target genes but not part of the target genes in the subset. The loss is defined as the weighted MSE

|  |  |  |
| --- | --- | --- |
|  |  | (15) |

This function gives higher weights to genes with higher expression values, thus emphasizing accuracy on high confidence values and avoiding over penalizing genes with extremely low values.

*Evaluation metrics.* DeepImpute computes mean squared error (MSE) and Pearson's correlation coefficient between imputed and true expression.

*Result.* DeepImpute had the highest overall accuracy and offered faster computation time with less demand on computer memory compared to other methods like MAGIC, DrImpute, ScImpute, SAVER, VIPER, and DCA. Using simulated and experimental datasets (**Table 2**), DeepImpute showed benefits in increasing clustering results and identifying significantly differentially expressed genes. DeepImpute and DCA, show overall advantages over other methods and between which DeepImpute performs even better. The properties of DeepImpute contribute to its superior performance include 1) a divide-and-conquer approach which contrary to an autoencoder as implemented in DCA, resulting in a lower complexity in each sub-model and stabilizing neural networks, and 2) the subnetworks are trained without using the target genes as the input which reduces overfitting while enforcing the network to understand true relationships between genes.

* + 1. **LATE: Learning with AuToEncoder**

LATE [57] is an AE whose encoder takes the log-transformed expression as input. LATE implemented in Python with TensorFlow.

*Model*. LATE sets zeros for all missing values at the input and generates the imputed expressions at the decoder's output. LATE experimented with three different network architectures composed of 1, 3 and 5 hidden layers. LATE minimizes the MSE loss as defined in Eq. (9). One problem with this model is that it assumes that all the zeros in the scRNA-seq data are missing values but some zeros could be real and reflect the actual lack of expression.

*Evaluation metrics.* Like DeepImpute, LATE used MSE to evaluate the performance.

*Result.* Using synthetic data generated from pre-imputed data followed with random dropout selection at different degree, LATE is shown to outperform other existing methods like MAGIC, SAVER, DCA, scVI, particularly when the ground truth contains only a few or no zeros. However, when the data contain many zero expression values, DCA achieved a lower MSE than LATE, although LATE still has a smaller MSE than scVI. This result suggests that DCA likely does a better job identifying true zeros gene expression, partly due to that LATE does not make assumptions on the statistical distributions of the single-cell data that potentially have inflated zero counts.

* + 1. **scGMAI**

Technically, scGMAI [58] is a model for clustering but it includes an AE in the first step to combat dropout. The scGAMI’s AE model is implemented with TensorFlow.

*Model.* To impute the missing values, scGMAI applies an AE like LATE to reconstruct log-transformed expressions with dropout. One difference is that it chooses Softplus as the activation function since it is smoother than ReLU and thus more suitable for scRNA-seq data. The MSE loss as in (9) is adopted.

After imputation, scGMAI uses fast independent component analysis (ICA) on the AE reconstructed expression to reduce the dimension and then applies a Gaussian mixture model on the ICA reduced data to perform the clustering.

*Evaluation metrics.* It used clustering metrics including NMI, ARI, Homogeneity, and Completeness to evaluate the performance.

*Results.* To assess the performance, the AE in scGMAI was replaced by five other imputation methods including SAVER [32], MAGIC [55], DCA [23], scImpute [33], and CIDR[59]. A scGMAI implementation without AE was also compared. Seventeen scRNA-seq data (part of them are listed in **Table 2** as marked) were used to evaluate cell clustering performances. The results indicated that the AEs significantly improved the clustering performance in eight of seventeen scRNA-Seq datasets.

* + 1. **scIGANs**

Imputation approaches based on information from cells with similar expressions suffer from oversmoothing, especially for rare cell types. scIGANs [16] is a GAN-based imputation algorithm, which overcomes this problem by using the observed samples with missing values to train a GAN to generate samples with imputed expressions.

*Model.* The gene expression vector is first reshaped into a square image-like format and scIGAN takes the reshaped data as input. The model follows a BEGAN [60] framework, which substitutes the generator with an autoencoder that includes an encoder and a decoder and also replaces the discriminator with a function that computes the reconstruction error of the autoencoder (e.g. MSE). Then, the Wasserstein distance between the reconstruction errors of the real and generated samples are computed as the loss

|  |  |
| --- | --- |
|  | (16) |

The encoder and decoder are trained in a GAN fashion to minimize this Wasserstein distance. This framework forces the model to meet two computing objectives, i.e. reconstructing the real samples and discriminating between real and generated samples. Proportional Control Theory was applied to balance these two goals during the training.

After training, the decoder is used to generate new samples of a specific cell type. Then, the k-nearest neighbors (KNN) approach is applied to the real and generated samples to impute the real samples' missing expressions.

*Evaluation metrics*. It used a variety of clustering- and classification-based metrics including ARI, ACC, AUC and F-score.

*Results.* scIGANs was first tested on simulated samples with different dropout rates. Performance of rescuing the correct clusters was compared with 11 existing imputation approaches including DCA, DeepImpute, SAVER, scImpute, MAGIC, etc. scIGANs reported the best performance for all metrics. scIGAN was next evaluated for correctly clustering cell types on the Human brain scRNA-seq data and showed superior performance than existing methods again. scIGANs was next evaluated for identifying cell-cycle states using scRNA-seq datasets from mouse embryonic stem cells. The results showed that scIGANs outperformed competing existing approaches for recovering subcellular states of cell cycle dynamics. scIGANs was further shown to improve the identification of differentially expressed genes and enhance the inference of cellular trajectory using time-course scRNA-seq data from the differentiation from H1 ESC to definitive endoderm cells (DEC). Finally, scIGAN was also shown to scale to scRNA-seq methods and data sizes.

* 1. **Batch effect correction**

The goal of batch correction is to correct effects between data sets generated at different times or in different laboratories. These often occur due to uncontrolled variability in experimental factors, e.g., reagent quality, operator skill or even controlled variability like sequencing depth. The presence of batch effects can interfere with downstream analyses if they are not explicitly modeled.

* + 1. **BERMUDA: Batch Effect ReMoval Using Deep Autoencoders**

BERMUDA [61] deploys a transfer-learning method to remove the batch effect. It performs correction to the shared cell clusters among batches and therefore preserves batch-specific cell populations.

*Model.* BERMUDA has a conventional AE architecture that takes normalized, log-transformed expression of a cell as input. It has the general loss function but consists of two parts as

|  |  |  |
| --- | --- | --- |
|  |  | (17) |

where is the MSE reconstruction loss as defined in Eq. (9) and is the maximum mean discrepancy (MMD) loss that measures the differences in distributions among similar cell clusters in different batches. MMD is a non-parametric distance between distributions based on the reproducing kernel Hilbert space (RKHS) [51]. Instead of applying the MMD loss on batches entirely, BERMUDA considers the loss only between pairs of similar cell clusters shared among batches, where the MMD loss is defined as:

|  |  |
| --- | --- |
| ) | (18) |

where is the latent variable of **,** the input expression profile of a cell from cluster of batch , is if cluster of batch and cluster of batch are determined to be similar by MetaNeighbor [62] and , otherwise. equals zero when the underlying distributions of the observed samples are the same. By minimizing the MMD loss between the distributions of the latent variables of similar clusters, BERMUDA can be trained to remove batch effects in its latent variables.

*Evaluation metrics.* Evaluation of BERMUDA included three metrics: KBET, entropy of mixing, and silhouette index.

*Results.* BERMUDA was shown to outperform other methods like mnnCorrect [28], BBKNN[63], Seurat [9], and scVI [13] in removing batch effects on simulated and human pancreas data while preserving batch-specific biological signals. BERMUDA provides several improvements compared to existing methods: 1) capable of removing batch effects even when the cell population compositions across different batches are vastly different; and 2) preserving batch-specific biological signals through transfer-learning which enables discovering new information that might be hard to extract by analyzing each batch individually.

* + 1. **DESC: batch correction based on clustering**

DESC [64] is an AE that removes batch effect through clustering with hypothesis that batch differences in expressions are smaller than true biological variations between cell types, and, therefore, properly performing clustering on cells across multiple batches can remove batch effects without the need to define batches explicitly.

*Model.* DESC has a conventional AE architecture. Its encoder takes normalized, log-transformed expression as the input and uses decoder output as the reconstructed gene expression, which is equivalent to a Gaussian data distribution with being the mean. The loss function is

|  |  |  |
| --- | --- | --- |
|  |  | (19) |

where is reconstruction MSE as defined in Eq. (8) and is the clustering loss that regularizes the learned feature representations to form clusters. The clustering loss follows the design in the deep embedded clustering [65] . Let be the cluster index of a cell and assume that there are *K* clusters. Then, the clustering loss is defined as a Kullback–Leibler (KL) divergence

|  |  |  |
| --- | --- | --- |
|  |  | (20) |

where is the probability of cell belonging to cluster and computed as

|  |  |  |
| --- | --- | --- |
|  |  | (21) |

where is the center of cluster and is the target distribution calculated by normalizing with frequency per cluster

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | , and |  | (22) |

A standard *k*-means clustering algorithm is used to initialize cluster centers. The model is first trained to minimize only to obtain the initial weights before the model is trained to optimize the combined loss (19). When the training is done, each cell's cluster index can be assignedbased on . After the training, each cell is assigned with a cluster ID.

*Evaluation metrics*. ARI was used for clustering and the KL divergence (Eq. 20) was used to assess the batch effect removal.

*Results.* DESC was applied to the macaque retina dataset, which includes animal level, region level, and sample-level batch effects. The results showed that DESC is effective in removing the batch effect, whereas CCA [29], MNN [28], Seurat 3.0 [9], scVI [13], BERMUDA [61], and scanorama [66] were all sensitive to batch definitions. DESC was then applied to human pancreas datasets to test its ability to remove batch effects from multiple scRNA-seq platforms and yielded the highest ARI among the comparing approaches mentioned above. When applied to human PBMC data with interferon-beta stimulation, where biological variations are compounded by batch effect, DESC was shown to be the best in removing batch effect while preserving biological variations. DESC was also shown to remove batch effect for the monocytes and mouse bone marrow data and DESC was shown to preserve the pseudotemporal structure. Finally, DESC scales linearly with the number of cells and its running time is not affected by the increasing number of batches.

* + 1. **iMAP: Integration of Multiple single-cell datasets by Adversarial Paired-style transfer networks**

iMAP [67] combines AE and GAN for batch effect removal. It is designed to remove batch biases while preserving dataset-specific biological variations.

*Model.* iMAP consists of two processing stages, each including a separate DL model. In the first stage, a special AE, whose decoder combines the output of two separate decoders and , is trained such that

|  |  |  |
| --- | --- | --- |
|  | ; ) | (23) |

where is the one-hot encoded batch number of cell . can be understood as decoding the batch noise, whereas reconstructs batch-removed expression from the latent variable . The training minimizes the loss

|  |  |  |
| --- | --- | --- |
|  |  | (24) |

where is the MSE reconstruction loss defined in eq. (7) and is the content loss

|  |  |  |
| --- | --- | --- |
|  |  | (25) |

where is a random batch number. Minimizing further ensures the reconstructed expression would be batch agnostic and has the same content as .

However, the author indicated that due to the limitation of AE, this step is still insufficient for batch removal. Therefore, a second stage is included to apply a GAN model to make expression distributions of the shared cell type across different baches indistinguishable. To identified the shared cell types, a mutual nearest neighbors (MNN) strategy adapted from [28] was developed to identify MNN pairs across batches using batch effect independent as opposed to . Then, a mapping generator is trained using MNN pairs based on GAN such that , where and arethe MNN pairs from batch and an anchor batch . The WGAN-GP loss as in Eq. (10) was adopted for the GAN training. After training, is applied to all cells of a batch to generate batch-corrected expression.

*Evaluation matrics.* The classifier-based metric as described in section 4.2.2 was used.

*Results*: iMAP was first tested on benchmark datasets from human dendritic cells and Jurkat and 293T cell lines and then human pancreas datasets from five different platforms. All the datasets contain both batch-specific cells and batch-shared cell types. iMAP was shown to separate the batch-specific cell types but mix batch shared cell types and outperformed 9 other existing batch correction methods including Harmony, scVI, fastMNN, Seurat, etc. iMAP was then applied to the large-scale Tabula Muris datasets containing over 100K cells sequenced from two platforms. iMAP could not only reliably integrate cells from the same tissues but identify cells from platform-specific tissues. Finally, iMAP was applied to datasets of tumor-infiltrating immune cells and shown to reduce the dropout ratio and the percentage of ribosomal genes and noncoding RNAs, thus improving detection of rare cell types and ligand-receptor interactions. iMAP scales with the number of cells, showing minimal time cost increase after the number of cells exceeds thousands. Its performance is also robust against model hyperparameters.

* 1. **Dimension reduction, latent representation, clustering, and data augmentation**

Dimension reduction is indispensable for many type of scRNA-seq data analysis, considering the limited number of cell types in each biospecimen. Furthermore, biological processes of interests often involve the complex coordination of many genes, therefore, latent representation which capture biological variation in reduced dimentions are useful in interpreting many experiment conditions. In addition, many deep learning models further exploit latent dimentions and generative factors to produce augmented data that may in return to enhance the clustering, e.g., due to low representation of certain cell types. Therefore, we categories all these algorithms together in this section.

* + 1. **Dimension reduction by AEs with gene-interaction constrained architecture**

This study [68] considers AEs for learning the low-dimensional representation and specifically explores the benefit of incorporating prior biological knowledge of gene-gene interactions to regularize the AE network architecture.

*Model.* Several AE models with single or two hidden layers that incorporate gene interactions reflecting transcription factor (TF) regulations and protein-protein interactions (PPIs) are implemented. The models take normalized, log-transformed expressions and follow the general AE structure, including dimension-reducing and reconstructing layers, but the network architectures are not symmetrical. Specifically, gene interactions are incorporated such that each node of the first hidden layer represented a TF or a protein in the PPI; only genes that are targeted by TFs or involved in the PPI were connected to the node. Thus, the corresponding weights of and are set to be trainable and otherwise fixed at zero throughout the training process. Both unsupervised (AE-like) and supervised (cell-type label) learning were studied.

*Evaluation metrics.* Performance of cells clustering was evaluated by six metrics including NMI, ARI, completeness, Fowlkes–Mallows score [69], homogeneity, and v-measure [44]. Performance of cell-type retrieval was evaluated by the mean of average precision.

*Results.* Regularizing encoder connections with TF and PPI information considerably reduced the model complexity by almost 90% (7.5-7.6M to 1.0-1.1M). The clusters formed on the data representations learned from the models with or without TF and PPI information were compared to those from PCA, NMF, independent component analysis (ICA), t-SNE, and SIMLR [40]. The model with TF/PPI information and 2 hidden layers achieved the best performance by five of the six measures (0.87-0.92) and the best average performance (0.90). In terms of the cell-type retrieval of single cells, the encoder models with and without TF/PPI information achieved the best performance in 4 and 3 cell types, respectively. PCA yielded the best performance in only 2 cell types. The DNN model with TF/PPI information and 2 hidden layers again achieved the best average performance (mean of average precision, 0.58) across all cell types. In summary, this study demonstrated a biologically meaningful way to regularize AEs by the prior biological knowledge for learning the representation of scRNA-Seq data for cell clustering and retrieval.

* + 1. **Dhaka: a VAE-based dimension reduction model.**

Dhaka [70] was proposed to reduce the dimension of scRNA-Seq data for efficient stratification of tumor subpopulations.

*Model.* Dhaka adopts a general VAE formulation. It takes the normalized, log-transformed expressions of a cell as input and outputs the low-dimensional representation.

*Evaluation matrics.* ARI was used to determine the quality of the resulting clustering for each dimensionality reduction method. Spearman rank correlation was assessed to the scoring metric (lineage or differentiation) to contrast with other programs.

*Result.* Dhaka was first tested on the simulated dataset. The simulated dataset contains 500 cells, each including 3K genes, clustered into 5 different clusters with 100 cells each. The clustering performance was compared with other methods including t-SNE, PCA, SIMLR, NMF, an autoencoder, MAGIC, and scVI. Dhaka was shown to have an ARI higher than most other comparing methods. Dhaka was then applied to the Oligodendroglioma data and could separate malignant cells from non-malignant microglia/macrophage cells. It also uncovered the shared glial lineage and differentially expressed genes for the lineages. Dhaka was also applied to the Glioblastoma data and revealed an evolutionary trajectory of the malignant cells where cells were gradually progressing from a stemlike state to a more differentiated state. In contrast, other methods failed to capture this underlying structure. Dhaka was next applied to the Melanoma cancer dataset [71] and uncovered two distinct clusters that showed the intra-tumor heterogeneity of the Melanoma samples. Dhaka was finally applied to copy number variation data [72] and shown to identify one major and one minor cell clusters, of which other methods could not find.

* + 1. **scvis: a VAE for capturing low-dimensional structures**

scvis [73] is a VAE network that learns the low-dimensional representations capture both local and global neighboring structures in scRNA-Seq data.

*Model:* scvis adopts the generic VAE formulation described in section 3.1. However, it has a unique loss function defined as

|  |  |  |
| --- | --- | --- |
|  |  | (26) |

where is ELBO as in Eq. (3) and is a regularizer using non-symmetrized t-SNE objective function [73], which is defined as

|  |  |  |
| --- | --- | --- |
|  |  | (27) |

where and are two different cells, measures the local cell relationship in the data space, and measures such relationship in the latent space as

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | (28) |

with defined as the perplexity [74]. Because t-SNE algorithm preserves the local structure of high dimensional space after projecting to the lower dimension, promotes the learning of local structures of cells.

*Evaluation matrics.* KNN preservation and log-likelihood of low dimensional mapping are used to evaluate model performance.

*Results.* scvis was tested on the simulated data and outperformed t-SNE in a nine-dimensional space task. scvis preserved both local structure and global structure. The relative positions of all clusters were well kept but outliers were scattered around clusters. Using simulated data and comparing to t-SNE, scvis generally produced consistent and better patterns among different runs while t-SNE could not. scvis also presented good results on adding new data to an existing embedding, with median accuracy on new data at 98.1% for K= 5 and 94.8% for *K*= 65, when train *K* cluster on original data then test the classifier on new generated sample points. scvis was subsequently tested on four real datasets including metastatic melanoma, oligodendroglioma, mouse bipolar and mouse retina datasets. In each dataset, scvis was showed to preserve both the global and local structure of the data.

* + 1. **scVAE: VAE for single-cell gene expression data**

scVAE [75] includes multiple VAE models for denoising gene expression levels and learning the low-dimensional latent representation of cells. It investigates different choices of the likelihood functions in the VAE model to model different data sets.

*Model.* scVAE is a conventional fully connected network. However, different distributions have been discussed for to model different data behaviors. Specifically, scVAE considers Poisson, constrained Poisson, and negative binomial distributions for count data, piece-wise categorical Poisson for data including both high and low counts, and zero-inflated version of these distributions to model missing values. To model multiple modes in cell expressions, a Gaussian mixture is also considered for , resulting a GMVAE. The inference process still follows that of a VAE as discussed in section 3.1.

*Evaluation metrics.* ARI was used to assess the performance.

*Results.* scVAEs were evaluated on the PBMC data and compared with factor analysis (FA) models. The results showed that GMVAE with negative binomial distribution achieved the highest lower bound and ARI. Zero-inflated Poisson distribution performed the second best. All scVAE models outperformed the baseline linear factor analysis model, which suggested that a non-linear model is needed to capture single-cell genomic features. GMVAE was also compared with Seurat and shown to perform better using the withheld data. However, scVAE performed no better than scVI [13] or scvis [73], both are VAE models.

* + 1. **VASC: VAE for scRNA-seq**

VASC [76] is another VAE for dimension reduction and latent representation but it models dropout.

*Model:* VASC’s input is the log-transformed expression but rescaled in the range [0,1]. A dropout layer (dropout rate of 0.5) is added after the input layer to force subsequent layers to learn to avoid dropout noise. The encoder network has three layers fully connected and the first layer uses linear activation, which acts like an embedded PCA transformation. The next two layers use the ReLU activation, which ensures a sparse and stable output. This model's novelty is the zero-inflation layer (ZI layer), which is added after the decoder to model scRNA-seq dropout events. The probability of dropout event is defined as where is the recovered expression value obtained by the decoder network. Since backpropagation cannot deal with stochastic network with categorical variables, a Gumbel-softmax distribution [77] is introduced to address the difficulty of ZI layer. The loss function of the model takes the form , where *L*0 is the binary entropy because of the input is scaled to [0 1], and a loss performed using KL divergence *KL*(*Q*(*z*|*x*)||*P*(*z*)), where *z* is the latent variables (Gaussian distribution). After the model is well trained, the latent code can be used as the dimension-reduced features for downstream tasks and visualization.

*Evaluation metrics.* Four measures are used to assess the performance including NMI, ARI, homogeneity, and completeness.

*Results.* VASC was compared with PCA, t-SNE, ZIFA, and SIMLR on 20 datasets. In the study of embryonic development from zygote to blast cells, all methods roughly re-established the development stages of different cell types in the dimension-reduced space. However, VASC showed better performance to model embryo developmental progression. In the Goolam, Biase and Yan datasets, scRNA-seq data were generated through embryonic development stages from zygote to blast, VASC re-established development stage from 1, 2, 4, 8, 16 to blast, while other methods failed. In the Pollen, Kolodziejczyk ,and Baron dataset, VASC formed appropriate cluster, either with homogeneous cell type, preserved proper relative postions, or minimal batch influerence. Interestingly, tested on the PBMC dataset, VASC showed to identify the major global structure (B cells, CD4+, CD8+ T cells, NK cells, Dendritic cells), it also detected subtle differences within monocytes (FCGR3A+ vs CD14+ monocytes), indicating the capability of VASC handling large number of cells or cell types. Quantitative clustering performance in NMI, ARI, homogeneity and completeness was also performed. VASC always ranked top two in all the datasets. In terms of NMI and ARI, VASC best performed on 15 and 17 out of 20 datasets, respectively.

* + 1. **scDeepCluster**

scDeepCluster [78] is an AE network that simultaneously learns feature representation and performs clustering via explicit modeling of cell clustersas in DESC.

*Model:* Similar to DCA, scDeepCluster adopts a ZINB distribution for as in Eq. (11) and (12). The loss function is

|  |  |  |
| --- | --- | --- |
|  |  | (30) |

where is the negative log-likelihood of the ZINB data distribution as defined in Eq. (13) and a clustering loss performed using KL divergence as Eq. (20) defined in DESC algorithm. Comparing to csvis, in terms of clustering regularization, scvis uses t-SNE objective function which is faithful to feature representation by keeping local structure, while scDeepcluster uses KL divergence based clustering which focuses more on clustering assignment.

*Evaluation metrics.* The following three metrics are used to evaluate the performance, NMI, clustering accuracy, and ARI.

*Results.* scDeepCluster was first tested on the simulation data and compared with other seven methods including DCA [15], two multi-kernel spectral clustering methods MPSSC [79] and SIMLR [40], CIDR [59], PCA + k-mean, scvis [73] and DEC[80]. In different dropout rate simulations, scDeepCluster significantly outperformed the other methods consistently. In signal strength, imbalanced sample size, and scalability simulations, scDeepcluster outperformed all other algorithms and scDeepCluster and most notably advantages for weak signals, robust against different data imbalance levels and scaled linearly with the number of cells. scDeepCluster was then tested on four real datasets (10X PBMC, Mouse ES cells, Mouse bladder cells, Worm neuron cells) and shown to outperform all other comparing algorithms. Particularly, MPSSC and SIMLR failed to process the full datasets due to quadratic complexity.

* + 1. **cscGAN: Conditional single-cell generative adversarial neural networks**

cscGAN [82] is a GAN model designed to augment the existing scRNA-seq samples by generating expression profiles of specific cell types or subpopulations.

*Model.* Two models, csGAN and cscGAN, were developed following the general formulation of WGAN described in section 3.3. The difference between the two models is that cscGAN is a conditional GAN such that the input to the generator also includes a class label or cell type, i.e. . The projection-based conditioning (PCGAN) method [83] was adopted to obtain the conditional GAN. For both models, the generator (three layers of 1024, 512, and 256 neurons) and discriminator (three layers of 256, 512, and 1024 neurons) are fully connected DNNs.

*Evaluation metrics.* The performance of cscGAN was assessed qualitatively by comparing the t-SNE plots of the real and generated samples. In the first approach, cluster-specific marker genes were obtained and SCENIC[84] was applied to learn the regulons for data including real and generated samples. Then, the ability of cscGAN to uncover the regulons from the real data was assessed. In the second approach, MMD between real and generated samples from cscGAN and other approaches were computed to measure their similarity. In the third approach, a classifier was trained to discriminate real from generated samples and the classification AUC was compared with that of a random guess.

*Results:* The performance of scGAN was first evaluated using PBMC data. The generated samples were shown to capture the desired clusters and the real data's regulons. Additionally, the AUC performance for classifying real from generated samples by a Random Forest classifier only reached 0.65, performance close to 0.5. Finally, scGAN's generated samples had a **smaller** MMD than those of Splatter, a state-of-the-art scRNA-seq data simulator [85]. Even though a large MMD was observed for scGAN when compared with that of SUGAR, another scRNA-seq simulator, SUGAR [86] was noted for prohibitively high runtime and memory. scGAN was further trained and assessed on the bigger mouse brain data and shown to model the expression dynamics across tissues. Then, the performance of cscGAN for generating cell-type-specific samples was evaluated using the PBMC data. cscGAN was shown to generate high-quality scRAN-seq data for specific cell types. Finally, the real PBMC samples were augmented with the generated samples. This augmentation improved the identification of rare cell types and the ability to capture transitional cell states from trajectory analysis.

* 1. **Multi-functional models**

Given the versatility of AE and VAE in addressing different scRAN-seq analysis challenges, DL models possessing multiple analysis functions have been developed. We survey these models in this section.

* + 1. **scVI: single-cell variational inference**

scVI [13] is designed to address a range of fundamental analysis tasks, including batch correction, visualization, clustering, and differential expression.

*Model.*scVI is a VAE that models the counts of each cell from different batches. scVI adopts a ZINB distribution for

|  |  |
| --- | --- |
|  | (31) |

which is defined similarly as Eq (11) in DCA, except that denotes the scaling factor for cell , which follows a log-Normal () prior as , therefore, represents the mean counts normalized by . Now, let be the batch ID of cell with being the total number of batches. Then, and are further modeled as functions of the -dimension latent variable and the batch ID by the decoder networks and as

|  |  |  |
| --- | --- | --- |
|  | , | (32) |

where the th element of and are and , respectively, and andare the decoder weights. Note that the lower layers of the two decoders are shared. For inference, both and are considered as latent variables and therefore is a mean-field approximate to the intractable posterior distribution and

|  |  |  |
| --- | --- | --- |
|  | = | (33) |

whose means and variances and are defined by the encoder networks and applied to and as

|  |  |  |
| --- | --- | --- |
|  | , | (34) |

where andare the encoder weights. Note that, like the decoders, the lower layers of the two encoders are also shared. Overall, the model parameters to be estimated by the variational optimization is **.** After inference is made, the latent vectors are used for visualization and clustering. provides a batch-corrected, size-factor normalized estimate of gene expression for each gene in each cell . An added advantage of the probabilistic representation by scVI is that it provides a natural probabilistic treatment of the subsequent differential analysis, resulting in lower variance in the adopted hypothesis tests.

*Evaluation metrics:* Run-time vs. dataset size was used to assess the scalability. To assess imputation, randomly selected non-zero entries in the count matrix were used as hold-out data and the distance between the imputed and the original values in the hold-out data were used to measure the imputation accuracy. For clustering, ARI, NMI, and the Silhouette index were used. The entropy of bach mixing was adopted to measure batch correction performance.

*Results:* scVI was evaluated for its scalability, the performance of imputation. For scalability, ScVI was shown to be faster than most nonDL algorithms and scalable to handle twice as many cells as nonDL algorithms with a fixed memory. For imputation, ScVI, together with other ZINB-based models, performed better than methods using alternative distributions. However, it underperformed for the dataset (HEMATO) with fewer cells. For the latent space, scVI was shown to provide a comparable stratification of cells into previously annotated cell types. Although scVI failed to ravel SIMLR, it is among the best in capturing biological structures (hierarchical structure, dynamics, etc.) and recognizing noise in data. For batch correction, it outperforms ComBat. For normalizing sequencing depth, the size factor inferred by scVI was shown to be strongly correlated with the sequencing depth. Interestingly, the negative binomial distribution in the ZINB was found to explain the proportions of zero expressions in the cells, whereas the zero probability is found to be more correlated with alignment errors. For differential expression analysis, scVI was shown to be among the best.

* + 1. **LDVAE: linearly decoded variational autoencoder**

LDVAE [87] is an adaption of scVI to improve the model interpretability but it still benefits from the scalability and efficiency of scVI. Also, this formulation applies to general VAE models and thus is not restricted to scRNA-seq analysis.

*Model.* LDVAE follows scVI’s formulation but replaces the decoder in (32) by a linear model

|  |  |  |
| --- | --- | --- |
|  |  | (35) |

where is the wieight matrix. Being the linear decoder provides interpretability in the sense that the relationship between latent representation and gene expression can be readily identified. LDVAE still follows the same loss and non-linear inference scheme as scVI.

*Evaluation metrics.* Thereconstruction errors of the VAEs were used to assess the performance.

*Results.* LDVAE’s latent variable could be used for clustering of cells with similar accuracy as a VAE. Although LDVAE had a higher reconstruction error than VAE, due to the linear decoder, the variations along the different axes of establish direct linear relationships with input genes. As an example from analyzing mouse embryo scRNA-seq, , the second element of , is shown to relate to simultaneous variations in the expression of gene and . In contrast, such interpretability would be intractable without approximation for a VAE. LDVAE was also shown to induce fewer correlations between latent variables and improved the grouping of the regulatory programs. LDVAE also scaled to a large dataset with ~2M cells.

* + 1. **SAUCIE**

SAUCIE [14] is an AE designed to perform multiple functions, including clustering, batch correlation, imputation, and visualization. SAUCIE is applied to the normalized data instead of count data.

*Model.* SAUCIE includes multiple model components designed for different functions.

1. Clustering: SAUCIE first introduced a "digital" binary encoding layer in the decoder that functions to encode the cluster ID. To learn this encoding, an entropy loss is introduced

|  |  |  |
| --- | --- | --- |
|  |  | (37) |

where is the probability (proportion) of activation on neuron by the previous layer. Minimizing this entropy loss promotes sparse neurons, thus forcing a binary encoding. To encourage clustering behavior, SAUCIE also introduced an intracluster loss as

|  |  |  |
| --- | --- | --- |
|  |  | (38) |

which computes the distance between the expressions of a pair of cells () that have the same cluster ID ().

1. Batch correction: To correct the batch effect, an MMD loss is introduced to measure the differences in terms of the distribution between batches in the latent space

|  |  |  |
| --- | --- | --- |
|  |  | (39) |

where is the total number of batches and is the latent variable of an arbitrarily chosen reference batch.

1. Imputation and visualization: The output of the decoder is taken by SAUCIE as an imputed version of the input gene expression. To visualize the data without performing an additional dimension reduction directly, the dimension of the latent variable is forced to 2.

Training the model includes two sequential runs. In the first run, an autoencoder is trained to minimize the loss with being the MSE reconstruction loss defined in (9) so that a batch-corrected, imputed input  can be obtained at the output of the decoder. In the second run, the bottleneck layer of the encoder from the first run is replaced by a 2-D latent code for visualization and a digital encoding layer is also introduced. This model takes the cleaned as the input and is trained for clustering by minimizing the loss . After the model is trained, is the imputed, batch-corrected gene expression. The 2-D latent code is used for visualization and the binary encoder encodes the cluster ID.

*Evaluation metrics.* For clustering, the Silhouette index was used. For batch correction, a mixing score similar in spirit to KBET is introduced to assess how well samples from two batches mix. For visualization, the precision and recall of the consistency of the neighboring cells of each cell between the original data space and the SAUCIE embedding space were calculated. For imputation, in addition to visual comprision of the imputed results, R2 statistics of the imputation results on a synthetic data were reported.

*Results.* SAUCIE was evaluated for clustering, batch correction, imputation, and visualization on both simulated and real scRNA-seq and scCyToF datasets. The performance was compared to minibatch *kmeans*, Phenograph [88] and 22 for clustering; MNN [28] and canonical correlation analysis (CCA) [29] for batch correction; PCA, Monocle2 [89], diffusion maps, UMAP [90], tSNE [74] and PHATE [91] for visualization; MAGIC [55], scImpute [33] and nearest neighbors completion (NN completion) for imputation. Results showed that SAUCIE had a better or comparable performance with other approaches. Also, SAUCIE has better scalability and faster runtimes than any of the other models. SAUCIE's results on the scCyToF dengue dataset were further analyzed in greater detail. SAUCIE was able to identify subtypes of the T cell populations and demonstrated distinct cell manifold between acute and healthy subjects.

* + 1. **scScope:**

scScope [92] is an AE with recurrent steps designed for imputation and batch correction.

*Model.* scScope has the following model design for batch correction and imputation.

1. Batch correction: A batch correction layer is applied to the input expression as

|  |  |  |
| --- | --- | --- |
|  |  | (40) |

where and is the ReLu activation function, is the batch correction matrix, is an indicator vector with entry 1 indicates the batch of the input, and is the total number of batches.

1. Recursive imputation: Instead of using the reconstructed expression as the imputed expression like in SAUCIE, scScope adds an imputer to to recursively improve the imputation result. The imputer is a single-layer autoencoder, whose decoder performs the imputation as

|  |  |  |
| --- | --- | --- |
|  |  | (41) |

where is the output of the imputer encoder, is the imputer decoder, and is a masking function that set the elements in that correspond to the non-missing values to zero. Then, will be fed back to fill the missing value in the batch corrected input as , which will be passed on to the main autoencoder. This recursive imputation can iterate multiple cycles as selected.

The loss function modifies the conventional reconstruction loss as

|  |  |  |
| --- | --- | --- |
|  |  | (42) |

where is the total number of recursion, is the reconstructed expression at th recursion, is another masking function that forces the loss to compute only the non-missing values in .

*Evaluation metrics.* The performance was tested on both real and simulated datasets. ARI was used to evaluate the ability to match SAUCIE outputs to capture cell subpopulations. For imputation, the reconstruction error on held-out biological data was used. For batch correction, entropy of mixing was applied.

*Results*. scScope was evaluated for its scalability, clustering, imputation, and batch correction. It was compared with PCA, MAGIC, ZINB-WaVE, SIMLR, AE, scVI, and DCA. For scalability and training speed, scScope was shown to offer scalability (for >100K cells) with high efficiency (faster than most of the approaches). For clustering results, scScope was shown to outperform most of the algorithms on small simulated datasets but offer similar performance on large simulated datasets. For batch correction, scScope performed comparably with other approaches but with faster runtime. For imputation, scScope produced smaller errors consistently across a different range of expression. scScope was further shown to be able to identify rear cell populations from a large mix of cells.

* 1. **Doublet classification**

In scRNA-Seq, doublets are technical artifacts formed when two or more cells are encapsulated into one reaction volume by chance, therefore need to be removed before quantifying gene expression.

* + 1. **Solo**

Solo [93] is a semi-supervised deep learning approach that identifies doublets with greater accuracy than other existing methods.

*Model.* Solo identifies doublets by embedding cells using VAE first and then appends a feed-forward neural network layer to the encoder to form a supervised classifier on the labeled training samples. The method assumes that most cells in an experiment are singlets and one can approximate a view into the doublet population by generating simulated doublets in silico from the observed data. In Solo, doublets are assumed to have less than twice the UMI depth on average but an option to scale the merged counts by a specific factor is also provided. The method operates in three phases: (1) doublet simulation where a number of doublets are generated by taking the sum of randomly chosen observed cells considered putative singlets, and then model was trained to differentiate these in silico doublets from the observed data; (2) cell embedding where Solo embeds cells in a nonlinear latent space via a VAE that follows the scVI architecture; and (3) classifier training where a standard discriminative classifier is added to the end of the scVI encoder and trained using the simulated doublet samples and the observed cells to predict doublet status. Only the added classifier layer was trained with a binary cross-entropy loss, the scVI encoder was fixed during training.

*Evaluation metrics.* The performance was measured using precision-recall (PR) and receiver operator characteristic (ROC) curves and summarized with average precision (AP) and area under the receiver operator curve (AUROC).

*Results*. Solo performance was tested on a singlet/doublet population simulated using the Splatter toolkit [85] and several experimental datasets, where Solo exceeds the performance of previous non-deep learning methods like Scrublet [94] and DoubletFinder [95] by greater margins in the larger and more complex datasets. Solo also outperformed existing computational methods for this task on a variety of cell line and tissue datasets with experimental doublet annotations.

* 1. **Automated cell type identification**

scRNA-Seq is able to catalog cell types in complex tissues under different conditions. However, the commonly adopted manual cell typing approach based on known markers is time-consuming and less reproducible. We survey deep learning models that take the task of automated cell type identification.

* + 1. ***DigitalDLSorter***

DigitalDLSorter [96] was proposed to identify and quantify the immune cells infiltrated in tumors captured in bulk RNA-seq, utilizing single-cell RNA-seq data.

*Model.*DigitalDLSorter is a 4-layer DNN (2 hidden layers of 200 neurons each and an output of 10 cell types). The DigitalDLSorter is trained with two single-cell datasets: breast cancers [97] and corectal cancers [98]. For each cell, it is determined to be tumor cell or non-tumor cell using RNA-seq based CNV method[97], followed with using xCell algorithm [99] to determine immune cell types for non-tumor cells. Different pseudo bulk (from 100 cells) RNA-seq datasets were prepared with known mixture proportion to train the DNN. The output of DigitalDLSorter is the predicted proportions of cell types in the input bulk sample.

*Evaluation.* Pearson Correlation between the predicted and real cell type proportions was used to assess the performance.

*Result.* DigitalDLSorter was first tested on simulated bulk RNA-seq samples. DigitalDLSorter achieved excellent agreement (linear correlation of 0.99 for colorectual cancer, and good agreement in quadratic relationship for breast cancer) at predicting cell types proportion. The proportion of immune and nonimmune cell subtypes of test bulk TCGA samples was predicted by DigitalDLSorter and the results showed very good correlation to other deconvolution tools including TIMER [97], ESTIMATE [100], EPIC [101] and MCPCounter [102]. Using DigitalDLSorter predicted CD8+ (good prognosis for overall and disease-free survival) and Monocytes-Macrophages (MM, indicator for protumoral activity) proportions, it is found that patients with higher CD8+/MM ratio had better survival for both cancer types than those with lower CD8+/MM ratio. Both EPIC and MCPCounter yielded non-significant survival association using their cell proportion estimate.

* + 1. **scCapsNet**

scCapsNet [103] is an interpretable capsule network designed for cell type prediction. The paper showed that the trained network could be interpreted to inform marker genes and regulatory modules of cell types.

*Model.* Two-layer architecture of scCapsNet takes log-transformed, normalized expressions as input to form a feature extraction network (consists of parallel sigle-layer neural networks) followed by a capsule network for cell-type classification (type capsules). For each of L parallel feature extraction layer, it generates a primary capsule as

|  |  |  |
| --- | --- | --- |
|  |  | (43) |

where is the weight matrix. Then, the primary capsules are fed into the capsule network to compute type capsules , one for each cell type, as

|  |  |  |
| --- | --- | --- |
|  |  | (44) |

where is the squashing function [104] to normalize the magnitude of its input vector to be less than one, is another trainable weight matrix, and are the coupling coefficients that represent the probability distribution of each primary capsule’s impact on the prediction of cell type . is not trained but computed through the dynamic routing process proposed in the original capsule networks [103]. The magnitude of each type of capsule represents the probability of a single cell belonging to cell type , which will be used for cell type classification.

The training of the network minimizes the cross-entropy loss by the back-propagation algorithm. Once trained, the interpretation of marker genes and regulatory modules can be achieved by determining first the important primary capsules for each cell type and then the most significant genes for each important primary capsule (identified based on directly). To determine the genes that are important for an important primary capsule , genes are ranked base on the scores of the first principal component computed from the columns of and then the markers are obtained by a greedy search along with the ranked list for the best classification performance.

*Evaluation matrics:* Accuracy of the predicted cell types was evaluated.

*Results.* scCapsNet’s performance was evaluated on human PBMCs [105] and mouse retinal bipolar cells [106] datasets and shown to have comparable accuracies (99% and 97% respectively) with DNN and other popular ML algorithms (SVM, random forest, LDA and nearest neighbor). However, the interpretability of scCapsNet was demonstrated extensively. First, examining the coupling coefficients for each cell type showed that only a few primary capsules have high values and thus is effective. Subsequently, a set of core genes were identified for each effective capsules using the greedy search on the PC-score ranked gene list. GO enrichment analysis showed that these core genes were enriched in cell type related biological functions. Mapping the expression data into space spanned by PCs of the columns of corresponding to all core genes uncovered regulatory modules that would be missed by the T-SNE of gene expressions, which demonstrated the effectiveness of the embeddings learned by scCapsNet in capturing the functionally important features.

* + 1. **netAE: network-enhanced autoencoder**

netAE [107] is a VAE-based semi-supervised cell type prediction model that deals with scenarios of having a small number of labeled cells.

*Model.* netAE works with UMI counts and assumes a ZINB distribution for as in Eq. (32) in scVI. However, netAE adopts the general VAE loss as in eq. (6) with two function-specific loss as

|  |  |  |
| --- | --- | --- |
|  |  | (45) |

where is a set of indices for all cells and is a subset of for only cells with cell type labels, is modified Newman and Girvan modularity [108] that quantifies cluster strength using , is the softmax function, and is the cell type label. The second loss in (34) functions as a clustering constraint and the last term is the cross-entropy loss that constrains the cell type classification.

*Evaluation metrics*. Accuracy of cell type classification was used to assess the performance.

*Results:* netAE was compared with popular dimension reduction methods including scVI, ZIFA, PCA and AE as well as a semi-supervised method scANVI [109]. For different dimension reduction methods, cell type classification from latent features of cells was carried out using KNN and logistic regression. The effect of different labeled samples sizes on classification performance was also investigated, where the sample size varied from as few as 10 cells to 70% of all cells. Among 3 test datasets (mouse brain cortex, human embryo development, and mouse hematopoietic stem and progenitor cells), netAE outperformed most of the baseline methods.  Latent features were visualized using t-SNE and cell clusters by netAE were tighter than those by other embedding spaces. There was also consistency of better cell-type classification with improved cell type clustering by netAE. This suggested that the latent spaces learned with added modularity constraint in the loss helped identify clusters of similar cells.  Ablation study by removing each of the three loss terms in Eq (45) showed a drop of cell-type classification accuracy, suggesting all three were necessary for the optimal performance.

* + 1. **scDGN - supervised adversarial alignment of single-cell RNA-seq data**

scDGN [110], or Single Cell Domain Generalization Network, is an domain adversarial network that aims to accurately assign cell types of single cells while performing batch removal (domain adaptation) at the same time. It benefits from the superior ability of domain adversarial learning to learn representations that are invariant to technical confounders.

*Model.* scDGN takes the log-transformed, normalized expression as the input and has three main modules: i) an encoder () for dimension reduction of scRNA-Seq data, ii) cell-type classifier, with parameters , and iii) domain (batch) discriminator, . The model has a Siamese design and the training takes a pair of cells , each from the same or different batches. The encoder network contains two hidden layers with 1146 and 100 neurons. classifies the cell type and predicts whether and are from the same batch or not. The overall loss is denoted by

|  |  |  |
| --- | --- | --- |
|  |  | (46) |

where is the cross entropy loss, is a contrastive loss as described in [111]. Notice that (46) has an adversarial formulation and minimizing this loss maximizes the misclassification of cells from different batches, thus making them indistinguishable. Similar to GAN training, scDGN is trained to iteratively solve: and .

*Evaluation metrics.* Prediciton accuracy was used to evaluate the performance.

*Results.* scDGN was tested for classifying cell types and aligning batches ranging in size from 10 to 39 cell types and from 4 to 155 batches. The performance was compared to a series of deep learning and traditional machine learning methods, including Lin et al. DNN [68], CaSTLe [112], MNN [28], scVI [13], and Seurat [9]. scDGN outperformed all other methods in the classification accuracy on a subset of scQuery datasets (0.29), PBMC (0.87), and 4 of the six Seurat pancreatic datasets (0.86-0.95). PCA visualization of the learned data representations demonstrated that scDGN overcame the batch differences and clearly separated cell clusters based on cell types, while other methods were vulnerable to batch effects. In summary, scDGN is a supervised adversarial alignment method to eliminate the batch effect of scRNA-Seq data and create cleaner representations of cell types.

* 1. **Biological function prediction**

Predicting biological function and/reponse to treatment at single cell level or cell types is critical to understand cell system functioning and potent response to stimulation. Utilize DL models that captures gene-gene relationship and their property at latent space, several models we reviewed below provide some exciting approach to achieve complex biological functions and outcomes.

* + 1. **CNNC: convolutional neural network for coexpression**

CNNC [113] is proposed to infer causal interactions between genes from scRNA-seq data.

*Model.* CNNC is a Convolutional Neural Network (CNN), the most popular DL model. CNNC takes expression levels of two genes from many cells and transforms them into a 32 x 32 image-like normalized empirical probability function (NEPDF), which measures the probabilities of observing different coexpression levels between the two genes. CNNC includes 6 convolutional layers, 3 maxpooling layers, 1 flatten layer, and one output layer. All convolution layers have 32 kernels of size 3x3. Depending on the application, the output layer can be designed to predict the state of interaction (Yes/No) between the genes or the causal interaction between the input genes (no interaction, gene A regulates gene B, or gene B regulates gene A).

*Evaluation matrics.* Prediction AUROC, AUPRC, and accuracy were assessed.

*Result.* CNNC was trained to predict transcription factor (TF)-Gene interactions using the mESC data from scQuery [114], where the ground truth interactions were obtained from the ChIP-seq dataset from the GTRD database [115]. The performance was compared with DNN, count statistics [116], and mutual information-based approach [117]. CNNC was shown to have more than 20% higher AUPRC than other methods and reported almost no false-negative for the top 5% predictions. CNNC was also trained to predict the pathway regulator-target gene pairs. The positive regulator-gene pairs were obtained from KEGG [118], Reactome [119], and negative samples were genes pairs that appeared in pathways but not interacted. CNNC was shown to have better performance of predicting regulator-gene pairs on both KEGG and Reactome pathways than other methods including Pearson correlation, count statistics, GENIE3 [120], Mutual information, Bayesian directed network (BDN), and DREMI [117]. CNNC was also applied for causality prediction between two genes, that is if two genes regulate each other and if they do, which gene is the regulator. The ground truth causal relationships were also obtained from the KEGG and Reactome datasets. Again, CNNC reported better performance than BDN, the common method developed to learn casual relationshop from gene expression data. CNNC was finally trained to assign 3 essential cell functions (cell cycle, circadian rhythm, and immune system) to genes. This is achieved by training CNNC to predict pair of genes from same function (e.g. Cell Cycle defined by mSigDB from gene set enrichment analysis (GSEA) [121]) as 1 and all other pairs as 0. The performance was compared with “guilt by association” and DNN, and CNNC were shown to have more than 4% higher AUROC and reported all positives for the top 10% predictions.

* + 1. **scGen, a generative model to predict perturbation response of single cells across cell types**

scGen [122] is designed to learn cell response to certain perturbation (drug treatment, gene knockout, etc) from single cell expression data and predict the response to the same perturbation for a new sample or a new cell type. The novelty of scGen is that it learns the response in the latent space instead of the expression data space.

*Model.* ScGen follows the general VAE for scRNA-seq data but uses the “latent space arithmetics” to learn perturbations' response. Given scRNAseq samples of perturbed (denoted as *p*) and unperturbed cells (denoted as *unp*), a VAE model is trained. Then, the latent space representation and are obtained for the perturbed and unperturbed cells. Following the notion that VAE could map nonlinear operations (e.g., perturbation) in the data space to linear operations in the latent space, ScGen estimate the response in the latent space as  , where is the average representation of samples from the same cell type or different cell types. Then, given the latent representation of an unperturbed cell for a new sample from the same or different cell type, the latent representation of the corresponding perturbed cell can be predicted as =. The expression of the perturbed cell can also be estimated by feeding into the VAE decoder. The scGen can also be expanded to samples and treatment across two species (using orthologues between species). When scGen is trained for species 1, or *s*1, with both perturbed and unperturbed cells but species 2, *s*2, with only unperturbed cells, then the latent code for the perturbed cells from *s*2 can be predicted as where captures the response of perturbation and represents the difference between species.

*Evaluation matrics.* Correlation and UMAP visualization were used to assess the performance.

*Result.* scGen was applied to predict perturbation of out-of-samples response in human PBMCs data, and scGen showed a higher average correlation (R2= 0.948) between predicted and real data for six cell types. Comparing with other methods including CVAE [123], style transfer GAN [124], linear approaches based on vector arithmetics(VA) [122] and PCA+VA, scGen predicted full distribution of ISG15 gene (strongest regulated gene by IFN-b) response to IFN- b [125], while others might predicted mean (CAVE and style transfer GAN) but failed to produce the full distribution. scGen was also tested on predicting the intestinal epithelial cells’ response to infection[126]. For early transit-amplifying cells, scGen showed good prediction (R2=0*.*98) for both H. poly and Salmonella infection. Finally, scGen was evaluated for perturbation across species using scRNA-seq data set by Hagai et al [127], which comprised of bone marrow-derived mononuclear phagocytes from mice, rats, rabbits, and pigs perturbed with lipopolysaccharide (LPS). scGen’s predictions of LPS perturbion responses were shown to be highly correlated (R2 = 0.91) with the real responses.

1. **Conclusion and discussions**

Single cell RNA-Seq technologies are tools that help exploring the cell-type composition of a particular sample and how it is dysregulated during development and disease.

**Bibliography**

1. Lahnemann D, Koster J, Szczurek E, McCarthy DJ, Hicks SC, Robinson MD, Vallejos CA, Campbell KR, Beerenwinkel N, Mahfouz A *et al*: **Eleven grand challenges in single-cell data science**. *Genome Biol* 2020, **21**(1):31.

2. Vitak SA, Torkenczy KA, Rosenkrantz JL, Fields AJ, Christiansen L, Wong MH, Carbone L, Steemers FJ, Adey A: **Sequencing thousands of single-cell genomes with combinatorial indexing**. *Nat Methods* 2017, **14**(3):302-308.

3. Wu H, Wang C, Wu S: **Single-Cell Sequencing for Drug Discovery and Drug Development**. *Curr Top Med Chem* 2017, **17**(15):1769-1777.

4. Bost P, Giladi A, Liu Y, Bendjelal Y, Xu G, David E, Blecher-Gonen R, Cohen M, Medaglia C, Li H *et al*: **Host-Viral Infection Maps Reveal Signatures of Severe COVID-19 Patients**. *Cell* 2020, **181**(7):1475-1488 e1412.

5. Kinker GS, Greenwald AC, Tal R, Orlova Z, Cuoco MS, McFarland JM, Warren A, Rodman C, Roth JA, Bender SA *et al*: **Pan-cancer single-cell RNA-seq identifies recurring programs of cellular heterogeneity**. *Nat Genet* 2020, **52**(11):1208-1218.

6. Navin NE: **The first five years of single-cell cancer genomics and beyond**. *Genome Res* 2015, **25**(10):1499-1507.

7. Mannarapu M, Dariya B, Bandapalli OR: **Application of single-cell sequencing technologies in pancreatic cancer**. *Mol Cell Biochem* 2021, **476**(6):2429-2437.

8. Wauters E, Van Mol P, Garg AD, Jansen S, Van Herck Y, Vanderbeke L, Bassez A, Boeckx B, Malengier-Devlies B, Timmerman A *et al*: **Discriminating mild from critical COVID-19 by innate and adaptive immune single-cell profiling of bronchoalveolar lavages**. *Cell Res* 2021, **31**(3):272-290.

9. Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck WM, 3rd, Hao Y, Stoeckius M, Smibert P, Satija R: **Comprehensive Integration of Single-Cell Data**. *Cell* 2019, **177**(7):1888-1902 e1821.

10. Wolf FA, Angerer P, Theis FJ: **SCANPY: large-scale single-cell gene expression data analysis**. *Genome Biol* 2018, **19**(1):15.

11. 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**(10):1303-1319.

12. Amodio M, Van Dijk D, Srinivasan K, Chen WS, Mohsen H, Moon KR, Campbell A, Zhao Y, Wang X, Venkataswamy M: **Exploring single-cell data with deep multitasking neural networks**. *Nature methods* 2019:1-7.

13. Lopez R, Regier J, Cole MB, Jordan MI, Yosef N: **Deep generative modeling for single-cell transcriptomics**. *Nat Methods* 2018, **15**(12):1053-1058.

14. Amodio M, van Dijk D, Srinivasan K, Chen WS, Mohsen H, Moon KR, Campbell A, Zhao Y, Wang X, Venkataswamy M *et al*: **Exploring single-cell data with deep multitasking neural networks**. *Nat Methods* 2019, **16**(11):1139-1145.

15. Eraslan G, Simon LM, Mircea M, Mueller NS, Theis FJ: **Single-cell RNA-seq denoising using a deep count autoencoder**. *Nat Commun* 2019, **10**(1):390.

16. 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**(15):e85.

17. 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**(1):211.

18. Tran HTN, Ang KS, Chevrier M, Zhang X, Lee NYS, Goh M, Chen J: **A benchmark of batch-effect correction methods for single-cell RNA sequencing data**. *Genome Biol* 2020, **21**(1):12.

19. Petegrosso R, Li Z, Kuang R: **Machine learning and statistical methods for clustering single-cell RNA-sequencing data**. *Brief Bioinform* 2020, **21**(4):1209-1223.

20. Abdelaal T, Michielsen L, Cats D, Hoogduin D, Mei H, Reinders MJT, Mahfouz A: **A comparison of automatic cell identification methods for single-cell RNA sequencing data**. *Genome Biol* 2019, **20**(1):194.

21. Picelli S, Bjorklund AK, Faridani OR, Sagasser S, Winberg G, Sandberg R: **Smart-seq2 for sensitive full-length transcriptome profiling in single cells**. *Nat Methods* 2013, **10**(11):1096-1098.

22. Macosko EZ, Basu A, Satija R, Nemesh J, Shekhar K, Goldman M, Tirosh I, Bialas AR, Kamitaki N, Martersteck EM *et al*: **Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets**. *Cell* 2015, **161**(5):1202-1214.

23. Chen G, Ning B, Shi T: **Single-Cell RNA-Seq Technologies and Related Computational Data Analysis**. *Front Genet* 2019, **10**:317.

24. Eisenstein M: **Single-cell RNA-seq analysis software providers scramble to offer solutions**. *Nat Biotechnol* 2020, **38**(3):254-257.

25. Vallejos CA, Marioni JC, Richardson S: **BASiCS: Bayesian Analysis of Single-Cell Sequencing Data**. *PLoS Comput Biol* 2015, **11**(6):e1004333.

26. Lun AT, Bach K, Marioni JC: **Pooling across cells to normalize single-cell RNA sequencing data with many zero counts**. *Genome Biol* 2016, **17**:75.

27. Hafemeister C, Satija R: **Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression**. *Genome Biol* 2019, **20**(1):296.

28. Haghverdi L, Lun ATL, Morgan MD, Marioni JC: **Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors**. *Nat Biotechnol* 2018, **36**(5):421-427.

29. Butler A, Hoffman P, Smibert P, Papalexi E, Satija R: **Integrating single-cell transcriptomic data across different conditions, technologies, and species**. *Nat Biotechnol* 2018, **36**(5):411-420.

30. Korsunsky I, Millard N, Fan J, Slowikowski K, Zhang F, Wei K, Baglaenko Y, Brenner M, Loh PR, Raychaudhuri S: **Fast, sensitive and accurate integration of single-cell data with Harmony**. *Nat Methods* 2019, **16**(12):1289-1296.

31. Peng T, Zhu Q, Yin P, Tan K: **SCRABBLE: single-cell RNA-seq imputation constrained by bulk RNA-seq data**. *Genome Biol* 2019, **20**(1):88.

32. Huang M, Wang J, Torre E, Dueck H, Shaffer S, Bonasio R, Murray JI, Raj A, Li M, Zhang NR: **SAVER: gene expression recovery for single-cell RNA sequencing**. *Nat Methods* 2018, **15**(7):539-542.

33. Li WV, Li JJ: **An accurate and robust imputation method scImpute for single-cell RNA-seq data**. *Nat Commun* 2018, **9**(1):997.

34. Roweis ST, Saul LK: **Nonlinear dimensionality reduction by locally linear embedding**. *Science* 2000, **290**(5500):2323-2326.

35. Welch JD, Hartemink AJ, Prins JF: **SLICER: inferring branched, nonlinear cellular trajectories from single cell RNA-seq data**. *Genome Biol* 2016, **17**(1):106.

36. Linderman GC, Rachh M, Hoskins JG, Steinerberger S, Kluger Y: **Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data**. *Nat Methods* 2019, **16**(3):243-245.

37. Becht E, McInnes L, Healy J, Dutertre CA, Kwok IWH, Ng LG, Ginhoux F, Newell EW: **Dimensionality reduction for visualizing single-cell data using UMAP**. *Nat Biotechnol* 2018.

38. Subelj L, Bajec M: **Unfolding communities in large complex networks: combining defensive and offensive label propagation for core extraction**. *Phys Rev E Stat Nonlin Soft Matter Phys* 2011, **83**(3 Pt 2):036103.

39. Traag VA, Waltman L, van Eck NJ: **From Louvain to Leiden: guaranteeing well-connected communities**. *Sci Rep* 2019, **9**(1):5233.

40. Wang B, Zhu J, Pierson E, Ramazzotti D, Batzoglou S: **Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning**. *Nat Methods* 2017, **14**(4):414-416.

41. Finak G, McDavid A, Yajima M, Deng J, Gersuk V, Shalek AK, Slichter CK, Miller HW, McElrath MJ, Prlic M *et al*: **MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data**. *Genome Biol* 2015, **16**:278.

42. Kharchenko PV, Silberstein L, Scadden DT: **Bayesian approach to single-cell differential expression analysis**. *Nat Methods* 2014, **11**(7):740-742.

43. Miao Z, Deng K, Wang X, Zhang X: **DEsingle for detecting three types of differential expression in single-cell RNA-seq data**. *Bioinformatics* 2018, **34**(18):3223-3224.

44. Rosenberg AaH, J. : **Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)**. 2007:410-420.

45. Hubert L, Arabie P: **Comparing Partitions**. *Journal of Classification* 1985, **2**(2-3):193-218.

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

47. Strehl A, Ghosh J: **Cluster ensembles---a knowledge reuse framework for combining multiple partitions**. *Journal of machine learning research* 2002, **3**(Dec):583-617.

48. Hou W, Ji Z, Ji H, Hicks SC: **A systematic evaluation of single-cell RNA-sequencing imputation methods**. *Genome Biol* 2020, **21**(1):218.

49. Sun S, Zhu, J., Ma, Y. and Zhou, X.: **Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis**. *Genome Biol* 2019, **20**(1):269.

50. Hubert LaA, P.: **Comparing Partitions**. *Journal of Classification* 1985, **2**:193-218.

51. Borgwardt KM, Gretton A, Rasch MJ, Kriegel HP, Scholkopf B, Smola AJ: **Integrating structured biological data by Kernel Maximum Mean Discrepancy**. *Bioinformatics* 2006, **22**(14):e49-57.

52. Buttner M, Miao Z, Wolf FA, Teichmann SA, Theis FJ: **A test metric for assessing single-cell RNA-seq batch correction**. *Nat Methods* 2019, **16**(1):43-49.

53. Cover TM: **Elements of information theory**: John Wiley & Sons; 1999.

54. Rosenberg A, Hirschberg J: **V-measure: A conditional entropy-based external cluster evaluation measure**. In: *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL): 2007*; 2007: 410-420.

55. van Dijk D, Sharma R, Nainys J, Yim K, Kathail P, Carr AJ, Burdziak C, Moon KR, Chaffer CL, Pattabiraman D *et al*: **Recovering Gene Interactions from Single-Cell Data Using Data Diffusion**. *Cell* 2018, **174**(3):716-729 e727.

56. Wang J, Agarwal D, Huang M, Hu G, Zhou Z, Ye C, Zhang NR: **Data denoising with transfer learning in single-cell transcriptomics**. *Nat Methods* 2019, **16**(9):875-878.

57. Badsha MB, Li R, Liu B, Li YI, Xian M, Banovich NE, Fu AQ: **Imputation of single-cell gene expression with an autoencoder neural network**. *Quant Biol* 2020, **8**(1):78-94.

58. Yu B, Chen C, Qi R, Zheng R, Skillman-Lawrence PJ, Wang X, Ma A, Gu H: **scGMAI: a Gaussian mixture model for clustering single-cell RNA-Seq data based on deep autoencoder**. *Brief Bioinform* 2020.

59. Lin P, Troup M, Ho JW: **CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data**. *Genome Biol* 2017, **18**(1):59.

60. Berthelot D, Schumm, T. and Metz, L.: **BEGAN: Boundary Equilibrium Generative Adversarial Networks**. *arXiv* 2017.

61. Wang T, Johnson TS, Shao W, Lu Z, Helm BR, Zhang J, Huang K: **BERMUDA: a novel deep transfer learning method for single-cell RNA sequencing batch correction reveals hidden high-resolution cellular subtypes**. *Genome Biol* 2019, **20**(1):165.

62. Crow M, Paul A, Ballouz S, Huang ZJ, Gillis J: **Characterizing the replicability of cell types defined by single cell RNA-sequencing data using MetaNeighbor**. *Nat Commun* 2018, **9**(1):884.

63. Polanski K, Young MD, Miao Z, Meyer KB, Teichmann SA, Park JE: **BBKNN: fast batch alignment of single cell transcriptomes**. *Bioinformatics* 2020, **36**(3):964-965.

64. Li X, Wang K, Lyu Y, Pan H, Zhang J, Stambolian D, Susztak K, Reilly MP, Hu G, Li M: **Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis**. *Nat Commun* 2020, **11**(1):2338.

65. Guo X, Gao, L., Liu, X., and Yin, J.: **Improved deep embedded clustering with local structure preservation**. *Proc 26th International Joint Conference on Artificial Integlligence* 2017:1753-1759.

66. Hie B, Bryson B, Berger B: **Efficient integration of heterogeneous single-cell transcriptomes using Scanorama**. *Nat Biotechnol* 2019, **37**(6):685-691.

67. 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**(1):63.

68. Lin C, Jain S, Kim H, Bar-Joseph Z: **Using neural networks for reducing the dimensions of single-cell RNA-Seq data**. *Nucleic Acids Res* 2017, **45**(17):e156.

69. Fowlkes EB, Mallows CL: **A Method for Comparing Two Hierarchical Clusterings**. *Journal of the American Statistical Association* 1983, **78**(383):553-569.

70. Rashid S, Shah S, Bar-Joseph Z, Pandya R: **Dhaka: Variational Autoencoder for Unmasking Tumor Heterogeneity from Single Cell Genomic Data**. *Bioinformatics* 2019.

71. Tirosh I, Izar B, Prakadan SM, Wadsworth MH, 2nd, Treacy D, Trombetta JJ, Rotem A, Rodman C, Lian C, Murphy G *et al*: **Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq**. *Science* 2016, **352**(6282):189-196.

72. Zahn H, Steif A, Laks E, Eirew P, VanInsberghe M, Shah SP, Aparicio S, Hansen CL: **Scalable whole-genome single-cell library preparation without preamplification**. *Nature Methods* 2017, **14**(2):167-173.

73. Ding J, Condon A, Shah SP: **Interpretable dimensionality reduction of single cell transcriptome data with deep generative models**. *Nat Commun* 2018, **9**(1):2002.

74. van der Maaten LH, G.: **Visualizing data using t-SNE**. *J Mach Learn* 2008, **9**:2579-2605.

75. Gronbech CH, Vording MF, Timshel PN, Sonderby CK, Pers TH, Winther O: **scVAE: variational auto-encoders for single-cell gene expression data**. *Bioinformatics* 2020, **36**(16):4415-4422.

76. Wang D, Gu J: **VASC: Dimension Reduction and Visualization of Single-cell RNA-seq Data by Deep Variational Autoencoder**. *Genomics Proteomics Bioinformatics* 2018, **16**(5):320-331.

77. Jang E. GSaPB: **Categorical reparameterization with gumbel-softmax**. *arXiv* 2016.

78. Tian T, Wan, J., Song, Q. et al.: **Clustering single-cell RNA-seq data with a model-based deep learning approach**. *Nat Mach Intell* 2019, **1**.

79. Regev A, Teichmann SA, Lander ES, Amit I, Benoist C, Birney E, Bodenmiller B, Campbell P, Carninci P, Clatworthy M *et al*: **The Human Cell Atlas**. *Elife* 2017, **6**.

80. Xie J, Girshick R, Farhadi A: **Unsupervised deep embedding for clustering analysis**. In: *International conference on machine learning: 2016*: PMLR; 2016: 478-487.

81. La Manno G, Gyllborg D, Codeluppi S, Nishimura K, Salto C, Zeisel A, Borm LE, Stott SRW, Toledo EM, Villaescusa JC *et al*: **Molecular Diversity of Midbrain Development in Mouse, Human, and Stem Cells**. *Cell* 2016, **167**(2):566-580 e519.

82. Marouf M, Machart P, Bansal V, Kilian C, Magruder DS, Krebs CF, Bonn S: **Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks**. *Nat Commun* 2020, **11**(1):166.

83. Miyato TaK, M: **cGANs with projection discriminator**. *Preprint* 2018.

84. Aibar S, Gonzalez-Blas CB, Moerman T, Huynh-Thu VA, Imrichova H, Hulselmans G, Rambow F, Marine JC, Geurts P, Aerts J *et al*: **SCENIC: single-cell regulatory network inference and clustering**. *Nat Methods* 2017, **14**(11):1083-1086.

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

86. Lindenbaum O, Stanley, J. S., Wolf, G. and Krishnaswamy, S. : **Geometry-based data generation**. *Advances in Neural Information Processing Systems* 2018.

87. Svensson V, Gayoso A, Yosef N, Pachter L: **Interpretable factor models of single-cell RNA-seq via variational autoencoders**. *Bioinformatics* 2020, **36**(11):3418-3421.

88. Levine JH, Simonds EF, Bendall SC, Davis KL, Amir el AD, Tadmor MD, Litvin O, Fienberg HG, Jager A, Zunder ER *et al*: **Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis**. *Cell* 2015, **162**(1):184-197.

89. Qiu X, Mao Q, Tang Y, Wang L, Chawla R, Pliner HA, Trapnell C: **Reversed graph embedding resolves complex single-cell trajectories**. *Nat Methods* 2017, **14**(10):979-982.

90. McInnes L, Healy, J. & Melville, J.: **Umap: uniform manifold approximation and projection for dimension reduction**. *ArXiv* 2018.

91. Moon KRea: **PHATE: a dimensionality reduction method for visualizing trajectory structures in high-dimensional biological data**. *bioRxiv* 2017.

92. 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**(4):311-314.

93. 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**(1):95-101 e105.

94. Wolock SL, Lopez R, Klein AM: **Scrublet: Computational Identification of Cell Doublets in Single-Cell Transcriptomic Data**. *Cell Syst* 2019, **8**(4):281-291 e289.

95. McGinnis CS, Murrow LM, Gartner ZJ: **DoubletFinder: Doublet Detection in Single-Cell RNA Sequencing Data Using Artificial Nearest Neighbors**. *Cell Syst* 2019, **8**(4):329-337 e324.

96. Torroja C, Sanchez-Cabo F: **Digitaldlsorter: Deep-Learning on scRNA-Seq to Deconvolute Gene Expression Data**. *Front Genet* 2019, **10**:978.

97. Chung W, Eum HH, Lee HO, Lee KM, Lee HB, Kim KT, Ryu HS, Kim S, Lee JE, Park YH *et al*: **Single-cell RNA-seq enables comprehensive tumour and immune cell profiling in primary breast cancer**. *Nat Commun* 2017, **8**:15081.

98. Li H, Courtois ET, Sengupta D, Tan Y, Chen KH, Goh JJL, Kong SL, Chua C, Hon LK, Tan WS *et al*: **Reference component analysis of single-cell transcriptomes elucidates cellular heterogeneity in human colorectal tumors**. *Nat Genet* 2017, **49**(5):708-718.

99. Aran D, Hu Z, Butte AJ: **xCell: digitally portraying the tissue cellular heterogeneity landscape**. *Genome Biol* 2017, **18**(1):220.

100. Yoshihara K, Shahmoradgoli M, Martinez E, Vegesna R, Kim H, Torres-Garcia W, Trevino V, Shen H, Laird PW, Levine DA *et al*: **Inferring tumour purity and stromal and immune cell admixture from expression data**. *Nat Commun* 2013, **4**:2612.

101. Racle J, de Jonge K, Baumgaertner P, Speiser DE, Gfeller D: **Simultaneous enumeration of cancer and immune cell types from bulk tumor gene expression data**. *Elife* 2017, **6**.

102. Becht E, Giraldo NA, Lacroix L, Buttard B, Elarouci N, Petitprez F, Selves J, Laurent-Puig P, Sautes-Fridman C, Fridman WH *et al*: **Estimating the population abundance of tissue-infiltrating immune and stromal cell populations using gene expression**. *Genome Biol* 2016, **17**(1):218.

103. Wang L, Nie, R., Yu, Z. et al.: **An interpretable deep-learning architecture of capsule networks for identifying cell-type gene expression programs from single-cell RNA-sequencing data**. *Nat Mach Intell* 2020, **2**:693-703.

104. Patel ND, Nguang SK, Coghill GG: **Neural network implementation using bit streams**. *IEEE Trans Neural Netw* 2007, **18**(5):1488-1504.

105. Zheng GX, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, Ziraldo SB, Wheeler TD, McDermott GP, Zhu J *et al*: **Massively parallel digital transcriptional profiling of single cells**. *Nat Commun* 2017, **8**:14049.

106. Shekhar K, Lapan SW, Whitney IE, Tran NM, Macosko EZ, Kowalczyk M, Adiconis X, Levin JZ, Nemesh J, Goldman M *et al*: **Comprehensive Classification of Retinal Bipolar Neurons by Single-Cell Transcriptomics**. *Cell* 2016, **166**(5):1308-1323 e1330.

107. Dong Z, Alterovitz G: **netAE: semi-supervised dimensionality reduction of single-cell RNA sequencing to facilitate cell labeling**. *Bioinformatics* 2021, **37**(1):43-49.

108. Newman ME: **Modularity and community structure in networks**. *Proc Natl Acad Sci U S A* 2006, **103**(23):8577-8582.

109. 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**(1):e9620.

110. 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**(5):501-513.

111. Hadsell R, Chopra S, LeCun Y: **Dimensionality reduction by learning an invariant mapping**. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06): 2006*: IEEE; 2006: 1735-1742.

112. Lieberman Y, Rokach L, Shay T: **CaSTLe–classification of single cells by transfer learning: harnessing the power of publicly available single cell RNA sequencing experiments to annotate new experiments**. *PloS one* 2018, **13**(10):e0205499.

113. Yuan Y, Bar-Joseph Z: **Deep learning for inferring gene relationships from single-cell expression data**. *Proc Natl Acad Sci U S A* 2019.

114. Alavi A, Ruffalo M, Parvangada A, Huang Z, Bar-Joseph Z: **A web server for comparative analysis of single-cell RNA-seq data**. *Nat Commun* 2018, **9**(1):4768.

115. Yevshin I, Sharipov R, Valeev T, Kel A, Kolpakov F: **GTRD: a database of transcription factor binding sites identified by ChIP-seq experiments**. *Nucleic Acids Research* 2016, **45**(D1):D61-D67.

116. Wang YXR, Waterman MS, Huang HY: **Gene coexpression measures in large heterogeneous samples using count statistics**. *Proceedings of the National Academy of Sciences of the United States of America* 2014, **111**(46):16371-16376.

117. Krishnaswamy S, Spitzer MH, Mingueneau M, Bendall SC, Litvin O, Stone E, Pe'er D, Nolan GP: **Systems biology. Conditional density-based analysis of T cell signaling in single-cell data**. *Science* 2014, **346**(6213):1250689.

118. Kanehisa M, Furumichi M, Tanabe M, Sato Y, Morishima K: **KEGG: new perspectives on genomes, pathways, diseases and drugs**. *Nucleic Acids Res* 2017, **45**(D1):D353-D361.

119. Fabregat A, Jupe S, Matthews L, Sidiropoulos K, Gillespie M, Garapati P, Haw R, Jassal B, Korninger F, May B *et al*: **The Reactome Pathway Knowledgebase**. *Nucleic Acids Res* 2018, **46**(D1):D649-D655.

120. Huynh-Thu VA, Irrthum A, Wehenkel L, Geurts P: **Inferring regulatory networks from expression data using tree-based methods**. *PloS one* 2010, **5**(9):e12776.

121. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES *et al*: **Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles**. *Proc Natl Acad Sci U S A* 2005, **102**(43):15545-15550.

122. Lotfollahi M, Wolf FA, Theis FJ: **scGen predicts single-cell perturbation responses**. *Nat Methods* 2019, **16**(8):715-721.

123. Duvenaud D, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, Aspuru-Guzik A, Adams R: **Advances in Neural Information Processing Systems 28**. *Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R, Eds* 2015:2224-2232.

124. Amodio M, Krishnaswamy S: **MAGAN: Aligning biological manifolds**. In: *International Conference on Machine Learning: 2018*: PMLR; 2018: 215-223.

125. Kang HM, Subramaniam M, Targ S, Nguyen M, Maliskova L, McCarthy E, Wan E, Wong S, Byrnes L, Lanata CM *et al*: **Multiplexed droplet single-cell RNA-sequencing using natural genetic variation**. *Nat Biotechnol* 2018, **36**(1):89-94.

126. Haber AL, Biton M, Rogel N, Herbst RH, Shekhar K, Smillie C, Burgin G, Delorey TM, Howitt MR, Katz Y *et al*: **A single-cell survey of the small intestinal epithelium**. *Nature* 2017, **551**(7680):333-339.

127. Hagai T, Chen X, Miragaia RJ, Rostom R, Gomes T, Kunowska N, Henriksson J, Park JE, Proserpio V, Donati G *et al*: **Gene expression variability across cells and species shapes innate immunity**. *Nature* 2018, **563**(7730):197-202.

128. Peng Y, Baulier E, Ke Y, Young A, Ahmedli NB, Schwartz SD, Farber DB: **Human embryonic stem cells extracellular vesicles and their effects on immortalized human retinal Muller cells**. *PLoS One* 2018, **13**(3):e0194004.

129. Stoeckius M, Hafemeister C, Stephenson W, Houck-Loomis B, Chattopadhyay PK, Swerdlow H, Satija R, Smibert P: **Simultaneous epitope and transcriptome measurement in single cells**. *Nat Methods* 2017, **14**(9):865-868.

130. Azizi E, Carr AJ, Plitas G, Cornish AE, Konopacki C, Prabhakaran S, Nainys J, Wu K, Kiseliovas V, Setty M *et al*: **Single-Cell Map of Diverse Immune Phenotypes in the Breast Tumor Microenvironment**. *Cell* 2018, **174**(5):1293-1308 e1236.

131. Chu LF, Leng N, Zhang J, Hou Z, Mamott D, Vereide DT, Choi J, Kendziorski C, Stewart R, Thomson JA: **Single-cell RNA-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm**. *Genome Biol* 2016, **17**(1):173.

132. Baron M, Veres A, Wolock SL, Faust AL, Gaujoux R, Vetere A, Ryu JH, Wagner BK, Shen-Orr SS, Klein AM *et al*: **A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure**. *Cell Syst* 2016, **3**(4):346-360 e344.

133. Camp JG, Sekine K, Gerber T, Loeffler-Wirth H, Binder H, Gac M, Kanton S, Kageyama J, Damm G, Seehofer D *et al*: **Multilineage communication regulates human liver bud development from pluripotency**. *Nature* 2017, **546**(7659):533-538.

134. Muraro MJ, Dharmadhikari G, Grun D, Groen N, Dielen T, Jansen E, van Gurp L, Engelse MA, Carlotti F, de Koning EJ *et al*: **A Single-Cell Transcriptome Atlas of the Human Pancreas**. *Cell Syst* 2016, **3**(4):385-394 e383.

135. Darmanis S, Sloan SA, Zhang Y, Enge M, Caneda C, Shuer LM, Hayden Gephart MG, Barres BA, Quake SR: **A survey of human brain transcriptome diversity at the single cell level**. *Proc Natl Acad Sci U S A* 2015, **112**(23):7285-7290.

136. Tirosh I, Venteicher AS, Hebert C, Escalante LE, Patel AP, Yizhak K, Fisher JM, Rodman C, Mount C, Filbin MG *et al*: **Single-cell RNA-seq supports a developmental hierarchy in human oligodendroglioma**. *Nature* 2016, **539**(7628):309-313.

137. Patel AP, Tirosh I, Trombetta JJ, Shalek AK, Gillespie SM, Wakimoto H, Cahill DP, Nahed BV, Curry WT, Martuza RL *et al*: **Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma**. *Science* 2014, **344**(6190):1396-1401.

138. Zahn H, Steif A, Laks E, Eirew P, VanInsberghe M, Shah SP, Aparicio S, Hansen CL: **Scalable whole-genome single-cell library preparation without preamplification**. *Nat Methods* 2017, **14**(2):167-173.

139. Pollen AA, Nowakowski TJ, Shuga J, Wang X, Leyrat AA, Lui JH, Li N, Szpankowski L, Fowler B, Chen P *et al*: **Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex**. *Nat Biotechnol* 2014, **32**(10):1053-1058.

140. Xin Y, Kim J, Okamoto H, Ni M, Wei Y, Adler C, Murphy AJ, Yancopoulos GD, Lin C, Gromada J: **RNA Sequencing of Single Human Islet Cells Reveals Type 2 Diabetes Genes**. *Cell Metab* 2016, **24**(4):608-615.

141. Yan L, Yang M, Guo H, Yang L, Wu J, Li R, Liu P, Lian Y, Zheng X, Yan J *et al*: **Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells**. *Nat Struct Mol Biol* 2013, **20**(9):1131-1139.

142. Chevrier S, Levine JH, Zanotelli VRT, Silina K, Schulz D, Bacac M, Ries CH, Ailles L, Jewett MAS, Moch H *et al*: **An Immune Atlas of Clear Cell Renal Cell Carcinoma**. *Cell* 2017, **169**(4):736-749 e718.

143. McGinnis CS, Patterson DM, Winkler J, Conrad DN, Hein MY, Srivastava V, Hu JL, Murrow LM, Weissman JS, Werb Z *et al*: **MULTI-seq: sample multiplexing for single-cell RNA sequencing using lipid-tagged indices**. *Nat Methods* 2019, **16**(7):619-626.

144. Han X, Wang R, Zhou Y, Fei L, Sun H, Lai S, Saadatpour A, Zhou Z, Chen H, Ye F *et al*: **Mapping the Mouse Cell Atlas by Microwell-Seq**. *Cell* 2018, **172**(5):1091-1107 e1017.

145. Hrvatin S, Hochbaum DR, Nagy MA, Cicconet M, Robertson K, Cheadle L, Zilionis R, Ratner A, Borges-Monroy R, Klein AM *et al*: **Single-cell analysis of experience-dependent transcriptomic states in the mouse visual cortex**. *Nat Neurosci* 2018, **21**(1):120-129.

146. Joost S, Zeisel A, Jacob T, Sun X, La Manno G, Lonnerberg P, Linnarsson S, Kasper M: **Single-Cell Transcriptomics Reveals that Differentiation and Spatial Signatures Shape Epidermal and Hair Follicle Heterogeneity**. *Cell Syst* 2016, **3**(3):221-237 e229.

147. Paul F, Arkin Y, Giladi A, Jaitin DA, Kenigsberg E, Keren-Shaul H, Winter D, Lara-Astiaso D, Gury M, Weiner A *et al*: **Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors**. *Cell* 2015, **163**(7):1663-1677.

148. Buettner F, Natarajan KN, Casale FP, Proserpio V, Scialdone A, Theis FJ, Teichmann SA, Marioni JC, Stegle O: **Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells**. *Nat Biotechnol* 2015, **33**(2):155-160.

149. Biase FH, Cao X, Zhong S: **Cell fate inclination within 2-cell and 4-cell mouse embryos revealed by single-cell RNA sequencing**. *Genome Res* 2014, **24**(11):1787-1796.

150. Deng Q, Ramskold D, Reinius B, Sandberg R: **Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells**. *Science* 2014, **343**(6167):193-196.

151. Klein AM, Mazutis L, Akartuna I, Tallapragada N, Veres A, Li V, Peshkin L, Weitz DA, Kirschner MW: **Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells**. *Cell* 2015, **161**(5):1187-1201.

152. Goolam M, Scialdone A, Graham SJL, Macaulay IC, Jedrusik A, Hupalowska A, Voet T, Marioni JC, Zernicka-Goetz M: **Heterogeneity in Oct4 and Sox2 Targets Biases Cell Fate in 4-Cell Mouse Embryos**. *Cell* 2016, **165**(1):61-74.

153. Kim JK, Kolodziejczyk AA, Ilicic T, Teichmann SA, Marioni JC: **Characterizing noise structure in single-cell RNA-seq distinguishes genuine from technical stochastic allelic expression**. *Nat Commun* 2015, **6**:8687.

154. Usoskin D, Furlan A, Islam S, Abdo H, Lonnerberg P, Lou D, Hjerling-Leffler J, Haeggstrom J, Kharchenko O, Kharchenko PV *et al*: **Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing**. *Nat Neurosci* 2015, **18**(1):145-153.

155. Zeisel A, Munoz-Manchado AB, Codeluppi S, Lonnerberg P, La Manno G, Jureus A, Marques S, Munguba H, He L, Betsholtz C *et al*: **Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq**. *Science* 2015, **347**(6226):1138-1142.

156. Yu Z, Liao J, Chen Y, Zou C, Zhang H, Cheng J, Liu D, Li T, Zhang Q, Li J *et al*: **Single-Cell Transcriptomic Map of the Human and Mouse Bladders**. *J Am Soc Nephrol* 2019, **30**(11):2159-2176.

157. Tusi BK, Wolock SL, Weinreb C, Hwang Y, Hidalgo D, Zilionis R, Waisman A, Huh JR, Klein AM, Socolovsky M: **Population snapshots predict early haematopoietic and erythroid hierarchies**. *Nature* 2018, **555**(7694):54-60.

158. Pijuan-Sala B, Griffiths JA, Guibentif C, Hiscock TW, Jawaid W, Calero-Nieto FJ, Mulas C, Ibarra-Soria X, Tyser RCV, Ho DLL *et al*: **A single-cell molecular map of mouse gastrulation and early organogenesis**. *Nature* 2019, **566**(7745):490-495.

159. Cao J, Spielmann M, Qiu X, Huang X, Ibrahim DM, Hill AJ, Zhang F, Mundlos S, Christiansen L, Steemers FJ *et al*: **The single-cell transcriptional landscape of mammalian organogenesis**. *Nature* 2019, **566**(7745):496-502.

160. Setty M, Tadmor MD, Reich-Zeliger S, Angel O, Salame TM, Kathail P, Choi K, Bendall S, Friedman N, Pe'er D: **Wishbone identifies bifurcating developmental trajectories from single-cell data**. *Nat Biotechnol* 2016, **34**(6):637-645.

161. Nestorowa S, Hamey FK, Pijuan Sala B, Diamanti E, Shepherd M, Laurenti E, Wilson NK, Kent DG, Gottgens B: **A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation**. *Blood* 2016, **128**(8):e20-31.

162. Cao J, Packer JS, Ramani V, Cusanovich DA, Huynh C, Daza R, Qiu X, Lee C, Furlan SN, Steemers FJ *et al*: **Comprehensive single-cell transcriptional profiling of a multicellular organism**. *Science* 2017, **357**(6352):661-667.

**Tables**

**Table 1.** Deep Learning algorithms reviewed in the paper

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **App** | **Algorithm** | **Models** | **Evaluation** | **Environment** | **Refs** |
| **Imputation** | | | | | |
|  | DCA | AE | DREMI | Keras, Tensorflow, scanpy | [15] |
|  | SAVER-X | AE+TL | t-SNE, ARI | R/sctransfer | [56] |
|  | DeepImpute | AE | MSE, Pearson’s correlation | Keras/Tensorflow | [17] |
|  | LATE | AE | MSE | Tensorflow | [57] |
|  | scGAMI | AE | NMI, ARI, HS and CS | Tensorflow | [58] |
|  | scIGANs | GAN | ARI, ACC, AUC, and F-score | PyTorch | [16] |
| **Batch correction** | | | | | |
|  | BERMUDA | AE+TL | kBET, entropy of Mixing, SI | PyTorch | [61] |
|  | DESC | AE | ARI, KL | Tensorflow | [64] |
|  | iMAP | AE+GAN | kBET, LISI | PyTorch | [67] |
| **Clustering, latent representation, dimension reduction, and data augmentation** | | | | | |
|  | Dhaka | VAE | ARI, Spearman Correlation | Keras/Tensorflow | [70] |
|  | scvis | VAE | KNN preservation, log-likelihood | Tensorflow | [73] |
|  | scVAE | VAE | ARI | Tensorflow | [75] |
|  | VASC | VAE | NMI, ARI, HS, and CS | H5py, keras | [76] |
|  | scDeepCluster | AE | ARI, NMI, clustering accuracy | Keras, Scanpy | [78] |
|  | cscGAN | GAN | t-SNE, marker genes, MMD, AUC | Scipy, Tensorflow | [82] |
| **Multi-functional models** (IM: imputation, BC: batch correction, CL: clustering) | | | | | |
|  | scVI | VAE | IM: L1 distance  CL: ARI, NMI, SI  BC: Entropy of Mixing | PyTorch, Anndata | [13] |
|  | LDVAE | VAE | Reconstruction errors | Part of scVI | [87] |
|  | SAUCIE | AE | IM: R2 statistics  CL: SI  BC: modified kBET  Visualization: Precision/Recall | Tensorflow | [14] |
|  | scScope | AE | IM: Reconstruction errors  BC: Entropy of mixing  CL: ARI | Tensorflow, Scikit-learn | [92] |
| **Doublet classification** | | | | | |
|  | Solo | VAE | Precision-recall, ROC | PyTorch | [93] |
| **Cell type Identification** | | | | | |
|  | DigitalDLSorter | DNN | Pearson correlation | R/Python/Keras | [96] |
|  | scCapsNet | CapsNet | Cell-type Prediction accuracy | Keras, Tensorflow | [103] |
|  | netAE | VAE | Cell-type Prediction accuracy, t-SNE for visualization | pyTorch | [107] |
|  | scDGN | DANN | Prediciton accuracy | pyTorch | [110] |
|  |  |  |  |  |  |
| **Function analysis** | | | | | |
|  | CNNC | CNN | AUROC, AUPRC, and accuracy | Keras, Tensorflow | [113] |
|  | scGen | VAE | Correlation, visualization | Tensorflow | [122] |

DL Model keywords: AE: autoencoder, AE+TL: autoencoder with transfer learning, AE: variational autoencoder, GAN: Generative adversarial network, CNN: convolutional neural network, DNN: direct neural network, DANN: domain adversarial neural network, CapsNet: capsule neural network

**Table 2a:** Simulated single-cell data/algorithms

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Title | Algorithm | # Cells | Simulation methods | Reference |
| Splatter | DCA, DeepImpute, PERMUDA, scDeepCluster, scVI, scScope, solo | ~2000 | Splatter/R | [85] |
| CIDR | sclGAN | 50 | CIDR simulation | [59] |
| NB+dropout | Dhaka | 500 | Hierachical model of NB/Gamma + random dropout |  |
| Bulk RNA-seq | SAUCIE | 1076 | 1076 CCLE bulk RNAseq + dropout conditional on expression level |  |
| SIMLR | scScope | 1 million | SIMLR, high-dimensional data generated from latent vector | [40] |

**Table 2b:** Human single-cell data sources used by different DL algorithms

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Title | Algorithm | Cell origin | # Cells | Data Sources | Reference |
| 68k PBMCs | DCA  SAVER-X  LATE, scVAE, scDeepCluster, scCapsNet, scDGN | Blood | 68,579 | 10X Single Cell Gene Expression Datasets |  |
| Human pluripotent | DCA | hESCs | 1,876 | GSE102176 | [128] |
| CITE-seq | SAVER-X | Cord blood mononuclear cells | 8,005 | GSE100866 | [129] |
| Midbrain and Dopaminergic Neuron Development | SAVER-X | Brain/ embryo ventral midbrain cells | 1,977 | GSE76381 | [81] |
| HCA | SAVER-X | Immune cell, Human Cell Atlas | 500,000 | HCA data portal |  |
| Breast tumor | SAVER-X | Immune cell in tumor micro-environment | 45,000 | GSE114725 | [130] |
| 293T cells | DeepImpute,  iMAP | Embryonic kidney | 13,480 | 10X Single Cell Gene Expression Datasets |  |
| Jurkat | DeepImpute,  iMAP | Blood/ lymphocyte | 3,200 | 10X Single Cell Gene Expression Datasets |  |
| ESC, Time-course | scGAN | ESC | 350, 758 | GSE75748 | [131] |
| Baron-Hum-1 | scGMAI, VASC | Pancreatic islets | 1,937 | GSM2230757 | [132] |
| Baron-Hum-2 | scGMAI, VASC | Pancreatic islets | 1,724 | GSM2230758 | [132] |
| Camp | scGMAI, VASC | Liver cells | 303 | GSE96981 | [133] |
| CEL-seq2 | PERMUDA, DESC | Pancreas/Islets of Langerhans |  | GSE85241 | [134] |
| Darmanis | scGMAI, sclGAN, VASC | Brain/cortex | 466 | GSE67835 | [135] |
| Tirosh-brain | Dhaka, scvis | Oligodendroglioma | >4800 | GSE70630 | [136] |
| Patel | Dhaka | Primary glioblastoma cells | 875 | GSE57872 | [137] |
| Li | scGMAI, VASC | Blood | 561 | GSE146974 | [64] |
| Tirosh-skin | scvis | melanoma | 4645 | GSE72056 | [71] |
| xenograft 3, and 4 | Dhaka | Breast tumor | ~250 | EGAS00001002170 | [138] |
| Petropoulos | VASC/netAE | Human embryos | 1,529 | E-MTAB-3929 |  |
| Pollen | scGMAI, VASC |  | 348 | SRP041736 | [139] |
| Xin | scGMAI, VASC | Pancreatic cells (α-, β-, δ-) | 1,600 | GSE81608 | [140] |
| Yan | scGMAI, VASC | embryonic stem cells | 124 | GSE36552 | [141] |
| PBMC3k | VASC, scVI | Blood | 2,700 | SRP073767 | [105] |
| CyTOF, Dengue | SAUCIE | Dengue infection | 11 M, ~42 antibodies | Cytobank, 82023 | [14] |
| CyTOF, ccRCC | SAUCIE | Immunue profile of 73 ccRCC patients | 3.5 M, ~40 antibodies | Cytobank: 875 | [142] |
| CyTOF, breast | SAUCIE | 3 patients |  | Flow Repository: FR-FCM-ZYJP | [130] |
| MULTI-seq | Solo | scRNA-seq, snRNA-seq | multiplex samples | GSE129578 | [143] |
| Chung, BC | DigitalDLSorter | Breast tumor | 515 | GSE75688 | [97] |
| Li, CRC | DigitalDLSorter | Colorectal cancer | 2,591 | GSE81861 | [98] |
| Pancreatic datasets | scDGN | Pancreas | 14693 | SeuratData |  |
| Kang, PBMC | scGen | PBMC stimulated by INF-b | ~15,000 | GSE96583 | [125] |
|  |  |  |  |  |  |

**Table 2c:** Mouse single-cell data sources used by different DL algorithms

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Title** | **Algorithm** | **Cell origin** | **# Cells** | **Data Sources** | **Reference** |
| **Brain cells from E18 mice** | DCA, SAUCIE | Brain Cortex | 1,306,127 | 10x: Single Cell Gene Expression Datasets |  |
| **Midbrain and Dopaminergic Neuron Development** | SAVER-X | Ventral Midbrain | 1907 | GSE76381 | [81] |
| **Mouse cell atlas** | SAVER-X |  | 405,796 | GSE108097 | [144] |
| **neuron9k** | DeepImpute | Cortex | 9128 | 10x: Single Cell Gene Expression Datasets |  |
| **Mouse Visual Cortex** | DeepImpute | Brain cortex | 114601 | GSE102827 | [145] |
| **murine epidermis** | DeepImpute | Epidermis | 1422 | GSE67602 | [146] |
| **myeloid progenitors** | LATE  DESC, SAUCIE | Bone marrow | 2,730 | GSE72857 | [147] |
| **Cell-cycle** | sclGAN | mESC | 288 | E-MTAB-2805 | [148] |
| **A single-cell survey** |  | Intestine | 7721 | GSE92332 | [126] |
| **Tabula Muris** | iMAP | Mouse cells | >100K |  |  |
| **Baron-Mou-1** | VASC | Pancreas | 822 | GSM2230761 | [132] |
| **Biase** | scGMAI, VASC | Embryos/SMARTer | 56 | GSE57249 | [149] |
| **Biase** | scGMAI, VASC | Embryos/Fluidigm | 90 | GSE59892 | [149] |
| **Deng** | scGMAI, VASC | Liver | 317 | GSE45719 | [150] |
| **Klein** | VASC  scDeepCluster  sclGAN | Stem Cells | 2,717 | GSE65525 | [151] |
| **Goolam** | VASC | Mouse Embryo | 124 | E-METAB-3321 | [152] |
| **Kolodziejczyk** | VASC | mESC | 704 | E-MTAB-2600 | [153] |
| **Usoskin** | VASC | Lumbar | 864 | GSE59739 | [154] |
| **Zeisel** | VASC, scVI, SAUCIE, netAE | Cortex, hippocampus | 3,005 | GSE60361 | [155] |
| **Bladder cells** | scDeepCluster | Bladder | 12,884 | GSE129845 | [156] |
| **HEMATO** | scVI | Blood cell | >10,000 | GSE89754 | [157] |
| **retinal bipolar cells** | scVI, scCapsNet  SAUCIE | retinal | ~25,000 | GSE81905 | [106] |
| **Embryo at 9 time points** | LDAVE | embryos from E6.5 to E8.5 | 116,312 | GSE87038 | [158] |
| **Embryo at 9 time points** | LDAVE | embryos from E9.5 to E13.5 | ~2 millions | GSE119945 | [159] |
| **CyTOF,** | SAUCIE | Mouse thymus | 200K, ~38 antibodies | Cytobank: 52942 | [160] |
| **Solo** | solo | Mouse kidneys | ~8,000 | GSE140262 | [93] |
| **Nestorowa** | netAE | hematopoietic stem and progenitor cells | 1,920 | GSE81682 | [161] |
| **small intestinal epithelium** | scGen | Infected with Salmonella and worm H. polygyrus | 1,957 | GSE92332 | [126] |

**Table 2d:** Single-cell data derived from other species

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Title** | **Algorithm** | **Species** | **Tissue** | **# Cells** | **SRA/GEO** | **Reference** |
| **Worm neuron cells**1 | scDeepCluster | C. elegans | Neuron | 4,186 | GSE98561 | [162] |
| **Cross species, stimulation with LPS and dsRNA** | scGen | Mouse, rat, rabbit, and pig | bone marrow-derived phagocyte | 5,000 to 10,000 /species | 13 accessions in ArrayExpress | [127] |

1. Processed data is available at <https://github.com/ttgump/scDeepCluster/tree/master/scRNA-seq%20data>

**Table 2e:** Large single-cell data source used by various algorithms

|  |  |  |
| --- | --- | --- |
| **Title** | **Sources** | **Notes** |
| **10X Single-cell gene expression dataset** | <https://support.10xgenomics.com/single-cell-gene-expression/datasets> | Contains large collection of scRNA-seq dataset generated using 10X system |
| **Tabula Muris** | <https://tabula-muris.ds.czbiohub.org/> | compendium of scRNA-seq data from mouse |
| **HCA** | <https://data.humancellatlas.org/> | Human single-cell atlas. |
| **scQuery** | <https://scquery.cs.cmu.edu/> | A web server cell type matching and key gene visualization. It is also a source for scRNA-seq collection (processed with common pipeline). |
| **SeuratData** | <https://github.com/satijalab/seurat-data> | List of datasets, including PBMC and human pancreatic islet cells. |