Deep Learning in Single-cell Analysis

DYLAN MOLHO, JIAYUAN DING, and WENZHUO TANG, Michigan State University, East Lansing, USA
ZHAOHENG LI, University of Washington, Seattle, USA
HONGZHI WEN, Michigan State University, East Lansing, USA
YIXIN WANG, Stanford University, Stanford, USA
JULIAN VENEGAS, Michigan State University, East Lansing, USA
WEI JIN, Emory University, Atlanta, USA
RENMING LIU and RUNZE SU, Michigan State University, East Lansing, USA
PATIOH DANAHER, NanoString Technologies, Seattle, USA
ROBERT YANG, Johnson & Johnson, New Brunswick, USA
YU LEO LEI, University of Michigan, Ann Arbor, USA
YUYING XIE and JILIANG TANG, Michigan State University, East Lansing, USA

Single-cell technologies are revolutionizing the entire field of biology. The large volumes of data generated by single-cell technologies are high dimensional, sparse, and heterogeneous and have complicated dependency structures, making analyses using conventional machine learning approaches challenging and impractical. In tackling these challenges, deep learning often demonstrates superior performance compared to traditional machine learning methods. In this work, we give a comprehensive survey on deep learning in single-cell analysis. We first introduce background on single-cell technologies and their development, as well as fundamental concepts of deep learning including the most popular deep architectures. We present an overview of the single-cell analytic pipeline pursued in research applications while noting divergences due to data sources or specific applications. We then review seven popular tasks spanning different stages of the single-cell analysis pipeline, including multimodal integration, imputation, clustering, spatial domain identification, cell-type deconvolution, cell segmentation, and cell-type annotation. Under each task, we describe the most recent developments in classical and deep learning methods and discuss their advantages and disadvantages. Deep learning tools and benchmark datasets are also summarized for each task. Finally, we discuss the future directions and the most recent challenges. This survey will serve as a reference for biologists and computer scientists, encouraging collaborations.

This research is supported by the National Science Foundation (NSF) under grant numbers CNS 2246050, IIS1845081, IIS2212032, IIS2212144, IOS2107215, DUE 2234015, DRL 2025244, IOS2035472, and IIS2123260; the National Institutes of Health (NIH) under grant numbers U01DE029255, R01DE026728, R01HL166508, and U01DE033330; the Army Research Office (ARO) under grant number W911NF-21-1-0198; the Home Depot; Cisco Systems Inc.; Amazon Faculty Award; Johnson & Johnson; the JP Morgan Faculty Award; and SNAP.

D. Molho, J. Ding, and W. Tang equal contribution.

Authors’ addresses: D. Molho, J. Ding, W. Tang, H. Wen, J. Venegas, R. Liu, R. Su Y. Xie, and J. Tang, Michigan State University, East Lansing, MI, USA; e-mails: molhodyl@msu.edu, dingjia5@msu.edu, tangwen2@msu.edu; Z. Li, University of Washington, Seattle, WA, USA; Y. Wang, Stanford University, Palo Alto, CA, USA; W. Jin, Emory University, Atlanta, GA, USA; e-mail: wei.jin@emory.edu; P. Danaher, NanoString Technologies, Seattle, WA, USA; R. Yang, Johnson & Johnson, Boston, MA, USA; Y. L. Lei, University of Michigan, Ann Arbor, MI, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM Trans. Intell. Syst. Technol., Vol. 15, No. 3, Article 40. Publication date: March 2024.
INTRODUCTION

As the basic building blocks of life, cells assume dynamic and complex functional states to inform higher-order structures [215, 242]. Towards that end, the advance of single-cell sequencing and imaging technologies has revolutionized the investigation of the gene-expression behaviors of cells. The advent of single-cell sequencing technology occurred in the early 1990s for complementary DNA (cDNA) [23, 75]. However, it was not until 2009, with the creation of the first single-cell RNA sequencing (scRNA-seq) method [302], that a true paradigm shift in the field occurred. Since then, steady progress in the creation of new next-generation sequencing platforms has led to over 100 currently existing techniques for single-cell sequencing [180, 331, 339]. These technologies measure a diverse collection of cell features including DNA sequences and epigenetic features, RNA expression, and profiles of surface proteins. Recent technological advances have also enabled the augmentation of these features with additional data, e.g., multimodal sequencing platforms and spatial transcriptomic technology.

This paradigm shift comes from the quantity of available data using high throughput methods [159, 298]. For example, one bulk tissue RNAseq data [177] can only quantify the average gene expressions of a group of cells ignoring the cellular heterogeneity and hence can serve as one sample for downstream analyses. In contrast, single-cell sequencing technologies generate tens of thousands to millions of samples/cells in a given experiment. Deep learning methods, which have consistently shown cutting-edge performance in various big data applications [70, 249], have fertile new ground for research that pushes the frontiers of biological science. Studies in single-cell data continue to expand exponentially and obtain new insights into immunology, oncology, developmental biology, pharmacology, and many other disciplines, just to name a few areas of applications [40, 96, 318].

Despite the success of single-cell data in numerous applications, difficulties arise due to the complexity of the data, which requires advanced analysis pipelines with a number of steps. Single-cell data preprocessing includes many stages of data pruning, normalization, and often challenging machine learning tasks like batch effect correction, data imputation, or dimensionality reduction. Moreover, specialized types of single-cell data require further processing such as multimodal data integration and cell-type deconvolution for spatial transcriptomics. These steps are crucial to facilitate downstream tasks ranging from clustering and cell annotation, disease prediction, and identification of gene coexpression networks to identification of developmental trajectories of cells transitioning between states [167]. For tasks with clear evaluation metrics, deep learning often achieves top performance against other classical machine learning techniques [229]. Deep learning can uniquely leverage its diverse architectures to capture networks of interdependencies between genes that alter other genes’ expression levels [14] and cells that communicate with other cells through mechanisms like ligand-receptor pairs [173]. Due to the richness of deep learning architectures and the customization of hyper-parameters and loss functions, deep learning models can
be more readily tailored to particular tasks in single-cell analysis compared to other machine learning methods. Deep learning has already rapidly proliferated throughout the field, but due to the multidisciplinary nature of the work, many remain unaware of this burgeoning area of research. We write this survey as a bridge between two large research communities in single-cell biology and computer science. We provide background on deep learning to those in biology and less familiar with machine learning modeling, and also provide some history and summary of single-cell data to computer scientists who are looking for novel applications for their methods.

In this survey, we review methods in the emerging use of deep learning for single-cell biology applications. The key messages of the survey are summarized below:

- The background knowledge on single-cell technologies and their evolution is exhaustively described in advance, including single-modality profiling, multi-modality profiling, and single-cell spatial transcriptomics.
- The fundamental concepts underlying a variety of deep learning models that have been widely used for single-cell research are presented in advance. What’s more, a list of popular deep learning tools is also provided for users’ easy pick-up.
- In addition to task definition and evaluation, the most recent breakthroughs in both traditional (non-deep learning) and deep learning methods (particularly Graph Neural Networks (GNNs)) is systematically reviewed for each task in this article.
- A comprehensive summary of the standard benchmark datasets and typical tools or baseline methodologies with accessible links by each task is presented.
- To the best of our knowledge, this is the most comprehensive survey of deep learning in single-cell analysis, involving around 200 models and over 30 benchmark datasets.

The rest of the survey is structured as follows. In Section 2, we discuss the history and major technologies for single-cell sequencing. In Section 3, we give an overview of deep learning concepts and popular deep architectures. Due to the categorization of the tasks involved in single-cell analysis, we group our review of deep learning methods by tasks. We first give an overview of the pipeline in Section 4. Then, we describe the individual task objectives and highlight the alternative machine learning methods used for the task before detailing the deep learning methods.

2 SINGLE-CELL TECHNOLOGIES

The goal of mapping genotypes to phenotypes presents a multitude of challenges to biologists performing transcriptome analysis [126]. Earlier bulk sampling methods were able to access the transcriptome of a group of cells, but they were unsuccessful in capturing the variation and unpredictability among individual cells, even if they belonged to the same cell type [97, 165]. This is because these methods produced average profiles for the entire cell population. Compared to bulk sampling technologies, single-cell technologies provide a higher resolution of cell differences and can attribute biological behaviors to individual cells [232, 287]. As an increasing array of single-cell profiling technologies has expanded the possibilities for computational biology analysis, more previously hidden signals of interest have become apparent. For instance, can we quantify how different cellular mechanisms interact with each other? How does the cell-cell locality proximity affect cellular functions? The introduction of multi-modal profiling technologies offers insights into addressing the first question [8, 203, 263, 289], while single-cell spatial transcriptomics reveals the underlying spatial relationships between cells [218, 222, 294]. In the following sections, we will focus on the aforementioned three categories of single-cell technologies, i.e., single-modality profiling in Section 2.1, multi-modality profiling in Section 2.2, and spatial transcriptomics in Section 2.3. We summarize a timeline of their development in Figure 1. In addition, we provide the taxonomy of the computational tasks mentioned in this work in Figure 10.
2.1 Single-modality Profiling

Sequencing technology was first developed by Eberwine et al. [75] and Iscove et al. [23] by expanding the cDNA. However, it wasn’t until the creation of scRNA-seq in 2009 [302] that single-cell methods truly gained major traction. Since then, a few major branches in single-cell technologies have emerged, targeting different aspects of cells, such as RNA in 2009 [302], DNA methylation in 2013 [107], protein in 2015, DNA accessibility in 2015, and histone modifications in 2021 [16].

Single-cell data is often given in the form of a matrix, with features (e.g. genes, proteins, or DNA interval) corresponding to the columns and each cell as a row, as shown in Figure 2.

Since its creation, scRNA-seq has had remarkable success in a number of different applications, such as cell developmental studies, classifying cell types, and gene regulation. For scRNA-seq, isolation of the cell is the first step for obtaining transcriptome information. Many technologies are differentiated according to their means of cell isolation before sequencing occurs. Earlier methods using serial dilution or robotic micromanipulation have low efficiency and throughput [24] when compared to more recent methods using microfluidic technologies [341]. One promising microfluidic technique for single-cell isolation is using microdroplets [309], which creates the uniform dispersion of water droplets in a medium of oil, allowing the separation of cells into individual droplets. While commercial microfluidic platforms like Fluidigm C1, ICELL8, and Chromium can benefit from high throughput, they face the challenge of high cost and often the requirement of uniform cell size in the sample. Once a cell is separated and lysed, messenger RNAs in this cell are reverse transcribed into more stable cDNAs with a unique cell “barcode.” The cDNAs are then amplified via Polymerase Chain Reaction (PCR) for better data capture before sequencing, which tends to introduce bias due to the uneven amplification efficiency. Therefore, besides the unique barcodes, the cDNA molecules in a cell are also given a Unique Molecular Identifier (UMI) to correct the amplification bias by collapsing the reads with the same UMI into one read. After
debiasing, sequence reads are mapped to the genome and are grouped into genes for the creation of a count matrix \[331\].

Beyond recording RNA expression levels in a cell, technology may also capture information about the chromatin accessibility of a cell’s chromosome. Eukaryotic genomes are hierarchically packaged into chromatin \[160\], and this packaging plays a central role in gene regulation \[161\]. Buenrostro et al. created a means for sampling the epigenome at the single-cell level through the **Assay for Transposase Accessible Chromatin using sequencing (ATAC-seq)** \[26\] in 2013. ATAC-seq allows the identification of accessible DNA, i.e., the nucleosome-free regions of the genome \[120\]. DNA accessibility within the genome can be used to identify regulatory elements in different cell types that cause the activation or repression of gene expression \[310\]. scATAC-seq produces a count matrix with a number of reads per open chromatin regions, which lead to very large matrices with hundreds of thousands of regions. Furthermore, the data is known to be very sparse, where it is common to have the non-zero entries make up less than 3% of the data \[179\].

Gene expression can also be affected by a number of additional factors that are investigated under the umbrella of epigenetics, which studies mechanisms like DNA methylation and histone modification that do not change the DNA sequence but can change gene activity and expression \[19\]. DNA methylation occurs when methyl groups are bonded to the DNA molecule, which can repress gene transcription and is associated with a number of key biological processes \[226\]. In mammals, DNA methylation occurs most often in particular portions of the base pair sequence, namely CG (denoted CpG) portions where a cytosine is followed by a guanine \[275\]. New technologies developed in the past decade for the profiling of DNA methylation use **bisulfite sequencing (scBS-seq)** \[85, 278\] or **reduced representation bisulfite sequencing (scRRBS-seq)** \[86, 125\] at a single-cell resolution. The output data for these are binary, indicating regions that are methylated by a 1, while 0 indicates no methylation.

### 2.2 Multi-modality Profiling

In addition to the cell transcriptome and epigenome, cell proteome is another focus of single-cell technologies, which consist of the proteins that are encoded by the mRNA of the cell. Comprehensive measurements of a cell’s proteome are integral to understanding how the genes respond to environmental changes, as well as for predicting cellular behavior since proteins are the functional units responsible for most of the cellular processes. While single-cell sequencing techniques for transcriptome measurements have widely proliferated, single-cell proteomics methods have made slower progress. Unlike most of the sequencing technologies that have a standard process, proteomic measurements are often bespoke and designed for specific applications \[324\]. However, some technologies developed have made significant strides in not only capturing protein information of cells but also combining this with mRNA measurements. Specifically, **Cellular Indexing of Transcriptomes and Epitopes by Sequencing (CITE-seq)**, a new technology introduced in 2017, simultaneously sequences mRNA and measures the surface proteins on a cell \[289\]. The method can sample over 1,000 genes and 80 proteins per cell but, like many other sequencing techniques, suffers from high noise. In addition, CITE-seq is incapable of detecting intracellular proteins \[15\].

The repertoire of multi-modal single-cell technologies bridges RNA expression not only to protein but also to DNA methylation, chromatin accessibility, and histone modifications. One of the first methods to simultaneously sequence RNA and chromatin accessibility is a droplet-based method named SNARE-seq. Published in 2019, it uses Tn5 transposase to capture accessible chromatin and creates shared barcodes between RNA and accessible regions \[44\]. The same year, paired-seq raised the throughput by two orders by combining a ligation-based combinatorial indexing strategy and an amplify-and-split library dedicating method \[373\]. SHARE-seq \[203\] further...
increased the throughput and resolution by adapting Paired-seq and SPLiT-seq [263], a scRNA-seq technology. In 2020, 10X also released 10X Multiome, a commercialized product for the joint profiling of RNA and chromatin accessibility. Beyond the co-profiling of RNA and chromatin accessibility, scM&T-seq [8], based on G&T-seq [206], allows for parallel analysis of single-cell RNA and DNA methylation. Another emerging field is the joint profiling of RNA and histone modifications. Example technologies in this category include Paired-Tag [372] and CoTECH [345]; both became available in 2021.

2.3 Single-cell Spatial Transcriptomics

Single-cell technologies that capture transcriptomic, proteomic, or epigenetic information do so with great precision but with the loss of spatial information of the cells within the tissues. However, the cells’ relative locations within tissue is critical to understanding normal development and disease pathology. With spatial transcriptomic technologies, researchers are able to measure transcriptomics and leverage the spatial information or relative locations of cells in a tissue for better performing downstream tasks [12, 58, 211, 225, 306, 327, 335]. For example, motivated by the fact that a pair of ligand and receptor with closer distance is easier to bind, HoloNet [173] builds up a directed graph based on the expression of ligand–receptor gene lists and the physical distance between the sender cell and receiver cell to represent cell–cell communication events. However, the early generations of spatially resolved profiling technologies are not at the single-cell resolution but instead sampled in groups called “spots,” which capture several cells. It requires additional work to determine the cell type proportion in spots, a process called cell type deconvolution. Alternatively, many cell imaging platforms provide RNA spatial information at the cellular and subcellular level, but the individual cells must be identified through cell segmentation methods.

Major technologies or platforms for spatial transcriptomics include multiplexed error-robust fluorescence in-situ hybridization (MERFISH), sequential fluorescence ISH (seqFISH+), Slide-Seq, Visium by 10x [294], GeoMx Digital Spatial Profiler (DSP) [218] by NanoString, and CosMx Spatial Molecular Imager (SMI) by NanoString. MERFISH [222], first introduced in 2015, is a single-molecule-fluorescence-in-situ-hybridization (smFISH)-based technology that can be applied to fresh-frozen samples to provide subcellular resolution. While traditionally the procedure of these smFISH-based technologies is complex, a number of commercialized platforms have emerged recently, such as Vizgen, Rebus Esper, Molecular Cartography, and Resolve Biosciences [227], which allow more convenient sequencing of spatial transcriptomics at a lower cost. As an alternative to MERFISH, seqFish+ [80, 196, 271] employs “pseudocolor” as a combination of colors to increase the amount of detectable transcripts [252].

Beyond early in situ hybridization methods, a number of sequence-based technologies have emerged. Closely related to scRNA-seq technologies, these sequencing-based methods barcode RNAs such that each read can be mapped to its corresponding spatial location through the associated barcodes. The rest of the sequencing read is mapped to the genome to identify the transcript of origin, collectively generating a gene-expression matrix. Stahl et al. [284] first proposed this method, which has been adapted by commercial platforms such as 10x Visium. 10x Visium fixes spatially barcoded oligos to each spot in a capture slide (area 6.5 mm²), with the barcoding done through DNA extension and reverse transcription for formalin-fixed paraffin-embedded tissues (FFPE) and fresh frozen tissues, respectively. In particular, the 10x Visium expression slide contains four capture slides, each with an area of 6.5 mm² where fresh frozen or FFPE tissues are placed. Each of the capture slides contains a grid of approximately 5,000 barcoded spots that are 55 μm in diameter with a center-to-center distance of 100 μm between any two adjacent spots. On average, there are 1 – 10 cells in each of these spots, and ~18,000 unique genes in human (~20,000 in mouse) can be quantified. Another major sequencing-based technology is Slide-Seq.
which captures mRNA by placing barcoded beads on slides, which achieves a high resolution of 10 micron. Technological innovations further improved sequencing resolution in recent years. For instance, high-definition spatial transcriptomics (HDST) [258] uses wells rather than slides, whereas built upon Slide-Seq, Slide-seqV2 [288] raised the resolution to near-cellular level while reaching RNA capture efficiency of roughly 50% of scRNA-sequencing. Finally, spatio-temporal enhanced resolution omics sequencing (Stereo-seq) [37] deposits barcoded DNA nanoballs in patterned arrays to achieve single-cell resolution while maintaining high sensitivity.

While 10x Visium and Slide-Seq do not profile at cellular resolution, Nanostring’s GeoMx DSP is capable of cellular resolution through user-drawn profiling regions. GeoMx DSP uses PC-linker to link barcodes via antibodies to proteins and RNA for identification. The spatial regions of interest (ROI) on the tissue are flexible and can be user-defined, or with pre-defined layouts (such as a square grid). During imaging, the DSP barcodes from each ROI are UV-cleaved and collected for sequencing, and the spatial information is recorded. Due to the flexibility of the ROI definitions, the ROI can be a range of sizes, from a single cell to hundreds of cells. The RNA assay can quantify >18,000 target genes, and the protein assay can quantify >96 proteins.

Though GeoMX can produce cellular resolution sequencing, its scalability is limited. The most recent platform, CosMx Spatial Molecular Imager (SMI) [171], is able to profile consistently at single-cell, and even subcellular, resolution. CosMx SMI follows much of the initial protocol as GeoMx DSP, with barcoding and ISH hybridization. However, the SMI instrument performs 16 cycles of automated cyclic readout, and in each cycle the set of barcodes (readouts) are UV cleaved and removed. These cycles of hybridization and imaging yield spatial resolved profiling of RNA (>980 target genes) and protein (>80 validated proteins) at single-cell (∼10μm) and subcellular (∼1μm) resolution.

Multiplex imaging technologies have significantly advanced higher spatial resolution for single-cell profiling. Spatially resolved transcriptomic data, along with corresponding imaging data, enables single-cell or even subcellular analysis on both spatially morphological and pixel resolution information. Recently, antibody-based multiplexed imaging methods have dominated the multiplexing approaches, as they can capture cellular organization and tissue phenotypic heterogeneity at the protein level. They utilize various protein markers for cellular identification. Immunohistochemistry (IHC) [57], first reported in 1942, is one of the most commonly used multiplexed imaging methods. It uses appropriately labeled antibodies to bind specifically to their target antigens in situ (in the original site), which can be better captured by current light or fluorescence microscopy. Due to the limited protein readouts, methods including multiplexed immunofluorescence (MxIF) [93] and cyclic immunofluorescence (CyCIF) [181, 182] were proposed to add more new antibodies in multiple rounds of staining. Another imaging platform, Co-Detection by IndeXing (CODEX) [98], is designed for up to 40 proteins using cyclic detection of DNA-indexed antibody panels. Imaging mass cytometry (IMC) [95] is an evolutionary technology that leverages mass spectroscopy to obtain images from tissues with 40+ labels simultaneously. This vastly reduces data noise and enhances the multiplex capability. Multiplexed ion beam imaging (MIBI) [147] is also performed by imaging tissues with secondary ion mass spectrometry based on metal-labeled antibodies. These multiplexed imaging tools provide high-dimensionality imaging assays at the single-cell level and enable analyzing and understanding of the single-cell function and tissue structure.

3 CONCEPTS IN DEEP LEARNING

The aforementioned advances in single-cell technologies bring in large-scale datasets, paving the way for the application of deep learning methods. Over recent years, deep learning has
Fig. 3. Taxonomy of the deep learning methods.

Table 1. A Summary of Existing Popular Frameworks for Deep Learning

| Tool              | Category          | Language   | Availability                                      |
|-------------------|-------------------|------------|---------------------------------------------------|
| Pytorch [241]     | Deep Learning     | C++, Python| https://github.com/pytorch/pytorch                 |
| Tensorflow [2]    | Deep Learning     | C++, Python| https://github.com/tensorflow/tensorflow          |
| Keras [49]        | Deep Learning     | C++, Python| https://github.com/keras-team/keras               |
| MXNet [46]        | Deep Learning     | C++, Python| https://github.com/apache/incubator-mxnet        |
| DGL [329]         | Graph Learning    | Python     | https://github.com/dmlc/dgl                        |
| PyG [87]          | Graph Learning    | Python     | https://github.com/pyg-team/pytorch_geomeric      |
| networkx [109]    | Graph Learning    | Python     | https://github.com/networkx/networkx              |
| PyTorch-VAE [295] | Variational AutoEncoders | Python | https://github.com/AntixK/PyTorch-VAE             |
| OpenCV [60]       | Image Processing  | C++, Python| https://github.com/opencv/opencv                  |

demonstrated significant success in quantitatively analyzing single-cell data by employing complex topologies for artificial neural networks [83, 201]. In this section, we present an overview of the fundamental concepts underlying a number of deep learning models that have been widely used for single-cell research. We include the taxonomy of the downstream methods in Figure 3. More methods can be found in Appendix A. We summarize popular frameworks for deep learning in Table 1.

3.1 Multi-layer Perceptrons

Multi-layer perceptrons (MLPs), sometimes referred to as feedforward neural networks, are the simplest case of an artificial neural network (ANN). The term “feedforward networks” is used to distinguish the architecture from recurrent neural networks, where the former has connections between the nodes that do not form a cycle, while the latter does. A feedforward neural network consists of multiple layers of neurons. Each neuron, as shown in Figure 4(a), is similar to a
perceptron [212], which performs linear transformation over an input vector \( x \) and adds a non-linear activation function before output. This is why it is often recognized as an MLP.

Formally, we denote the input of a neuron layer as a vector \( x \). Each dimension of \( x \) is fed to a neuron. A layer of neurons can thus be formulated as

\[
x^{(i+1)} = \sigma(W^{(i)}x^{(i)} + b^{(i)}),
\]

where \( x^{(i)} \in \mathbb{R}^d \) is the input vector that feeds to the layer \( i \), \( W_i \in \mathbb{R}^{h \times d} \) and \( b_i \in \mathbb{R}^h \) are a weight matrix and a bias vector to be learned in the layer \( i \), and \( \sigma \) is a non-linear activation function. \( x^{(i+1)} \in \mathbb{R}^h \) is the output vector of layer \( i \), which is also used as the input of the next layer \( i + 1 \).

Typically we stack multiple layers to form an MLP as shown in Figure 4(b). The first layer is referred to as the input layer, the last as the output layer, and the layers in between as the hidden layers. The output of each layer is fed to the next layer, and thus the connections between neurons of adjacent layers are dense, which are also known as “fully connected layers” or “dense layers.” Generally, a feedforward neural network as a whole is a non-linear transformation from the input space to the output space. For example, in Figure 4(b), the input of the whole network is vector \( x \) at the left side, and the output of the network is from the output layer at the right side, which is vector \( y \).

### 3.2 Autoencoders

**Autoencoders (AEs)** [264] are neural networks to reduce dimensionality or reduce noise from different types of data via reconstructing the original input. Conventional AEs consist of two parts: an encoder and a decoder, as shown in Figure 5. The encoder is to map the input data \( X \) to a latent space \( Z \), which is frequently significantly smaller than the original input space so that only the significant features of data variation are captured in the latent space. The decoder will map the latent vector \( Z \) back to the original input dimension space and recover the original data as \( X' \). We take an AE with only two layers as the simplest form as an example. The encoder with a single hidden layer maps the input data \( X \in \mathbb{R}^{N \times K} \) into the latent space representation \( Z \in \mathbb{R}^{N \times P} \) as defined in Equation (2), where \( N \) indicates the number of input samples, \( K \) is the original feature dimension, and \( P \) is the compressed feature dimension:

\[
Z = \sigma(XW + b),
\]

where \( W \) is the weight matrix, \( b \) is the bias vector, and \( \sigma \) is the activation function such as ReLU. The decoder is responsible for mapping the latent representation \( Z \) back to the original input space.
defined as follows:

\[ X' = \sigma(ZW' + b'), \quad (3) \]

where \( X' \in \mathbb{R}^{N \times K} \) is the reconstructed data points. The objective of typical AEs is to minimize the distance between the original input \( X \) and the recovered data points \( X' \) defined as

\[ L = \|X - X'|^2. \quad (4) \]

In this simplest form, the parameters \( \theta = \{W, b, W', b'\} \) are optimized during training [265].

Note that the model selection of the encoder and decoder of the AEs is very flexible and does not have to be consistent. For example, the encoder can consist of Graph Neural Networks [268], while the decoder can be multilayer perceptron [248] in the same AE. Furthermore, there are numerous architectural variants of AEs [135, 314, 340, 361]. One popular variant is the variational AE (VAE) [150, 151]. VAEs account for the irregularity of the latent space by returning a distribution over the latent space as opposed to a single point. To achieve this goal, a regularization term for the returning distribution is incorporated into the loss function to ensure a more organized latent space.

### 3.3 Generative Adversarial Networks

**Generative Adversarial Networks (GANs) [104]** are a useful tool to generate realistic synthetic data. As illustrated in Figure 6, the general idea of GANs is to train two neural networks that compete against one another in a two-player game. The two networks in a GAN are referred to as the generator (\( G \)) and the discriminator (\( D \)). The training objective of \( D \) is to discriminate between pseudo data and real data. On the other hand, the goal of \( G \) is to generate pseudo samples that have the same distribution as the real data and thus can fool the discriminator into not being able to differentiate pseudo data from real data. Formally, the objective function can be written as

\[
\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z}(z)[\log(1 - D(G(z)))], \quad (5)
\]

where \( p_z(z) \) is defined as a prior on input noise variables, \( G \) is a generator that generates pseudo data given a random vector \( z \), and \( D \) is a discriminator that is optimized to output one when given real data \( x \) and output zero when given pseudo data \( G(z) \). Overall, the minmax objective function is optimizing \( D \) and \( G \) simultaneously, and finally, an optimal generator is produced.

Typically, the parameters of \( G \) and \( D \) are iteratively updated during the training, where eventually both parameters would be optimized. During the process, the generator is gradually trained to generate realistic data that can cheat the discriminator to be classified as real samples, while the discriminator also progressively becomes stronger at distinguishing between real and pseudo data. Compared to other generative models, the adversarial training between \( G \) and \( D \) networks can get rid of the assumptions on the prior distribution, and thus can learn any distribution.
3.4 Convolutional Neural Network

A Convolutional Neural Network (CNN) \cite{169, 235} is a type of ANN typically used for analyzing image data. It aims to discover the ideal set of filters (weights) that can detect the required features for a given task (e.g., image classification). A typical CNN architectural block consists of three distinct types of layers. These types consist of convolutional layers, pooling layers, and fully connected layers. By stacking these layers, a CNN architecture is produced. Next, we’ll introduce them individually.

- **Convolutional Layer**: The convolutional layer is the fundamental component of a CNN. It requires three components: input data, a filter kernel, and a feature map. If the input is an image, the feature filter kernel can be described as a two-dimensional array of weights, which is applied to the image and shifted by a stride until the filter has swept across the entire image. For each filter operation, a dot product is calculated between the input data and the filter and the result of the dot product is cached into an output. The mapping from the original input or feature filter to the final output is called a feature map. As shown in Figure 7, a kernel size of 3*3 is defined to extract features from an image channel with a size of 5*5, and a feature map with a size of 3*3 is finally formed from this convolutional operation. Note that the first convolutional layer is typically responsible for capturing low-level characteristics such as edges, color, gradient direction, and so forth. With additional layers, the architecture also responds to high-level characteristics.

- **Pooling Layer**: The pooling layer reduces the number of input parameters to achieve dimension reduction. In a manner comparable to the convolutional layer, the pooling procedure is performed by sweeping a weightless filter across the whole input. The pooling kernel aggregates the data in the receptive field to populate the output array. Max pooling and average pooling are the two main aggregation functions. Max pooling chooses the highest value for transmission to the output, and average pooling computes the value that represents the receptive field’s average intensity and then transmits that value.

- **Fully Connected Layer**: Each node in the output layer is directly connected to a node in the preceding layer in the fully connected layer. When this layer occurs as the final layer of the network, it performs classification or regression based on the features extracted through convolutional and pooling layers.
Fig. 7. An illustration of convolutional neural networks.

3.5 Graph Neural Networks

GNNs [268] are a class of neural networks that operate on graph-structured data. They often employ a message-passing mechanism in which the representation of a node is derived from the representations of its neighbors via a recursive computation. So, eventually, the node representation can encode high-order structural information via several aggregation layers. Currently, GNNs have demonstrated promising performance for graph-structured data [119, 204, 365].

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph, where $\mathcal{V}$ is the set of $N$ nodes $\{v_1, v_2, \ldots, v_N\}$ and $\mathcal{E}$ is the set of edges. The node feature matrix can be denoted as $X \in \mathbb{R}^{N \times K}$, where $K$ is the number of node features. The graph structure can be described by adjacency matrix $A \in \mathbb{R}^{N \times N}$, where $A_{i,j}$ denotes the relationship between nodes $v_i$ and $v_j$. Thus, a graph can also be denoted as $\mathcal{G} = (A, X)$.

As shown in Figure 8 from [204], graph filtering including feature transformation and feature aggregation would be performed on node features $X$ and graph structure $A$. After graph filtering, graph structure $A$ would be kept the same while original node features $X$ would be updated as $X_f$. Here, we take a graph convolution network (GCN) [152] as an example to elaborate more details on the graph filtering operation. In Equation (6), $Z$ is the new graph embedding for each node learned from a two-layer GCN:

$$ Z = \sigma(\tilde{A}\sigma(\tilde{A}XW_1)W_2), $$

where $\tilde{A} = D^{-1/2}AD^{-1/2}$ is the normalized adjacency matrix, and $D$ is the degree matrix. $\sigma$ is the activation function such as ReLU. $W_1$ and $W_2$ are the learned weight matrices in the first and second GCN layer training.
4 SINGLE-CELL ANALYSIS PIPELINE

Single-cell data is used in a wide range of applications and requires a number of different stages of data processing to prepare the data and develop biological analysis. The pipeline consists of tasks that can be grouped by overall goal; we illustrate a typical single-cell analysis pipeline in Figure 9. The first collection of tasks consists of applying pre-processing functions that improve the data quality and normalize the information. Then a number of transformations of the data are performed to generate a better representation that eliminates possible instrumentation noise from the biologically salient signal. Given these commonly performed tasks, the user then chooses select analysis, based on the application, to capture high-level biological features, e.g., discovering the types of cells in the population. The final stage takes the output data for the chosen application that pursues scientific insights in fields like oncology, immunology, and so forth.

The data type of many single-cell technologies consists of sparse matrices with rows and columns indicating cells and features (usually counts of genes expressed), respectively. For technologies that mark cells via barcodes, errors leading to empty or doubly marked cells require quality control to remove them and other low-information data from the dataset [159]. Cells with a very low percentage of non-zero features may be removed, as well as cells with more individually expressing features but a small total count of features [133, 230]. Further data normalization is also necessary for mitigating instrumentation variation between cell samples [125]. When samples are measured in different batches, large biases can be introduced between the groups, leading to a large body of techniques dedicated to removing these “batch effects” [110]. A careful balance needs to be struck between applying enough normalization to remove technical artifacts but not so much that natural biological variation is also lost. While a standard collection of transforms has been established in the field, recent research suggests that accepted normalization techniques should be carefully chosen based on the dataset; different depth normalization and variance stabilization methods can have large effects on downstream interpretation (see [21, 54]).

After initial pre-processing is completed, the data is further transformed and simplified to better capture biological information and remove instrumentation biases. Sparsity in the data is near ubiquitous in data produced from single-cell technologies. The main method for addressing this gap is via imputation, a principled means of replacing zeros in data with non-zero values. When data is sampled in batches, large biases can be introduced that need to be addressed via batch effect removal. Similarly, multi-modal technologies can produce multiple types of data from the same cell, which requires a means to combine the data types into a single representation. Lastly, in order to
reduce the number of parameters and to better capture information and remove noise, dimension reduction techniques are employed to obtain a low-dimensional representation of the input data. All these processes aim to obtain a better representation of data for chosen downstream tasks, and many modern methods perform multiple of these data transformations simultaneously. Machine learning becomes especially important in these transformations, and deep learning in particular has shown remarkable success. In the following sections, we will survey two representative data transformation tasks, i.e., multi-modal data integration in Section 8 and imputation in Section 6.

Given the intermediate representation obtained by the above transformations, the pipeline begins to branch depending on the type of data and the end goals for scientific inquiry. Generally, these methods involve some form of data annotations to inform the biological properties of the cells. The most common form of this is clustering and cell annotation, which involve partitioning cells by their cell types or gene expression behavior and other biological markers. Cell annotation (Section 7) specifically uses known cell markers and other prior knowledge to classify the sampled cells [272], whereas single-cell RNA-seq clustering (Section 5) is unsupervised and only uses the given data to group cells. The type of clustering that is performed can depend on the type of technology used to sample the cells. For image-based single-cell profiling, the clustering of the pixels of the image into distinct cells is known as the cell segmentation problem. As well, the newly developed spatial transcriptomic single-cell sequencing presents unique challenges that require alterations to the clustering methods applied to scRNA-seq data. The spatial domain task partitions the sample into geometric clusters dependent on the samples’ spatial positions and according to their gene expression behavior. Furthermore, the spatial samples mix multiple cells together in spots, which requires the identification of the cell types in each spot; this task is known as cell deconvolution. Since all of these tasks are well designed for the application of deep learning, we review them in Section 9 for the spatial domain problem, Section 11 for cell segmentation, and Section 10 for cell deconvolution.

Beyond clustering and annotation, if cells are sampled multiple times throughout stages of cell growth and differentiation, the gene expression between similar cells indicates cell developmental trajectories. Connecting cells over time as being part of a cell trajectory is a central part of understanding stem cell behavior [316] and cancer [84]. Cell-cell communication (CCC) and interactions are additional biological signals that can be captured that indicates information about tissues, cell processes, and intercellular relationships [5]. These relations given between cells determined in trajectory analysis and CCC give a means to understand how “close” cells are to one another. In contrast to spatial transcriptomics, scRNA-seq does not impart spatial information, so the above computations can be used in place of underlying geometric information.

The end goal of any single-cell analysis is to solve problems and address hypotheses developed in highly important fields of biology. Already there have been major accomplishments made in oncology, immunology, differential expression analysis, and the creation of cell type atlases. In oncology, single-cell sequencing has allowed for the identification of rare cells in tumors and how they contribute to tumor progression [51]. As well, single-cell analysis has the potential to uncover causes of metastasis and therapy resistance in cancer cells, which could lead to novel treatments and therapies [231]. Single-cell data has also been a boon for immunology, which studies the biology of immunity and can provide insights into diseases and diagnostics. Previous bulk cell measurements were only able to take average responses from populations of heterogeneous cells, which obfuscates the behavior of important rare immune cells [36]. Differential expression analysis is fundamental to identifying genes causally linked to phenotype states and changes, and the analysis of gene expression behavior on the level of individual cells allows for novel discoveries [281, 319]. Single-cell data has led to the creation of massive cell atlases that serve as a foundational
tool for studying complex tissues and cancers [315, 325]. These applications, and many others, will continue to see novel research gains as single-cell analysis techniques continue to advance.

5 CELL TYPE ANNOTATION

The information with cellular level granularity scRNA-seq data allows the identification of cell identity, which can shed light on sample heterogeneity and biological roles of different cell populations in different organs and biological statuses. Thus, one of the main tasks in scRNA-seq data analysis is cell type annotation. Traditionally, single cells are first clustered and then manually annotated by experts based on genetic profiles [31, 88, 113, 307]. However, such a method faces several drawbacks. Manual annotation results are susceptible to the choice of clustering method and parameters, experts’ subjectivity, and the use non-standardized cell type ontologies, which can make them non-comparable across datasets and experiments. Furthermore, this method is also time-consuming and laborious, which makes it less scalable to large datasets [3, 129, 240]. In response to these challenges, automatic cell type annotation methods have been proposed in recent years. Currently, most cell type annotation algorithms use either marker genes or reference datasets with well-curated cell type labels. As cell type annotation can be treated as a classification task, where each cell is assigned with a label, the performance of these methods can be evaluated using datasets with known cell type labels. We summarize representative tools or baseline methods for the cell type annotation task in Table 2, and five curated cell atlases with annotated cell types to train and benchmark cell type annotation algorithms in Table 3.

5.1 Traditional Methods

Many methods have been developed to identify single-cell identities that rely on traditional statistical or machine learning methods. To date, most methods require clustering first and then assign a cell type identity to each cluster. They use either marker genes or reference datasets. scCATCH [273], CellAssign [362], SCINA [366], SCSA [34], scSorter [106], cellMeSH [210], and scType [132] are based on prior knowledge about cell-type-specific marker genes. scCATCH, SCSA, and cellMeSH use a well-curated cell marker database. scCATCH and SCSA devise scoring models to assign cell types to cell clusters, while cellMeSH applies a probabilistic model to a database weighted by the strength of association. However, these methods assign all clusters to some known...
Table 2. A Summary of Clustering Analysis Tools

| Tools       | Algorithm | Description                                                                 | Language |
|-------------|-----------|-----------------------------------------------------------------------------|----------|
| DANCE       |           | A correlation-based method                                                   |          |
| CellAssign  [362] | Classical | A probabilistic model leveraging gene marker information                     | R        |
| Garnett     [247] | Classical | An elastic-net-regression-based classifier that uses selected representative cells for each cell type | R        |
| SingleR     [10]  | Classical | A correlation-based method                                                   | R        |
| CHEATAH     [63]  | Classical | A correlation-based method that hierarchically assign cell types              | R        |
| SingleCellNet [301] | Classical | A random forest [25] classifier                                              | R        |
| scMatch     [124] | Classical | A correlation-based method                                                   | Python   |
| SCINA       [366]  | Classical | An expectation-maximization algorithm assuming a bimodal distribution for each signature genes | Python   |
| CIPR        [77]   | Classical | An algorithm that assigns cell types based on similarity with reference samples using marker genes information | R        |
| SciBet      [172]  | Classical | A maximum likelihood estimation method that utilizes marker gene information under assumed multinomial distribution for gene expressions | R        |
| SCSA        [34]   | Classical | A scoring method that leverages known marker genes and their confidence levels | Python   |
| CellMeSH    [210]  | Classical | A probabilistic model using the constructed CellMeSH database               | Python   |
| scSorter    [106]  | Classical | A semi-supervised method that clusters and assigns cell types based on marker gene information | R        |
| scType      [132]  | Classical | A scoring method for cell type assignment based on marker genes              | R        |
| CellTypist  [95]   | Classical | An ensemble regression model with stochastic gradient descent learning      | Python   |
| SuperCT     [344]  | MLP       | A fully connected MLP model using binarized expression data from MCA with transfer learning to incorporate new cell types | Python   |
| ACTINN      [359]  | MLP       | An MLP consisting of three hidden layers                                     | Python   |
| EnClaSC     [47]   | MLP       | Combination of a few-sample learning strategy to assign query cells and an MLP model for cells unassigned to rare cell types | Python   |
| scNym       [149]  | MLP       | A semi-supervised method that utilizes adversarial neural network           | Python   |
| scIAE       [356]  | AE        | An ensemble framework integrating stacked denoising and sparse autoencoders. | Python   |
| scDeepSort  [274]  | GNN       | Adaptation of GraphSAGE [111] to address a weighted cell-gene graph to produce node representations to pass a linear classifier | Python   |
| sigGCN      [330]  | GCN       | Integration of a GCN and an MLP whose concatenated learned features pass through the classification layer | Python   |

Table 3. A Summary of Cell Type Annotation Datasets

| Cell Atlas           | Species  | Atlas Description |
|----------------------|----------|-------------------|
| Human Cell Atlas     [257] | Human    | 33 organs 30 million cells |
| Tabula Sapien        [308] | Human    | 24 organs 500k cells |
| Human Cell Landscape [114] | Human    | 56 tissues 700k cells |
| Tabula Muris         [307] | Mouse    | 20 organs 100k cells |
| Mouse Cell Atlas     [113] | Mouse    | 51 tissues 400k cells |

cell labels and thus could not identify unknown cell types not present in the database, while CellAssign, SCINA, scSorter, and scType support unassigned clusters. scType also devises a scoring metric that is applied to its cell type marker database. CellAssign uses a Bayesian framework where a negative binomial is fitted to the marker gene expression profiles given a cell type. SCINA assumes each marker gene follows a bimodal distribution, where the high mode represents the predicted cell type. scSorter first clusters query cells into known cell types and then separates cells with unknown cell types using information from both marker genes and non-marker genes.
Alternatively, many methods utilize reference datasets instead. Among these methods, scmap-cluster \[154\], SingleR \[10\], CHETAH \[63\], scMatch \[124\], and CIPR \[77\] assign cell types to query clusters based on measures of correlations with reference datasets, such as Pearson correlation, Spearman correlation, or cosine similarity. scmap \[154\] also supports cell type assignment to individual cells. By comparison, a number of methods use supervised models. For example, Garnett \[247\] builds a hierarchical tree of cell types based on marker genes and then uses elastic net regression to assign cell types to clusters. SingleCellNet uses the top-scoring pair algorithm \[279\] to select informative genes, transforms data into a binary form, and train a multi-class random forest \[25\] classifier using reference datasets. SciBet \[172\] first selects gene features based on statistic differential entropy, fits a multinomial model for gene expressions in each cell type, and then uses a maximum likelihood approach to assign cell types. CellTypist \[67\] uses a logistic regression model with stochastic gradient descent learning for cell type identification using a curated pan-tissue database for immune cell types. By comparison, scPred \[6\] performs cell type identification for individual cells by first selecting informative PCs from singular value decomposition and training a support vector machine (SVM) classifier. A special case is scType \[132\], which is unsupervised since it uses neither marker genes nor reference datasets.

5.2 Deep Learning Methods

Deep learning methods have been applied to address the cell type annotation task in recent years. ACTINN \[199\] trains a neural network with three hidden layers using reference datasets to predict cell type for each cell. Similarly, SuperCT \[344\] trains an artificial neural network model with two hidden layers each followed by a dropout layer against over-fitting using a well-curated database including the Mouse-Cell-Atlas. Its inputs are binarized gene expressions; i.e., an entry in the transformed expression matrix is assigned a value of 1 if the corresponding cell expresses the gene and 0 otherwise, while each neuron in the output layer represents a candidate cell type. SuperCT provides three models: v1m and v2m for mice, and v1h for human. New cell types can be incorporated by superCT through transfer learning, where the weights for the first hidden layer are frozen.

EnClaSC \[47\] draws upon ensemble learning. Query cells are first screened by a scoring strategy designed to identify rare cell types. Training samples of rare cell types are paired one by one, and their gene expressions are concatenated to form a paired sample. If the two training cells in a pair are of the same rare cell type, the paired sample is labeled with 1, or 0 otherwise. In the test phase, each query cell is paired with each training cell in a similar fashion. A tree-based model, LightBGM \[146\], is used to predict whether a query cell is of the same rare cell type of a paired training cell. The final score for a query cell and a rare cell type is normalized prediction results for the given query cell and all training cells belonging to the said rare cell type class weighted by Pearson correlations of their gene expressions. Cells unassigned by this scoring strategy are examined by an ANN model consisting of four dense layers and two dropout layers. In this phase, training cells are sampled with replacement 10 times to train 10 neural networks, whose prediction results are voted to obtain a final prediction.

scNym \[149\] is another deep learning method that combines semi-supervised learning and adversarial neural network. In particular, it utilizes information from both reference and testing datasets during the training phase by using MixMatch \[18\] semi-supervision and domain adversarial \[91\] iteratively to accommodate for the differences in data distribution between training and testing datasets due to variability in experiment platforms or conditions. Under the MixMatch framework, query cells are pseudo labeled \[131, 322\] using the current classifier such that the pseudo label’s cross-entropy is minimized. The query cells are then randomly paired with
training cells to generate pair-wise weighted averages to train the scNym model. The use of a convex pair-wise combination of training and testing samples assumes that the linear interpolation of feature vectors leads to that of associated targets [363], which encourages generalization between the training data and testing data. Given the embeddings learned by the scNym model, the domain adversarial framework consisting of a two-layer neural network is then used to distinguish whether the example comes from the training or the testing data to compete with the classifier. The model parameters are optimized based on objectives combining classification, MixMatch interpolation consistency, and the domain adversarial.

scIAE [356] also uses an ensemble classification framework where the final prediction is based upon multiple base models, each consisting of two stacked sparse denoising autoencoders. The inputs to each base model are gene-wise randomly subsetted expression matrices of both the training and the testing data. The subsetted training data is used to train a stacked sparse denoising autoencoder to project the subsetted testing data to lower-dimensional embeddings. The classifier obtained from the training data is then applied to the testing data’s lower-dimensional representation for prediction. The prediction results generated by all base models are then integrated for final prediction through voting.

In addition, scDeepSort [274] and sigGCN [330] are two methods that draw upon graphs. sigGCN employs a GCN to reflect the non-linear topological relationship among cells. It first constructs a gene-wise weighted adjacency matrix using the STRING database [299] to create a gene interaction network where node features in this network are defined as corresponding gene expressions. This graph is used as the input to a GCN-based autoencoder consisting of a convolutional layer and a maxpooling layer followed by a flattened layer and a fully connected (FC) layer. Another FC layer is then used as the decoder to reconstruct gene expressions from the obtained embeddings, which is associated with gene expression reconstruction loss measured by mean squared errors. Meanwhile, a two-layer neural network is trained using the training gene expression matrix in addition to the GCN-based autoencoder so that the hidden layers produced by the two models are concatenated. Finally, the concatenated features are then used to train the classification layer whose loss is given by the negative log-likelihood corresponding to the true cell type label. Furthermore, in addition to the gene expression reconstruction loss and the classification loss, a regularization loss is incorporated to prevent overfitting. The three types of loss are combined to train the sigGCN model to obtain optimized parameters.

By comparison, scDeepSort [274] uses a graph neural network using a weighted bipartite graph where both cells and genes are its nodes. The gene expression value for each cell–gene pair is the edge weights between them. Features for gene nodes are obtained from principal component analysis, while those for cell nodes are the weighted aggregation of gene node features they connect to. The gene–cell graph then goes through a weighted graph aggregator layer adapted from GraphSage [111], which transforms a node along with information taken from its local neighborhood. In particular, the aggregation layer in scdeepSort incorporated the weighted adjacency matrix normalization to address gene expression variability and the learnable sharing confidence to deal with batch effect and dropouts in scRNA-seq data. Each edge will have a corresponding learnable sharing confidence parameter weighing the connection of the neighbors, in addition to another parameter for self-loops for cells. The learned confidence parameters inform which genes are discriminatory for annotation. Finally, cell node representations in the latent representations produced by the aggregation layer pass a linear classifier to predict cell labels. When the trained model is applied to testing datasets, query cells are connected to gene nodes of the trained cell–gene graph with the corresponding gene expression as the edge weight between the new cell node and gene node.
6 IMPUTATION

The increased resolution of single-cell RNA sequencing methods comes at a cost in the form of the increased sparsity of the data. The sequencing technology may fail to capture a number of the expressed genes of an individual cell due to low RNA capture rate, as well as the stochasticity of mRNA expression. This artificially large proportion of false zero counts is defined as “dropout” [148, 296]. The zero-inflated data can have sparsity rates from 40% to anywhere as high as 85% to 90% [251]. This requires several preprocessing steps like gene selection and dimension reduction to improve the data signal and make downstream tasks more viable. Single-cell DNA methylation sequencing technologies like scBS-seq and scRRBS-seq have limited amounts of DNA starting material per cell, so methylation sequencing can be limited by the degree of CpG coverage. This leads to missing data across the genome in each cell, requiring imputation for further analysis. A direct way to address dropout in both cases is to perform imputation, a method to correct false zeros by estimating realistic values for those gene–cell pairs. For scRNA-seq data, imputation creates artificial count values for genes that fail to express, while for DNA methylation the imputation will only provide the binary one or zero. However, imputation methods must be careful to distinguish the zeros in the data that represent the true absence of gene expression or methylation in specific cells from the zero counts due to dropout. Otherwise, information will be lost during analysis from removing the truly expressed zeros. We summarize all tools or baseline methods for imputation in Table 4 and useful benchmarks in Table 5.

Imputation methods can be trained and evaluated in a couple of different ways. Training an imputation method frequently occurs as an unsupervised learning problem for a single dataset. If simulated datasets are used, the underlying true counts are known, so reconstruction error (usually mean squared error) can be used for evaluation. On empirical datasets, there is no knowledge of the true counts, so other methods have to be used. One choice is to induce artificial dropout on the non-zero values so that reconstruction error can be computed on those values as an evaluation metric. Alternatively, some metrics may be used to determine how well the imputation process preserves or enhances certain biological features of the data. For example, if cell types or sampling times of the cells are known in a dataset, then some clustering metrics may be used to evaluate the success of the imputation, including adjusted rand index (ARI), cluster accuracy, and silhouette score. Similarly, single-cell datasets may have genes that are known to differentially express (DEGs) if bulk RNA sequencing is done alongside scRNA sequencing [281]. Such differential expression can be evaluated by how well DEGs are detected after imputation, using AUC or F-scores.

6.1 Traditional Methods

A large number of methods exist for the task of scRNA-seq data imputation, mainly focusing on generative probability models or matrix factorization. One of the early methods is MAGIC, which uses Markov diffusion networks to replace zeros in gene expression count data. MAGIC [320] calculates distances between cell expression vectors, converts the distances using the Gaussian kernel, and then uses them as a weighted adjacency matrix to perform Markov diffusion. This method is shown to have shortcomings in that it replaces all zeros despite some being true zeros and replaces the non-zero values with new potentially biased values. Many other methods also use cell–cell graphs to capture information of similar cells to determine how genes with dropout behave in similar cells that have nonzero expression. netSmooth [260] imputes using a diffusion process similar to MAGIC. DrImpute [103] takes clusters of cells and simply takes averages over the clusters to find imputation values.

Another popular imputation technique uses matrix factorization techniques. VIPER [42] develops a non-negative sparse regression model to learn the underlying count data. scRMD [38]
### Table 4. A Summary of Popular Imputation Tools

| Tools         | Algorithm | Description                                                                 | Language | Reference |
|---------------|-----------|-----------------------------------------------------------------------------|----------|-----------|
| DANCE         |           | MAGIC uses a diffusion model to smooth values and impute zeros.              | R        | [66]      |
| DrImpute [103]| Classical | D Impute clusters the data and averages the values of the clusters.       | R        |           |
| scImpute [175]| Classical | scImpute uses a statistical model to identify dropouts, and non-negative least squares regression to impute. | R        |           |
| CEDR [183]    | Classical | CEDR uses a statistical model to identify dropouts, and weighted means in clusters are used for imputation values. | R        |           |
| ALRA [188]    | Classical | ALRA uses Singular Value Decomposition to get low-rank representation of data, then attempts to restore biological zeros using thresholding. | R        |           |
| CMFImpute [347]| Classical | CMF-Impute performs a matrix optimization to find a regularized SVD for the scRNA-seq expression data. | Matlab   |           |
| scRMD [38]    | Classical | scRMD finds a robust regularized matrix decomposition using nuclear and ℓ₁ norm regularization. | R        |           |
| MCImpute [224]| Classical | MCImpute performs a nuclear norm minimization optimization for regularized matrix completion. | Matlab   |           |
| netSmooth [260]| Classical | netSmooth performs network diffusion on the expression data with priors for the covariance structure. | R        |           |
| SAVER [?]     | Classical | SAVER is a probability model that generates counts using negative binomial distribution. | R        |           |
| LATE [13]     | AE        | LATE is a undercomplete autoencoder trained on reconstructing the non-zero inputs, while TRANSLATE combines this autoencoder with transfer learning by using a reference dataset to initialize parameters. | Python   |           |
| AutoImpute [300]| AE        | AutoImpute uses an overcomplete autoencoder trained on non-zero entries for the imputation of dropout locations. | Python   |           |
| scScope [65]  | AE        | scScope uses an iterative autoencoder that cycles output into input while applying batch effect correction. | Python   |           |
| DCA [81]      | AE        | DCA is an undercomplete autoencoder that predicts parameters of chosen distributions like ZINB to generate the imputed data. | Python   |           |
| scGNN [328]   | GNN       | scGNN uses three autoencoders in a cycle using a graph autoencoder, a plain autoencoder, and a cluster-specific collection of autoencoders. The convergence of a cell graph is used in the regularization of a final autoencoder that performs the imputation. | Python ✓ |           |
| GraphSCI [253]| GNN       | GraphSCI uses two autoencoders, one being a graph autoencoder on a gene graph, and the other reconstructs the input using the graph as additional input. | Python ✓ |           |
| scGANs [349]  | GAN       | scGANs use a GAN to model the generation of scRNA-seq data using the generated expression data for the imputation. | Python   |           |

### Table 5. A Summary of Imputation Datasets

| Dataset         | Species | Tissue | Dataset Dimensions | Protocol                  |
|-----------------|---------|--------|--------------------|--------------------------|
| Tabula Muris    | Mouse   | 20 tissues | 53,760 cells      | SMART-Seq2 10x Genomics  |
| Tabula Sapiens  | Human   | 24 tissues | 483,152 cells     | Illumina NovaSeq 6000    |
| 10X PMBC 4K     | Human   | PBMC   | 4,271 cells 16,653 genes | 10x Genomics     |
| Human Cell Atlas| Human   | Various tissues | 30.6 million cells | HiSeq X Ten         |

Leverages regularized robust matrix optimization, netNMF-sc [79] adopts non-negative matrix factorization with network regularizations, and MCImpute [224] builds low-rank matrix factorization. CMF-Impute [347] uses matrix optimization via singular value decomposition (SVD), and ALRA [188] applies SVD and thresholding.
SAVER is a generative method that attempts to model the probability distribution of the gene expression count data. The negative binomial (NB) distribution is used to model the data but is treated equivalently as a Poisson distribution with a Gamma prior on the Poisson parameter. Using the data, an empirical Bayes framework is used to estimate the Gamma prior via Poisson Lasso regression. Then the posterior distribution is used to output the expression as the posterior mean with uncertainty quantification given by the distribution. bayNorm similarly utilizes a Bayesian model to generate imputed values. scImpute combines a few techniques by creating neighbor relations between cells leading to clusters, before using a probabilistic model to determine which zeros are dropout events, and then using non-negative least squares (NNLS) regression to impute the values. CIDR likewise employs a generative model to give a probability of dropout for zero values, which is determined via non-linear least-squares regression, before using the probabilities to calculate a weighted mean for the imputation value.

While the preponderance of imputation methods is applied to scRNA-seq data, a couple of methods have been developed for single-cell DNA methylation data. Melissa imputes values based on Bayesian estimation, while CaMelia makes use of the machine learning technique of boosting.

6.2 Deep Learning Methods

A natural deep learning architecture for the task of imputation is the autoencoder, due to its prevalence in data denoising and missing data applications. One of the early autoencoder methods for scRNA imputation is AutoImpute. It is unlike many denoising autoencoders since it employs an overcomplete AE model rather than the usual undercomplete one. The AutoImpute model is a simple two-layer AE, one layer each for the decoder and encoder. The hidden layer of the encoder output is designed to be larger than the input dimension size dependent on the training dataset. Only the encoder has an activation function, and the decoder is entirely linear. The objective function is a commonly used $L^2$ reconstruction loss penalty function, with the addition of $L^2$-norm regularization terms on the weights of the encoder and decoder.

The deep count autoencoder (DCA) is another autoencoder-based imputation method. In contrast to the overcomplete model of AutoImpute, DCA has the standard undercomplete architecture. DCA consists of four layers, two for the decoder and two for the encoder, and takes a form similar to a standard VAE. Instead of modeling a distribution and attempting reconstruction of the input, DCA directly predicts the parameters of a chosen probability distribution of the input data. By choosing a probability distribution like the zero-inflated negative binomial (ZINB) as the generating data distribution, the DCA outputs the three parameters of the distribution and employs a log-likelihood loss function to train the model. This triple output requires three separate layers transforming the output of the decoder. Other potential probability distributions include the NB, which may better represent the count data of scRNA-seq even with dropout, as well as the Poisson distribution. Additional regularization can be placed on the loss function in the case of ZINB, penalizing the $L^2$ weight of the zero-inflation parameter.

scScope is a combination autoencoder and recurrent neural network that performs imputation along with batch effect removal. Like some of the above methods, scScope operates in an iterative manner, taking the output of the model and running it through the model sequentially until some stopping condition is met. However, a recurrent element combines original input with sequential output. The input has an initial batch effect correction for input data $x \in \mathbb{R}^d$ using learnable parameter $B \in \mathbb{R}^{d \times K}$ of the form

$$x_c = \text{ReLU}(x - Bu),$$
where $K$ is the number of batches, and $u$ is a one-hot vector indicating the batch of $x$. The batch-corrected vector is then input to the AE, which has a single-layer encoder and a single-layer decoder. The final layer performs a fully connected transformation with ReLU as the activation function and then makes all entries that are not originally zero equal to zero, giving a vector of imputed values on all the potential dropout entries. This vector is added to $x_c$ and then put through the AE model again. The objective function is the standard $L_2$-error reconstruction loss on all the non-zero entries of $x_c$. SAVER-X, an adaptation of the SAVER model, combines an autoencoder with a Bayesian hierarchical model, while also using transfer learning. The autoencoder is pretrained on a reference dataset to leverage the information in some potentially larger reference datasets before being further trained on the target dataset.

scGNN [328] utilizes a graph autoencoder trained in an iterative loop with other standard autoencoders. The initial training of scGNN feeds the data through three models that give their output to the next model for input, and this continues until convergence. The first model is a standard undercomplete autoencoder with two layers for the encoder and decoder and is trained using reconstruction loss with an additional penalty term that captures regulatory signals in the gene expression data. A $k$-nearest neighbor (KNN) cell graph is created based on the embedding learned by the feature autoencoder and pruned of outliers using the Isolation Forest algorithm. Then a graph autoencoder takes the output graph and the reconstructed output of the first autoencoder and runs both through a graph autoencoder that has two layers using graph convolutions and a decoder that attempts to reconstruct the graph from the embedding using a sigmoid activation function on the gram matrix of the embedding. The loss function of the graph autoencoder is cross-entropy between the original graph and the reconstructed graph. After output from graph embedding is given, a $k$-means clustering method is used to cluster the cells embedding, with the number of clusters determined by the Louvain algorithm. Then the final model has one autoencoder trained for each cell cluster, using the same architecture and loss function as the first autoencoder except with no regularization added. After the iterative process stops, a final autoencoder is trained for imputation. It uses the same architecture as before but also adopts three additional regularizers in its training objective function derived from the cell graph created in the iterative process.

Another method that uses a graphical convolution network with an autoencoder is GraphSCI [253]. This method constructs a graph from the data with genes taken as the nodes and the edges between them given by the Pearson correlation coefficient (PCC) of the (log-normalized) expression data, so that if the correlation of two genes is $>.3$ or $<-.3$, an edge is established between them. Given this graph, the autoencoder and graph autoencoder make use of the expression data and graph data to reconstruct its input. The graph autoencoder applies a two-layer GCN to the gene expression matrix using the adjacency matrix of the graph and uses the encoding as the means and variances to Gaussian distributions. The distributions are used to generate a reconstruction of the adjacency matrix, where each edge of the graph follows a Gaussian distribution parameterized by a pair of mean and variance parameters given in the encoding. A reparameterization method forgoes directly sampling from these distributions and instead computes the reconstruction of the adjacency matrix as the sigmoid function applied to the inner product of latent vectors, one computed for each gene. The autoencoder is designed similarly, taking the gene expression matrix as input and applying two layers to yield an encoding that gives the parameters of a ZINB distribution. The first layer takes the adjacency matrix into account by taking the element-wise product with the weight matrix. The reparameterization computes the reconstruction of the expression matrix as a collection of latent vectors times a diagonal size matrix that is given by the size factors of the input cells. The training of the model is iterative, using the reconstruction of the gene expression matrix and the adjacency matrix in each iteration. The loss function uses the variational lower
bound on both autoencoders, using the KL divergence on the encoded variables for the ZINB and Gaussian distributions.

For the application of DNA methylation imputation, DeepCpGm [9] is a method that takes both DNA sequences and methylation states as inputs and outputs imputed methylation states. The DNA sequence is input into a CNN via a one-hot encoding of base pairs in the DNA sequence, so the four base pairs would be encoded by vectors \((1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0),\) and \((0, 0, 0, 1)\). Each CpG site with a sequence of 1,001 base pairs centered on the site would be input into the DNA CNN module. The CNN consists of two pairs of a convolution layer followed by a pooling layer, before going into a fully connected layer. The methylation state sequence over the CpG sites is first transformed before being input into a bidirectional gated recurrent network (GRU). The states are given by \(T\) vectors representing the methylation states around a target CpG site for \(T\) different cells. A fully connected layer transforms the vectors into a representation that seeks to capture interactions between methylation states and distances within cells, before the sequence of vectors is fed into the GRU, which seeks to model dependencies between cells. The GRU is bidirectional to analyze the dependencies between cells independent of the order of the cells given in the data. The forward and backward GRUs produce \(T\) hidden state vectors, with the last hidden states of the GRUs concatenated and given as output. Finally, the output of the CNN and GRU modules are concatenated and transformed by a two fully connected layer NN, with the final layer output using the sigmoid activation function to produce a dimension \(T\) vector giving the predictions of the methylation rate in \([0,1]\) of a chosen CpG site for \(T\) cells. The training uses a negative log-likelihood loss function with a \(L_2\) weight-regularized penalty function.

7 CLUSTERING

One critical phase of single-cell analysis is to characterize cell types within a given tissue. Cell type groups are typically made up of cells with similar transcriptome files. Knowing specific cellular types will help us to reveal the diversity between different groups and predict gene expression of incoming cells [90]. Single-cell sequencing technology offers gene expression at the single-cell level. It allows researchers to elucidate the transcriptome heterogeneity between individual cells, which unveils the underlying subgroup structure [357]. As an unsupervised learning method, clustering has been proven to be the most effective methodology for identifying cell subgroups utilizing transcriptomic data. Classical clustering methods like K-means, Leiden, and Louvain are widely applied in single-cell analysis, while the latter two graph-based methods become the dominant choice when utilized as an additional clustering module. ScDeepCluster [311] summarized some popular benchmark datasets like 10X PBMC 4K [369], Mouse Bladder Cells [112], and Worm Neuron Cells [30]. Based on those benchmarks, there are popular metrics to evaluate clustering performance: the adjusted Rand index (ARI), the accuracy (ACC), and the normalized mutual information (NMI). A high metric score indicates that the model successfully identifies the underlying subgroup structure and satisfyingly recovers the cell type group assignment. We summarize some popular clustering analysis tools in Table 6 and available datasets in Table 7.

7.1 Traditional Methods

Hierarchical clustering [140] has been widely adopted by early works. BISCUIT [250] incorporates parameters denoting technical variation into a hierarchical Dirichlet process mixture model. SINCERA [108] presents a generally applicable analytic pipeline for processing single-cell data with hierarchical clustering. CIDR [184] performs hierarchical clustering on the first few principal coordinates of the dissimilarity matrix between the imputed gene expression. Corr [138] calculates cell-pair differentiability correlation as the input of the hierarchical clustering module. Corr
Table 6. A Summary of Clustering Analysis Tools

| Tools   | Algorithm | Description                                    | Language | Language  |
|---------|-----------|------------------------------------------------|----------|----------|
| SC3     | Classical | Consensus clustering of single-cell RNA-seq data | R        | DANCE [66] |
| Seurat  | Classical | Integrated analysis of multimodal single-cell data | R        |          |
| scedar  | Classical | Scalable Python package for single-cell RNA-seq data | Python   |          |
| SIMLR   | Classical | Multi-kernel package for large-scale genomic analysis | R        | Matlab   |
| SPRING  | Classical | Interface for visualizing high-dimensional expression | Python   | Matlab   |
| ASAP    | Classical | Platform for analysis and visualization of single-cell RNA-seq data | R        | Python   |
| RaceID  | Classical | Branching point detection in single-cell data by K-branches clustering | R        |          |
| CIDR    | Classical | Fast clustering through imputation for single-cell RNA-seq data | R        |          |
| SOUP    | Classical | Semi-soft clustering of single-cell data | R        |          |
| scRNA   | Classical | Clustering by transfer learning on single-cell RNA-seq data | Python   |          |
| scDesk  | AE        | Deep soft K-means clustering with self-training for single-cell RNA sequence data | Python   |          |
| scVI    | AE        | Deep generative modeling for single-cell transcriptomics | Python   |          |
| DESC    | AE        | Clustering and batch effect removal by autoencoder | Python   |          |
| scDHA   | AE        | Hierarchical autoencoder for single-cell data analysis | Python   |          |
| scDeepCluster | AE | Soft clustering by autoencoder on single-cell RNA-seq data | Python   | ✓        |
| scDCC   | AE        | Autoencoder-based deep embedding for constrained clustering analysis of single cell RNA-seq data | Python   | ✓        |
| DescSCC | GNN       | Graph convolutional network for clustering single-cell RNA-seq data | Python   | ✓        |
| scGNN   | GNN       | Graph neural network framework for imputation and clustering | Python   | ✓        |
| graph-sc| GNN       | GNN-based embedding for clustering scRNA-seq data | Python   | ✓        |
| scTAG   | GNN       | ZINB-based graph embedding autoencoder for single-cell RNA-seq interpretations | Python   | ✓        |
| scDSC   | GNN       | Deep structural clustering for single-cell RNA-seq data jointly through autoencoder and graph neural network | Python   | ✓        |

integrates variance analysis with hierarchical clustering by calculating the proportion of between-group variance to decide where to cut the hierarchical dendrogram.

Other classical clustering methods like K-means [207] and shared nearest neighbor (SNN) clustering [137] have been applied to single-cell data to identify cell types. RaceID [48] utilizes the distance matrix as the input of a K-means algorithm for rare cell type identification. Similar to K-means, the K-branches clustering algorithm [48] locally fits half-lines as the representatives of branches in the differentiation trajectory of cells. Based on the SNN clustering, SNN-clique [346] identifies clusters in the SNN graph by finding maximal quasi-cliques.

There are also other clustering methods for single-cell analysis. SOUP [373] identifies group structure with a non-negative membership matrix representing the membership of cells to clusters. SOUP recovers the membership matrix from the top eigenvectors of the similarity matrix, where the number of eigenvectors is determined by the number of clusters. Based on non-negative matrix factorization (NMF), scRNA [219] factorizes the source dataset into a gene-independent target data matrix and a data-size-independent dictionary, derives a new expression matrix from those two parts, and performs clustering on the reconstructed data.

7.2 Deep Learning Methods
In single-cell analysis, traditional clustering methods may lead to suboptimal results, since single-cell data usually contains a large number of zero elements and its high heterogeneity makes things
Table 7. A Summary of Clustering Datasets

| Dataset                        | Species | Tissue     | Dataset Dimensions | Protocol          |
|--------------------------------|---------|------------|--------------------|-------------------|
| Tabula Muris [56]              | Mouse   | 20 tissues | 53,760 cells       | SMART-Seq2        |
|                                |         |            |                    | 10x Genomics      |
| 10X PBMC 4K [369]              | Human   | PBMC       | 4,271 cells 16,653 genes | 10x Genomics      |
| Mouse Bladder Cells [112]      | Mouse   | Bladder    | 2,746 cells 20,670 genes | Microwell-seq     |
| Worm Neuron Cells [30]         | Worm    | Nerve      | 4,186 cells 13,488 genes | sci-RNA-seq       |
| Human Kidney Cells [41]        | Human   | Kidney     | 5,685 cells 33,658 genes | RNA-seq           |
| Mouse Embryonic Stem Cells     | Mouse   | Embryo     | 2,717 cells 24,175 genes | Droplet Barcoding |
| Adam [4]                       | Mouse   | Kidney     | 3,660 cells 23,797 genes | Drop-seq          |
| Muraro [228]                   | Human   | Pancreas   | 2,122 cells 19,046 genes | CEL-seq2          |
| Romanov [259]                  | Mouse   | Hypothalamus | 2,881 cells 21,143 genes | RNA-seq           |

even harder. To overcome this challenge, deep learning methods have been adopted. DEC [343] proposes a new Kullback-Leibler (KL) divergence loss between the Student’s $t$-distribution of embedding space and the auxiliary target distribution. The proposed KL divergence loss makes it possible for models to achieve self-training under an unsupervised setting. DCA [82] introduced a deep count AE network for denoising and imputation. The noise model is based on ZINB distribution, and the mean, dropout, and dispersion of data are estimated by different activation functions respectively. The ZINB model-based denoising autoencoder and the KL divergence loss then became the backbone of many popular clustering methods.

ScDeepCluster [311] adds a clustering layer to DCA. ScDeepCluster performs clustering based on soft assignment of denoised embedding. The model is optimized by minimizing the KL divergence loss of the embedded space and the ZINB loss of the denoising module. Based on scDeepCluster, scDCC [312] integrates prior information into the modeling process by merging pairwise constraints into the training process. A pairwise constraint can have two types: must-link (ML) and cannot-link (CL). As a tradeoff, the ML constraint forces the paired instances to have similar soft labels, while the loss of the CL constraint encourages different soft labels. In another aspect, scziDesk [41] adds a weighted soft K-means clustering algorithm with inflation operation to the DCA structure. The model is optimized by minimizing the ZINB loss, KL divergence loss, and weighted K-means clustering loss. ScVI [192] uses variational inference to approximate the posterior distribution, where the variational lower bound is optimized by stochastic optimization. The conditional distributions are specified by a ZINB model-based VAE, and the cell type clustering is based on the K-means algorithm on the latent space. With a focus on the batch effect, DESC [178] initializes parameters obtained from a ZINB model-based autoencoder and iteratively optimizes the KL divergence loss. This iterative procedure moves each cell to its nearest cluster centroid and simultaneously removes batch effect over iterations. To conduct clustering and time-trajectory inference, scDHA [313] consists of a non-negative kernel autoencoder and a stacked Bayesian
autoencoder. The kernel autoencoder shrinks the non-negative coefficients of the less important features toward zero, and the stacked Bayesian autoencoder is a VAE with two stacked decoders obtained by two different transformations.

GNNs have been applied to single-cell clustering. GraphSCC [360] accounts for structural relations between cells by constructing a cell–cell KNN graph, processes it through a GCN, and optimizes the network by a dual self-supervised module together with a denoising autoencoder network. The target distribution is calculated from the latent space of the autoencoder, and two KL divergence losses are calculated using Student’s \( t \)-distribution of embedded spaces of an autoencoder and GCN, respectively. Inspired by weighted cell–gene graph introduced by scdeepsort [274], graph-sc [52] models the expression data as a gene-to-cell graph by a graph autoencoder network and then clusters embeddings with the K-means or Leiden algorithm. The encoder iterates only over the cell nodes, and the decoder is trained to reconstruct the normalized adjacency matrix. Focusing on the cell–cell KNN graph, scTAG [357] optimizes a ZINB-based topology adaptive graph convolutional autoencoder with clustering KL divergence loss, ZINB reconstruction loss, and graph reconstruction loss. The topology adaptive graph convolutional encoder processes the node features with a polynomial convolution kernel and produces latent embeddings for soft assignment clustering. With a concentration in the cell–cell KNN graph, scDSC [90] consists of a GNN module and a ZINB-based autoencoder and adapts a multi-module mutual supervision strategy to achieve end-to-end training. The mutual supervision consists of GNN KL divergence loss, binary cross-entropy loss between the target distribution and soft label, ZINB reconstruction loss, and expression matrix reconstruction loss.

8 MULTIMODAL INTEGRATION

The scope and resolution of biology research are being revolutionized by contemporary single-cell omics technology. For instance, scRNA-seq [130] has recently developed into a useful technique for studying biological responses and accelerating innovation in medicine. In the meantime, techniques for profiling cellular properties from different omics are continuously being adapted to work at the single-cell level (e.g., scDNA-seq [209], scATAC-seq [166], REAP-seq [244]). More recently, the quick development of single-cell technology has made it possible to measure a variety of omics in a cell concurrently, including gene expression, protein abundance, and chromatin accessibility. For instance, cellular indexing of transcriptomes and epitopes by sequencing (CITE-seq) [290] allows for the joint profiling of mRNA expression and surface protein abundance, while sci-CAR [29], Paired-seq [371], and SNAP-color [45] allow for the simultaneous quantification of mRNA expression and chromatin accessibility. There are many important applications and achievements of multimodal single-cell analysis. For example, the joint analysis of single-cell multi-omics data can discover novel cell populations [115] or regulatory networks [74, 136, 202, 359]. Since most of the methods in the downstream analysis are also commonly used in the unimodal analysis, for simplicity, we concentrate on the core issue with multimodal single-cell analysis, i.e., multimodal integration.

Multimodal data integration is not a single, clearly defined task because of the variety of modalities and biological problems [254]. Existing methods can be roughly classified into two categories. One is the integrated analysis of several unimodal datasets (i.e., datasets from multiple modalities but not aligned), and the other is the analysis of multimodal datasets (i.e., datasets that provide multi-omic features for each single cell). Both methods can offer fresh insights into cellular states, thus benefiting downstream analysis. Next, we summarize the most common settings of existing methods. Formally, the objective of data alignment among unimodal datasets is to find a function that can project multiple sources into a shared latent space. Given \( k \) datasets \( D_1, D_2, \ldots, D_k \), the
goal is to find an embedding $H \in \mathbb{R}^{n \times d}$ for all the data points, where $n = \sum_{i=1}^{k} |D_i|$, and $d$ is a predefined number of dimensions. Some alternative objectives include (1) finding popularity-to-popularity correspondence, which can be summarized as $p$ clusters $C = \{c_1, c_2, \ldots, c_p\}$ that are shared among datasets, so that $D_1 \cup D_2 \cup \ldots \cup D_k \mapsto C$, and (2) finding cell-to-cell correspondence, which can be described as a pair-wise similarity matrix $S \in \mathbb{R}^{n \times n}$. Note that the alternative objectives can be easily achieved given the embeddings obtained from the first objective. Therefore, many methods start with the first objective, but there are also methods that directly get to the latter two goals. Meanwhile, the objective of data integration for multimodal datasets is to obtain a joint embedding, which can reduce the dimensions and preserve essential information from multi-modalities to improve cell state identification. Formally, given $k$ modalities $M_1 \in \mathbb{R}^{n \times d_1}$, $M_2 \in \mathbb{R}^{n \times d_2}$, $\ldots$, $M_k \in \mathbb{R}^{n \times d_k}$, where $d_i$ is the feature dimensions of the $i$th modality, and $n$ is the cell number. In this setting, the correspondence between cells among different modalities is already known. Therefore, for each cell, we have features from all the modalities. The goal is to find an embedding matrix $H \in \mathbb{R}^{n \times d}$ for all cells so that each cell would be represented by a $d$-dimensional vector in the downstream analysis.

One challenging problem for multimodal data integration is that the resulting embedding $H$ is hard to evaluate due to the lack of ground truth. One common evaluation method is to calculate Normalized Mutual Information (NMI) between embedding-based clustering results and predefined cell type labels. However, in most cases those cell type labels are acquired from clustering algorithms as well, resulting in a double-dipping issue. Another way is to manually evaluate the outcome of multimodal data integration in a specific downstream analysis. However, it is not conducive to a unified benchmark for comparison. One potential way to benchmark multimodal integration, as suggested in a recent work [197], is to leverage multi-omics aligned data, where two omics are simultaneously measured in each cell (e.g., CITE-seq [290]), to provide ground truth for multimodal integration. More specifically, to comprehensively evaluate the power of various integration methods, three key tasks are defined, i.e., modality prediction, modality matching, and joint embedding. The modality prediction task is to predict one omic from another omic. The modality matching task is to align observations in different modalities that are actually from the same cells, while the ground-truth label is given. The joint embedding task is to comprehensively evaluate the embedding $H$ by various metrics based on biological states of cells and batch effect removal.

Since the joint measurement technology is not yet widespread enough, most existing works focus on identifying cell correspondences across modalities, in order to support the joint analysis with multiple unimodal sources. We categorize this type of research as “data alignment among unimodal datasets” in the following subsection. Because multimodal assays are becoming more and more popular, we categorize methods that focus on leveraging aligned data as a separate category and discuss them in Section 8.2. We summarize representative tools or methods for multimodal integration in Table 8 and useful benchmarks in Table 9.

### 8.1 Data Alignment among Unimodal Datasets

The general idea of data alignment is to identify cell-to-cell correspondences or similarities. More concretely, given two or more datasets, the integration methods sought to find a unified interface for data points from all the datasets, so that downstream analysis can get rid of the discrepancy between different modalities. For instance, projecting all modalities into a shared latent space is one of the most popular methodologies in this task. Due to the uncertainty of ground truth, there are various underlying key assumptions for multimodal integration, which can be broadly divided into two categories. First, feature correspondence exists across modalities. For example, features from different modalities are controlled by the same gene or there exist certain correlations.
Table 8. A Summary of Multimodal Integration Tools

| Tool       | Algorithm | Description                                                                 | Language |
|------------|-----------|-----------------------------------------------------------------------------|----------|
| SCIM [286] | Classical | Autoencoders aligned with a adversarial discriminator                       | Python   |
| SOMatic [136] | SOM      | A SOM model that links scATAC-seq regions with scRNA-seq genes              | C++      |
| scJoint [186] | MLP      | MLP-based dimension reduction with regularization losses                     | Python   |
| SMILE [348] | MLP      | MLP with contrastive learning                                                | Python   |
| scDAT [367] | MLP      | MLP-based modality translator regularized by gene–activity matrix           | Python   |
| CMAE [353] | AE       | Autoencoders with adversarial loss, alignment loss, and prior               | Python   |
| FABEL [342] | AE       | Two encoders and two decoders for translation and reconstruction            | Python   |
| Cobolt [101] | AE      | Modality fusion with posterior mean of multimodal distributions             | Python   |
| scMM [221] | AE       | A mixture-of-experts multimodal variational autoencoder                     | Python   |
| scMVAE [376] | AE      | A PoE inference network with variational autoencoders                       | Python   |
| DCCA [377] | AE       | Autoencoders with cross-modal knowledge transfer                            | Python   |
| scMoGNN [338] | GNN     | A GNN model with cell-feature heterogeneous graph                          | Python   |
| GLUE [35]  | GNN      | An autoencoder framework with prior knowledge-based guidance graph          | Python   |
| scMoFormer [305] | Transformer | A transformer framework with multimodal heterogeneous graph            | Python   |
| MAGAN [7]  | GAN      | A GAN model with a reconstruction loss to enhance alignment                 | Python   |

Table 9. A Summary of Multimodal Integration Datasets

| Dataset       | Species       | Tissue                          | Dataset Size | Protocol                   |
|---------------|---------------|---------------------------------|--------------|----------------------------|
| sci-CAR [29]  | Human, Mouse  | Human lung, human kidney, mouse kidney | 4,825 aligned human cells, 11,296 aligned mouse cells | RNA-seq and ATAC-seq |
| PBMC CITE-seq | Human PBMC    | 35,455 aligned cells            | RNA-seq, ADT and HTO |
| 10x Genomics CITE-seq [369] | Human PBMC | 5,527 aligned cells | RNA-seq and ADT |
| Multiome Chromium X [369] | Human PBMC | 11,984 aligned cells | RNA-seq and ATAC-seq |
| Multiome unsorted [369] | Human PBMC | 12,012 aligned cells | RNA-seq and ATAC-seq |
| SNARE-seq [45] | Human, Mouse  | Human cell line mixture, mouse brain | 1,047 aligned human cells, 15,390 aligned mouse cells | RNA-seq and ATAC-seq |
| SHARE-seq [202] | Mouse Mouse skin | 34,774 mouse skin cells | RNA-seq and ATAC-seq |
| scNMT-seq [53] | Mouse embryonic stem cells, embryoid body cells | Less than 100 | RNA-seq, BSseq and ATAC-seq |
| Multimodal PBMC [220] | Human, Mouse Mainly PBMC | 4,644 cells of CITE-seq, 4,502 cells of ASAP-seq (after quality control) | RNA-seq, ADT and ATAC-seq |

between features. Typical approaches include feature correlation-based methods [136, 186, 367] and NMF-based methods [74, 139, 163, 190]. Second, an underlying manifold is shared among different modalities, or in other words, some shared low-dimensional latent factors exist in different modalities. Manifold alignment [32, 134, 285] and coupled clustering [358, 359] are well-received methods based on this assumption.

There are limitations to each of these assumptions. Feature correspondence assumption assumes that feature correspondence across modalities presupposes a degree of similarity or direct relationship that may not exist or be clearly identifiable. Features in different modalities might be influenced by a multitude of factors, not just gene control, leading to incomplete or misleading correspondence. In addition, for high-dimensional data spaces, like those in single-cell analysis, establishing accurate feature correspondence becomes increasingly complex. Lastly, many correlation-based methods and NMF-based approaches often rely on linear assumptions, which might not capture the complex, non-linear relationships present in multimodal data. On the other hand, manifold assumption that assumes a shared underlying manifold across modalities is often overly simplistic. Biological data can exhibit manifold structures that are highly complex and variable, making them difficult to align using standard manifold alignment or coupled clustering techniques. Meanwhile, manifold alignment methods can be sensitive to noise and outliers, which are common in single-cell datasets. They may also lack the flexibility to adapt to the unique characteristics of different datasets.
Although affected by the same limitations imposed by the assumptions, deep learning methods can alleviate some of these problems to a certain extent compared to traditional methods. Deep learning methods can automatically learn complex, non-linear mappings between features from different modalities. Techniques such as variational autoencoders or deep canonical correlation analysis can uncover underlying relationships that are not apparent through linear correlation or NMF methods. Additionally, deep learning models can handle high-dimensional data more effectively, learning abstract representations that traditional methods might miss. Furthermore, deep learning methods, aided by stochastic gradient descent and mini-batch sampling, are generally scalable to large-scale datasets containing millions of cells, a feat not consistently achievable with traditional methods. In the following subsections, we first introduce traditional methods and then mention more advanced deep learning methods.

8.1.1 Traditional Methods. The study of multimodal integration started with some classic algorithms, such as statistical modeling. MATCHER [337], one of the earliest works in this field, uses a Gaussian process latent variable model (GPLVM) to model high-dimensional single-cell data as a function of latent variables (e.g., pseudo time). scACE [187] develops a statistical model-based joint clustering method that is specifically designed for single-cell genomic data. scAMACE [334] extends scACE [187] to broader data types while performing joint clustering in a similar statistical way. MOFA+ [11] uses a GPU-accelerated efficient variational inference algorithm to infer a small number of latent factors as integrated representation.

The most popular group of methods is matrix factorization. scAI [139] introduces an adapted matrix factorization that is designed specifically for the integration of epigenetic and transcriptomic data. CoupledNMF [74] relies on an alignment between gene expressions and accessible chromatin regions, obtained from the paired expression and chromatin accessibility (PECA) model [73]. LIGER [190] introduces integrative non-negative matrix factorization (iNMF [354]), which jointly decomposes data matrices from different modalities to get a cell representation in a shared latent space. UINMF [163] follows iNMF and further separates the metagene matrix into shared and unshared components, thus enabling unshared features to inform the factorization. BindSC [71] aligns both cells and features with a bi-order canonical correlation analysis (CCA) algorithm, where rows and columns from two data matrices are iteratively aligned by CCA. Seurat v3 [293] applies CCA and mutual nearest neighbors (MNNs) to identify “anchors” (i.e., correspondences of cells across datasets). SingleCellFusion [198] uses CCA and a restricted MNN to impute features in one modality from another modality. Hence, it requires cell correspondences between modalities.

Another important group of methods is manifold alignment. Harmonic alignment [285] projects data points to a shared embedding using principal components analysis (PCA) and removes multi-dataset-specific effects iteratively. UnionCom [32] first generates a distance matrix within the same dataset. Then it matches the distance matrices across datasets via matrix optimization. MMD-MA [276] first introduces Maximum Mean Discrepancy (MMD) loss to achieve similar distributions in the latent space across different datasets. Then it adds two extra losses: one to preserve structure between the input space and the latent space, and the other to avoid collapse to a trivial solution. MultiMAP [134] recovers geodesic distances on a shared latent manifold that involves all datasets and then acquires low-dimensional embedding based on a neighboring graph on the manifold.

Some of the remaining methods also explore coupled clustering, which skips the step of projection to a low-dimensional latent space. DC3 [359] adds a coupling term to the cost function to improve joint clustering, where the coupling term is derived from empirical correlations between modalities. coupleCoC+ [358] is a co-clustering framework that requires feature correspondences but also takes advantage of unlinked features. Gradient-boosted regression [168] implements a
gradient-boosting classification tree model to predict clustering labels on one modality from another modality, correspondingly determining the correspondence between cells from the resulting decision tree.

### 8.1.2 Deep Learning Methods

Compared with traditional methods, deep learning methods are receiving increasing attention, and many of them are capable of jointly handling unimodal and multimodal datasets. They have advantages in handling high-dimensional data, mining non-linear relationships, and being scalable. Here, we introduce three representative deep learning methods that focus on the alignment of unimodal datasets.

SOMatic [136] links scATAC-seq regions with scRNA-seq genes using **self-organizing maps (SOMs)**. SOMs are a type of artificial neural networks that are referred to as Kohonen networks [158]. They are unsupervised methods to generate low-dimensional data representation. SOMatic starts with separate training and clustering SOMs for each modality and then links the clusters from both modalities via a linking algorithm from GREAT [214].

SCIM [286] matches cells from multiple datasets based on low-dimensional latent representations, which are obtained from an autoencoder framework with an adversarial objective. During training autoencoders for each modality, the latent spaces are aligned to have a comparable structure using a neural network discriminator and an adversarial loss. The autoencoders and the discriminator are simultaneously trained, resulting in a joint embedding space.

GLUE [35] initializes omics-specific cell embeddings with variational autoencoders. Then, in order to link the latent spaces for different modalities, GLUE makes use of a prior knowledge-based guidance graph to learn feature embeddings. They replaced the decoders in previous omics-specific autoencoders with an inner product between feature embeddings and cell embeddings. Furthermore, a discriminator is implemented to align the latent space of different omics via adversarial learning.

### 8.2 Data Integration for Multimodal Dataset

Increasingly sophisticated co-assay techniques, such as CITE-seq and SNARE-seq, bring us unprecedented multimodal data at the single-cell level. They provide new insights into the interaction between different modalities, as well as a comprehensive understanding of the cellular system. In light of this, a new group of methods is dedicated to making good use of paired data to improve cell state identification. In paired data, we observe features of more than one modality for every single cell. Therefore, the cell-to-cell correspondence is given, and the integration approaches can focus on learning high-quality data representations that encode the fundamental cellular states and their collective roles in tissues. In this category, we review representative methods that are able to take advantage of paired data. Some of them are completely supervised, while most of them are semi-supervised and utilize paired data to improve unpaired integration.

Most methods are deep learning methods, except Pamona [33]. Pamona is a non-deep learning approach. It uses a partial Gromov-Wasserstein distance-based manifold alignment framework to project multi-modalities to a shared latent space, preserving both common and modality-specific structures. The prevalence of deep learning methods is attributed to the unique challenges of this task. Paired multimodal data often involve heterogeneous types (e.g., transcriptomic and proteomic data), each with its own characteristics and scale. Integrating these diverse data types in a way that retains their unique biological signals poses a significant challenge. Meanwhile, it is also crucial to not only capture the commonalities but also preserve the unique, modality-specific structures that might be critical for understanding cellular functions. Compared to traditional methods, deep learning models can effectively process and integrate data of varying types and scales, making them particularly suited for multimodal datasets. They can learn to emphasize the important features of each modality while combining them into a coherent representation.
Among the deep learning methods, the vast majority of them are based on autoencoders, except for one GNN model [338]. We roughly divide all deep learning models into two classes according to whether cell correspondence is indispensable. The first class of methods is flexible enough to leverage both aligned and unaligned data. Cross-modal Autoencoders [353] uses autoencoders to map vastly different modalities (including images) to a shared latent space. Specifically, a discriminator and adversarial loss are added to force the distributions of different modalities to be matched in the latent space. To make use of prior knowledge, an additional loss term can further be added to align specific markers or anchoring cells. MAGAN [7] is one of the earliest works in this area featuring a very natural design. It consists of two GANs that learn mutual mappings between two domains. In addition to the traditional adversarial loss from generators and discriminators, a reconstruction loss is introduced. Namely, if the first generator can generate modality 2 from modality 1, then the second generator should generate modality 1 from modality 2, where the back-translation should be the same as the original features. To exploit the paired data, MAGAN can further involve a supervised loss for the generator. Cobolt [101] uses autoencoders as well to project two modalities into shared latent space. It fuses modalities and prior by taking the posterior mean of multimodal distributions as a summary of the shared representation passing to the decoder. Therefore, the framework is flexible to integrate both paired and unpaired data, and more than two modalities simultaneously. scDART [367] starts with a gene activity module, where a neural network transforms ATAC-seq data into scRNA-seq data and is regularized by a pre-defined gene–activity matrix (GAM). Next, a projection module projects data into low-dimensional latent space, where trajectory structure is preserved by adding a cell pair-wise distance loss. Besides, a Maximum Mean Discrepancy (MMD) loss is also added to project multiple datasets into shared space. With respect to paired multimodal data, an additional anchor loss can be added to utilize the alignment information. The advantage of scDART is the preservation of cell trajectories in continuous cell populations. scJoint [186] takes the gene activity score matrix and gene expression matrix as input. They first project data into latent space through neural-network-based dimension reduction, where three losses are jointly enabled to train an ideal encoder. Neural-network-based dimension reduction (NNDR) loss is used to force features in the low-dimensional embedding space to be orthogonal. A cosine similarity loss is used to maximize the similarity between the best-aligned multimodal data pair, which helps utilize alignment information. A cross-entropy loss with cell type annotations further enhances the representation of scRNA-seq datasets. After the projection, it transfers cell type labels through KNN in embedding space and further improves the modalities mixing with transferred cell type labels in a metric training loss. SMILE [348] uses contrastive learning to maximize the mutual information between original feature space and latent space in order to learn discriminative representations for cells, which is clustering friendly. Besides, its contrastive learning method utilizes paired data to improve integration.

The other class of models accepts only aligned data for training, although trained models can be used on unaligned data. BABEL [342] trains two neural-network-based encoders and two decoders on the paired data to translate data from one modality to the other and to reconstruct itself, thus eventually obtaining shared embedding. Both BABEL and Cross-modal Autoencoders consider autoencoders as the projecting function toward latent space, yet they use discriminative loss and translation loss to align embedding from multimodalities. scMM [221] leverages a mixture-of-experts multimodal variational autoencoder to explore the latent dimensions associated with multimodal regulatory programs. It models the raw count data of each modality via different probability distributions in an end-to-end way. scMVAE [376] adapts variational autoencoders to a Product of Expert (PoE) inference network to estimate the joint posterior from all omics, combined with a Gaussian Mixture Model (GMM) prior. Two interesting points are that the model uses library factors to normalize input expressions and re-scales the output, and instead of direct
reconstruction, the decoder predicts the parameters of a probabilistic distribution to fit the input expressions, which is the training objective. DCCA \cite{377} combines autoencoders for each modality with attention transfer that enables autoencoders to learn through mutual supervision across modalities based on the similarity of embeddings. Namely, instead of having a shared latent space, they use the scRNA-seq data to train a teacher network, guiding a student network that works with scATAC-seq data. scMoGNN \cite{338} constructs a cell-feature graph based on paired data to capture both cell similarity and gene similarity, incorporating feature relations as well. A graph encoder is applied to reduce the dimensionality. The proposed method works on large-scale paired data and achieves good performance on comprehensive benchmarking tasks. scMoFormer \cite{305} introduces a multimodal heterogeneous graph consisting of protein–protein interactions, gene–gene interactions, gene–protein encoding, and gene expression counts. Transformers are implemented on each modality, and GNNs are employed to incorporate both the prior information and the data-specific information.

9 SPATIAL DOMAIN

The spatial domain aims to partition spatial data into a series of meaningful clusters. Each identified cluster from this analysis is considered as a spatial domain. Spots in the same spatial domain are similar to each other and coherent in gene expression and histology, and are dissimilar to those in different spatial domains \cite{127}. Hidden spatial domains to be identified cannot be directly observed from gene expression or other data. A lot of methods study how to infer latent variables from those gene expressions along with other types of data to characterize the hidden spatial domains. The identification of the spatial domain plays a crucial role in medical analysis. Histological staining of cancer tissue slides can serve as a stimulating example \cite{145, 208}. Due to the varying affinities of staining agents, cancer areas and normal tissues can be discriminated visually. This enables pathologists to grade and stage individual slides of cancer tissue depending on the size and location of malignant spots. The inputs of the task consist of three components: (1) gene expression at the spot or cell level, (2) spot coordinates to indicate the spatial information of each spot, and (3) histology image. The goal of the task is to cluster or group spots based on the input features of each spot. In order to evaluate the effectiveness of various spatial domain techniques, ARI \cite{355} is used to determine how well an algorithm’s projected clustering labels match the actual labels. We summarize all tools or baseline methods for the spatial domain task in Table 10 and useful benchmarks in Table 11.

9.1 Traditional Methods

One of the broadly used traditional methods is based on the Louvain clustering method \cite{20}. It comes from the community detection family to extract the community structure from large networks. The normal scRNA-seq data clustering methods \cite{20, 143}, on the other hand, tend to focus on gene expression levels, whereas the spatial domain requires taking into account both spatial information and morphological image aspects. stLearn \cite{246} first develops a distance measure utilizing morphological similarity and neighborhood smoothing for gene expression normalization. The normalized data is then utilized to identify clusters that represent the transcriptional patterns of particular cell types and phenotypes. If cells are spatially separated, clusters are further sub-clustered. Apart from these, BayesSpace \cite{368} and a hidden Markov random field (HMRF) method \cite{374} together assume that latent variables can characterize hidden spatial domains from gene expression data, and the latent variables can’t be seen in the wild but inferred from gene expression data. BayesSpace is a Bayesian statistical method and assigns greater weights to geographical areas that are physically closer for resolution enhancement of spatial transcriptomics and for better clustering analysis. The computationally intensive Markov chain Monte Carlo
Table 10. A Summary of Spatial Domain Tools

| Tool       | Algorithm | Description                                                                 | Language | DANCE [66] |
|------------|-----------|-----------------------------------------------------------------------------|----------|------------|
| louvain    | Classical | Iterative modularity optimization for network community detection          | Python   | ✓          |
| K-means    | Classical | Iterative observation assignment to the cluster with the nearest mean      | Python   |            |
| stLearn    | Classical | Integration of spatial location, morphology, and gene expression to determine cell–cell interactions, cell types, and spatial trajectories | Python   |            |
| BayesSpace | Classical | A low-dimensional representation clustering with the gene expression and spatial information | R        |            |
| SpatialDE  | Classical | Gene identification using spatial coordinates                                | Python   |            |
| Giotto     | Classical | Giotto Analyzer and Viewer to process, analyze, and visualize single-cell spatial expression data | R        |            |
| SpaGCN     | GNN       | A GCN-based method via integrating gene expression and histology to find spatial domains and variable genes | Python   | ✓          |
| STAGATE    | GNN       | A graph attention autoencoder to learn low-dimensional latent embeddings from gene expression and spatial information | Python   | ✓          |
| SEDR       | GNN       | An autoencoder for learning gene representations while variational graph autoencoder for learning geographical information | Python   |            |
| CCST       | GNN       | A GCN-based unsupervised cell clustering method to improve ab initio cell grouping and find novel sub-cell types | Python   |            |

Table 11. A Summary of Spatial Domain Datasets

| Dataset                                           | Species | Tissue                                      | Dataset Dimensions | Protocol          |
|---------------------------------------------------|---------|---------------------------------------------|--------------------|-------------------|
| Mouse Posterior Brain 10x Visium Data [1]          | Mouse   | Posterior brain                             | 3,353 spots        | 10X Visium        |
|                                                   |         |                                             | 31,053 genes       |                   |
| LIBD Human Dorsolateral Prefrontal Cortex [237]    | Human   | Dorsolateral prefrontal cortex              | Slice 151673:      | 10X Visium        |
|                                                   |         |                                             | 3,639 spots        |                   |
|                                                   |         |                                             | 33,538 genes       |                   |
|                                                   |         |                                             | Slice 151507:      |                   |
|                                                   |         |                                             | 4,226 spots        |                   |
|                                                   |         |                                             | 33,538 genes       |                   |
| Human Primary Pancreatic Cancer ST Data [44]       | Human   | Primary pancreatic cancer tissue            | 224 spots          | Spatial Transcriptomics |
|                                                   |         |                                             | 16,448 genes       |                   |

(MCMC) component in BayesSpace, however, makes it impractical for use with high-throughput spatial transcriptomics data. An HMRF-based method [374] utilizes HMRFs to identify spatial domains. They first build a neighborhood graph to show how the captured locations are related spatially. The label of the immediate neighbor nodes affects the cell states. The HMRF model dictates that there is genetic and spatial cohesion within the clusters. The graph would be divided into several components via the HMRF model, and each component corresponds to a spatial domain. What’s more, spatial clustering can also be performed by the creation of a single-cell “neighborhood” matrix with columns providing neighborhood characteristics, such as the percentage of each cell type or the average gene expression within a radius [269, 277]. The matrix can then be directly clustered.
9.2 Deep Learning Methods

GNN-based methods have recently played a more and more important role in the task of the spatial domain. Even though several GNN models [68, 89, 127, 174] have been proposed on this task, their ways to construct graphs are different. CCST [174] constructs a graph only based on spatial information, and each node in the graph represents one spot. Gene expression would be considered as the initial node representation for further feature aggregation. Clustering methods would be finally performed based on newly learned spot representation after feature aggregation. SpaGCN [127] first constructs an undirected weighted graph of spots from the spatial transcriptomics data. The edge weight between two nodes would be determined by the distance between them. Then GCN [152] is utilized to aggregate spot gene expression from neighborhoods and update spot gene expression. Finally, the unsupervised clustering method would be applied to new learned spot gene expression for clustering iteratively. Different from the aggregation method used in SpaGCN, STAGATE [68] is a graph attention-based autoencoder framework to learn low-dimensional latent embeddings with both spatial information and gene expressions. With the help of the attention mechanism, which is located in the middle layer of the encoder and decoder, it would be easier to learn the edge weights of the networks and employ them to update the spot representation via collective aggregation of information from neighbors. SEDR [89] is a spatial embedding representation of both gene expression and spatial data. SEDR first uses a deep autoencoder to create a low-dimensional latent representation of gene expression and spatial data. SEDR first uses a deep autoencoder to create a low-dimensional latent representation of gene expression, and then it uses a variational graph autoencoder to integrate this representation with the accompanying spatial information.

10 CELL TYPE DECONVOLUTION

Cell type deconvolution is the task of estimating cell-type proportions in mixed-cell RNA sequencing data, such as bulk RNA-seq or spatial transcriptomics data. In particular, a primary goal of spatial transcriptomics is to determine cellular structure across tissues. While single-cell RNA sequencing can help determine cell types within a tissue, it does not spatially tag the cells, and thus cannot provide a mapping of cell types within the tissue. On the other hand, spatial transcriptomics technologies measure gene expression of small (spatially tagged) regions, but most platforms do not have single-cell resolution capabilities and do not annotate cell types within these regions. This makes it difficult to determine cell type proportion from the spatial transcriptomics data, which motivates the task of cell type deconvolution on spatial transcriptomics data. The inputs of the task consist of four components: (1) mixed-cell gene expression (to be deconvoluted), (2) reference scRNA-seq, (3) (optional) spot coordinates to indicate the location within the tissue of each spot, and (4) (optional) histology image.

Typically, cell type deconvolution models are evaluated on datasets with ground-truth cell type proportions using mean squared error (MSE), mean absolute error (MAE), correlation, cross-entropy, and Jensen-Shannon divergence (JSD). In most cases, however, non-simulated datasets do not have ground-truth cell type proportions. In this unsupervised setting, if profiled marker proteins are also provided with the dataset, one evaluation metric [62] is the correlation between predicted cell type proportions and the respective marker proteins. We summarize representative tools or baseline methods for the cell type deconvolution task in Table 12 and useful benchmarks in Table 13.

10.1 Traditional Methods

Most traditional methods for cell type deconvolution are based on non-negative linear regression. The most basic method is NNLS, where some reference scRNA-seq gene expression is used to create a cell profile matrix, which is then regressed onto the mixed-cell gene expression. The resulting
A non-negative matrix factorization of an annotated scRNA reference matrix.

### DWLS [317] Classical
Weighted NNLS: dampened weighting is applied to genes

### SpatialDWLS [69] Classical
A subset of cell types chosen via PAGE enrichment analysis

### SpatialDecon [62] Classical
A multiplicative log-normal error model

### cell2location [157] Classical
Bayesian hierarchical model of spatial expression counts with a spatially informed prior on cell-type compositions

### CARD [205] Classical
Conditional autoregressive based model that incorporates spatial correlation of cell-type composition

### RNA-Sieve [61] Classical
A likelihood-based inference model that estimates cell-type proportion through a maximum likelihood method

### Scaden [217] GNN
A graph convolutional network whose graph is constructed on Mutual Nearest Neighbors of low-dimensional embeddings of simulated and real mixed-cell data

### DSTG [282] GNN
A graph convolutional network whose graph is constructed on Mutual Nearest Neighbors of low-dimensional embeddings of simulated and real mixed-cell data

| Dataset | Species | Tissue | Dimensions | Protocol |
|---------|---------|--------|------------|----------|
| Mouse Posterior Brain 10x Visium Data [369] | Mouse | Posterior brain | 3,353 spots 31,053 genes | 10X Visium |
| Mouse Olfactory Bulb [369] | Mouse | Olfactory bulb | 1,185 spots 11,176 genes | 10X Visium |
| HEK293T and CCRF-CEM cell line mixture [62] | Human | Pancreas | 56 mixtures 1,414 genes | NanoString GeoMx |
| Human PDAC [223] | Human | Pancreas | 1,819 spots 19,738 genes | Spatial Transcriptomics |

(non-negative) coefficients are then used as the cell type proportion predictions. Here, the idea is that the single cells’ expression will aggregate linearly, respective to their proportion in the mixed-cell sample. The cell profile or signature matrix is typically constructed through the median or mean across cells within each cell type of interest. Most other traditional methods build on this key idea.

NMFReg [258] uses non-negative matrix factorization to construct a basis in the lower-dimension gene space, which is then used for the deconvolution of the mixed-cell data via NNLS. Building on NMFReg, SPOTlight [78] uses non-negative matrix factorization to produce the cell-topic profile matrix. However, it introduces a spot-topic profile matrix, which is then regressed onto the cell-topic profile matrix instead of the full mixture/spot data. DWLS [317] applies a dampened weighting scheme to the standard NNLS framework. Here, each gene’s error term is weighted by the squared inverse of the predicted mixed-cell expression level. This is done to reduce bias toward highly expressed genes, or genes that are highly represented across cell types. Extending on DWLS, SpatialDWLS [69] first includes the first step of enrichment analysis (using the PAGE method). The fold change for each gene’s expression value is computed between each spatial location and mean expression over all spatial locations, which is then standardized and gives an enrichment score. A threshold is then applied to the enrichment scores to determine the subset of

---

ACM Trans. Intell. Syst. Technol., Vol. 15, No. 3, Article 40. Publication date: March 2024.
cell types to infer to. Altering the classical assumption of an additive error linear model, SpatialDecon \cite{62} implements a non-negative linear regression-based method that assumes a log-normal multiplicative error model.

The first method to incorporate the spatial information of spatial transcriptomics bulk expression regions is **Conditional AutoRegressive Model-based Deconvolution (CARD)** \cite{205}. As with most classical methods, CARD assumes a linear model between the mixed-cell expression matrix and a cell profile matrix, constructed from a single-cell reference. However, spatial information is incorporated through a conditional autoregressive (CAR) assumption on the model, which applies an intrinsic prior on the cell type proportions by modeling proportions in each location as a weighted combination of proportions in all other locations. The weights in CARD are given by a Gaussian kernel function. Then a maximum likelihood framework is adopted to optimize the model.

RCTD \cite{27} and cell2location \cite{157} are two methods that statistically account for platform effects and other sources of gene expression variations to model cell type compositions in spatial transcriptomic data. As scRNA-seq data are used as references for cell type composition estimation of spatial transcriptomic data, the differences in the library preparation of the two sequencing technologies could lead to systematic bias that confounds cell type deconvolution results, termed as platform effects \cite{27}. RCTD applies a Poisson distribution to model observed gene counts, and its rate parameter accounts for cell type expression profiles, spot-specific effect, and gene-specific platform effect that follows a normal distribution on the log scale. These parameters are then estimated via MLE. In particular, RCTD assumes that in many cases there are only a few cell types co-present in a spot, and thus supports a "doublet mode" that restricts the number of cell types present in a given spot to mitigate overfitting. By comparison, cell2location takes a Bayesian approach and models spatial expression counts using a NB distribution. Similar to RCTD, the mean parameter in the NB distribution used by cell2location incorporates gene-specific platform effects, an additive shift, and cell type profile contributions, adjusted by spot-wise sensitivity. All parameters, including cell type proportions, are endowed with informative priors. In particular, cell2location acknowledges that cell type compositions are correlated across locations. Thus, it factorizes cell type proportions into the contribution of multiple latent cell type groups to pool information across spots and thus enhance statistical strength.

### 10.2 Deep Learning Methods

Some recent developments in cell type deconvolution have applied deep-learning-based methods. These approaches typically apply a transfer learning scheme wherein they first simulate mixed-cell data from scRNA reference data and use a common network to predict the cell type composition of both the simulated and real mixed-cell data. A notable feature of the deep-learning-based methods is that they model the cell type compositions directly; i.e., the model objective is on the predicted cell type compositions. This contrasts with most classical methods, where the predicted cell type proportions are the optimal parameters/coefficients from some regression model.

One of the early deep learning approaches to the cell type deconvolution problem is the Scaden \cite{217} method. First, scRNA reference data is randomly sampled from scRNA reference data to generate simulated mixed-cell samples. A fully connected network is then trained to predict the true cell type compositions of the simulated mixed-cell data, with cross-entropy loss function. This trained model is then applied to the real mixed-cell data to get cell type compositions. Building on this approach, DSTG \cite{282} is a GNN-based method, modeling similarities in expression between different mixed-cell samples. DSTG also generates simulated mixed-cell data from scRNA reference data. Different from Scaden, before training the GNN on this data, DSTG first aligns the simulated and real mixed-cell data in a lower-dimensional gene space using CCA. These embeddings are
then used to construct a graph by considering MNN as adjacent in the graph. Here, adjacencies can be between simulated-to-real and real-to-real samples.

## 11 CELL SEGMENTATION

Single-cell segmentation aims to achieve cell separation by generating pixel-wise annotations for individual cells in image-based single-cell profiling. Unlike scRNA-seq, tissue imaging obtained from microscopy and stained by various nuclear or membrane markers captures intact specimen imaging of cells and preserves valuable spatial information for transcriptionally distinct cell types and cell states from tissue images at single-cell resolution.

Accurate segmentation of single cells from these microscopy images helps to identify cell morphology and distribution in their spatial context, which benefits cancer diagnosis and clinical treatment. More important, single-cell segmentation not only defines whether a pixel is inside or outside of a cell but also assigns each pixel to each cell. This allows accurate localization of every single cell in images and offers significant implications for the feasibility of cellular-level analysis for downstream tasks like cell type identification and RNA/protein expression quantification [105, 292]. For RNA hybridization-based approaches without measuring the belongings of RNA to which cells, single-cell segmentation is an essential prerequisite for such mRNA assignment. Thus, the RNA detection can be converted into spatial single-cell data [189].

Due to the variety of cell types and microscopy instruments and the large number of cells, manual annotation of the whole cell boundaries in microscopy images is highly time-consuming and labor-intensive. It becomes much more challenging to separate adjacent cells when tissue cells are tightly packed with low signal-to-noise ratio staining. Therefore, different imaging-based methods have been widely proposed to realize accurate automatic or semi-automatic segmentation of single cells. Additional cellular or gene information indicating intracellular variability has also been integrated to improve single-cell segmentation performance. In general, the inputs of the task are microscopy images with target cells, and the goal is to obtain an intact whole-cell boundary for each cell. To assess the segmentation performance, a one-to-one matching process will be first performed between predicted cells and ground-truth cells at different matching thresholds, i.e., standard intersection over union (IoU) metric. Several commonly used evaluation metrics such as precision, recall, F1 score, Jaccard index, and standard average precision (AP) metric can be further applied to quantify the segmentation results at the single-cell level. We summarize representative tools or baseline methods for single-cell segmentation task in Table 14 and useful benchmarks in Table 15.

### 11.1 Traditional Methods

Early works typically adopt ideas from the traditional computer vision area. Unlike other segmentation tasks, densely clustered cells suffer from overlapping cell contours. To overcome this problem, Yan et al. [352] developed an automatic interaction model to segment cells from high-throughput RNAi fluorescent cellular images. They first modify the watershed algorithm [323] to segment cell nuclei regions and then adopt a multiphase level set method to predict the boundaries of neighboring cells further. To perform cellular analysis on epithelial tissues, a similar two-stage pipeline, MxIF [94], is completed to generate segmentation masks for fluorescent images. Specifically, a wavelet-based detection algorithm is utilized to segment the nucleus, followed by a variation of the probabilistic method [28] for whole-cell segmentation. The above methods, based on pre-selected nuclei/membrane fluorescent markers, fail to use multiple, automatically selected markers. Therefore, built on the watershed method, Schüffler et al. [270] designed an automatic marker selection manner with Spearman’s rank correlation to capture multiplexed imaging information and reveal the most useful channels for cell segmentation. Some other works incorporate
Table 14. A Summary of Cell Segmentation Tools

| Tool            | Algorithm | Description                                                                 | Language          |
|-----------------|-----------|------------------------------------------------------------------------------|-------------------|
| ImageJ [323]    | Classical | Watershed based                                                              | Python            |
| Schüffler et al. [270] | Classical | Watershed based                                                              | MATLAB etc.       |
| Ilastik [280]   | Classical | Random Forest classifier to perform a semantic segmentation and return a probability map of each class, which can be further transformed into individual objects by other methods | Python            |
| MorphoLab [170] | Classical | Watershed + GUI for 2D/3D images of cell tissues                               | Java              |
| CellProfiler [141] | Classical | Primary objects identification first (often nuclei) and then secondary object identification (cell edges) | Python            |
| MATURE [164]    | Classical | Combination of fluorescence microscopy and multiplex capability of IMC into a single workflow, nuclei, and whole cells identification based on probability maps generated by Ilastik | R                 |
| Oufti [236]     | Classical | Laplacian of Gaussian, valley, and cross-detection                            | MATLAB            |
| Cellpose [292]  | CNN       | U-Net with an auxiliary representation                                        | Python            |
| Mesmer [105]    | CNN       | A CNN architecture that generalizes to the full diversity of tissue types     | Python            |
| Baysor [245]    | CNN       | Segmentation performed using only molecule placement data or in combination with evidence from auxiliary stains in images | Python            |
| JSTA [189]      | CNN       | Integration of spatial transcriptomics and scRNAseq information to improve segmentation | Python            |
| MIRIAM [213]    | CNN       | Initial cell identification by random forest and cell shapes characterization via an autoencoder | Python           MATLAB |

Table 15. A Summary of Cell Segmentation Datasets

| Dataset   | Species | Tissue | Dataset Dimensions                                                                 | Protocol                          |
|-----------|---------|--------|-----------------------------------------------------------------------------------|-----------------------------------|
| ISBI-14   [195] | Human  | Cervix | ISBI-14: 16 EDF real cervical cytology images and 945 synthetic images             | Microscope                        |
| ISBI-15   [195] |        |        | ISBI-15: 17 multi-layer cervical cell volumes                                     |                                   |
| Cellpose  [292] | Various species | Various tissues | 668 images                                                                       | Microscope                        |
| TissueNet [375] | Human  | 9 organs | >1 million cells with nuclear                                                      | Immersion objective and dual Photometrics Prime |
| EVICAN    [238] | Human  | Various tissues | 4,600 images, ~26,000 segmented cells                                              | Microscope                        |
| LIVECell  [76]  | Human, Mouse | Various tissues | 5,239 images, 1.6 million cells                                                   | Phase-contrast microscopy         |

cell shape as prior knowledge to enhance segmentation performance. For example, learning from user annotations of the given training data, SVSS [267] serves as a tissue statistic shape model to enhance segmentation performance and quantify the quality of cell contours. Matisse [164] is a unified workflow based on existing open-source technologies. It combines multiple marker information from IMC data and fluorescence microscopy data to boost segmentation performance. They use the Ilastikuse tool [280] to generate probability maps, which are then fed into CellProfiler software [141] to acquire the final segmentation maps.

11.2 Deep Learning Methods

Deep-learning-based methods could be of great use in assisting single-cell segmentation, as they have shown great success for natural and other medical imaging applications [39, 117, 142, 191, 255, 351]. Existing deep-learning-based methods can be categorized into two groups according to their utilized data information and input modality. One is driven by cell morphological characteristics shown on images, which highly depend on the imaging quality and cell distributions. CNN is a fundamental deep neural network for the computer vision task of image segmentation. One of the most popular CNNs, U-Net [261], has shown significant success in many medical image segmentation tasks [118, 176, 200, 233, 234, 332, 333, 350]. It adopts a U-shape encoder-decoder architecture where the encoder serves as a contraction to capture semantically image contextual features. The decoder is a symmetric expanding path recovering spatial information. The two paths are connected using skip connections to recombine with essential high-resolution features from
Deep Learning in Single-cell Analysis

the encoder. To perform a cell segmentation task, the inputs to the U-Net are multi-channel images with different channels corresponding to nuclei or cytoplasm. After passing through the encoder and decoder, the model will generate a predicted probability map indicating that a pixel is inside or outside a cell. Since U-Net performs a pixel-based semantic segmentation and considers a cell segmentation task as a binary classification of pixels into cell foreground and background, it is prone to fail in overlapping cells, which need to be differentiated as individual object instances. Generally, traditional algorithms such as watershed will be combined to locate boundaries and separate touching cells. However, they show little improvement when cells are tightly clustered and stacked with similar brightness contrast. Although a commonly used CNN architecture for instance segmentation, Mask RCNN \[116\], has been dominant in generating a high-quality segmentation mask for each instance in natural imaging tasks, it is ill-equipped to handle high cell density. Recent works carefully incorporate another boundary detection task \[43\] or introduce contour-aware modules to the decoder path \[39, 128, 370\] for edge refinement. However, they are limited to cell nuclei, and thus giving a clear distinction between highly packed whole cells is still challenging in single-cell segmentation.

Transforming such a pure classification task into a regression model has been proven to be a feasible solution. Instead of directly categorizing each pixel into either foreground or background classes, a regression model allows us to predict a continuous variable (such as gradient and distance value) for each pixel. Cellpose \[292\] designs an auxiliary predictive U-Net head to predict two spatial flows for cells, i.e., horizontal and vertical flows, generated by spatial gradients of an energy function. The vector gradient flows lead all pixels toward the horizontal/vertical center of the cells they belong to, and these reversible flow maps are powerful representations to reconstruct cells with complex shapes and densities at test time. This framework is further extended to Cellpose 2.0 \[291\], which allows users to adapt their pre-trained segmentation models to their new datasets. Van Valen et al. \[321\] build a deep convolutional neural network to segment single cells in live-cell imaging, which indicates multiple cell types across the domains of life. Another regression model, DeepDistance \[162\], is trained to concurrently learn two distance metrics for each pixel in live-cell images. This multi-task strategy turns out to be effective in capturing shared feature representations to enhance a more accurate cell localization.

To deal with multiple imaging modalities and cell types, a large cell segmentation dataset, TissueNet \[105\], is constructed from nine organs and six imaging platforms. Specifically, based on this comprehensive labeled training dataset, they apply a more specified and more robust structure, Mesmer, with two semantic segmentation heads to predict the boundary of cells and two heads to predict the distance of each pixel within a cell to the cell centroid. Their model is built on a ResNet50 \[117\] backbone coupled to a Feature Pyramid Network \[185\] and trained in an end-to-end manner. Each cell’s obtained centroid and boundary are utilized to generate a final instance segmentation mask using a watershed algorithm. However, these algorithms are still insufficient for highly multiplexed imaging platforms. MIRIAM \[213\] is another pipeline that can deal with any multiplexed imaging data with various marker compositions. Initial segmentation masks are first generated by a random forest pixel classification method and then a seed-based watershed algorithm is applied. To deal with cells with irregular cell shapes and incomplete membrane markers, they extend the cell membranes by “connecting the dots” and regulate the cell shapes using an autoencoder neural networking. Their framework is validated to be robust to marker composition and image modality changes.

The above deep-learning-based methods in the first category have achieved significant performance, with more accurate and proper whole-cell boundary definition and adjacent single-cell separation. However, these models can only characterize cell morphology from its exhibited appearance in fluorescent images, which is sensitive to cell intensity and staining quality.
The other category introduces spatial gene expression patterns as extra supervision since the spatial dependency captured by gene information plays a crucial role in defining pixels at the junction of different cell types. Baysor [245] considers the RNA transcriptional composition and optimizes the joint likelihood of transcriptional composition and cell morphology. Specifically, each cell is modeled as a distribution of both spatial position and gene identity of each molecule. The mixture of all the cell distributions is regarded as a Bayesian mixture model (BMM), which is further optimized with Markov random fields (MRFs) before ensuring spatial clustering. JSTA [189] enhances image-based cell segmentation with the assistance of cell-type-specific gene expression. Motivated by the ability to convert raw spatial transcriptomic data into a single cell-level spatial expression map, the authors first train a DNN classifier to assign cell type probabilities to each pixel. These probabilities are utilized to adjust pixels around the cell borders iteratively. In these strategies, RNA information serves as valuable prior knowledge to constrain cell boundaries between neighboring cells.

12 CHALLENGES AND FUTURE WORK

Deep learning methods have recently proven highly effective in single-cell analysis, broadening the horizon of both qualitative and quantitative studies. As we delve deeper, we encounter emerging future directions accompanied by new challenges that warrant exploration. In the following contents, we will elaborate on three perspectives: multi-source integration, knowledge transfer, and enhancing general understanding through foundational models.

Throughout this review, we categorize the tasks and methods according to the data sequencing technologies. Given the increasing volume of data from diverse sources, it becomes essential to efficiently integrate cross-source information for a more profound comprehension of the underlying regulatory mechanisms within cellular systems. As discussed in Section 8, there are some existing works utilizing the deep learning framework to tackle the integration problem across modalities [7, 35, 101, 186, 286, 342, 353, 367]. While the majority of these works focus on downstream computational tasks, it is still unclear how deep learning can contribute to interpreting biological principles or offering guidance for experimental design. The field could benefit from novel deep learning approaches that bring in biological insights.

While current single-cell sequencing technologies provide vast amounts of data, certain experimental data remains limited. As precise condition control in single-cell experiments demands significant time and resources, there is a desire for computational methods that can extrapolate known results to unseen conditions [194, 262]. One representative task is perturbation prediction, which aims to predict transcriptional outcomes of diverse experimental perturbations [121, 193]. Understanding the transcriptional response of a cell to genetic perturbations provides fundamental insights into its functioning [155]. Meanwhile, predicting chemical perturbations offers valuable insights for drug discovery and drug repurposing [283]. Advancements in single-cell technologies empower us to delve deeper into phenotypes, a realm that has remained relatively unexplored. Deep learning methods oriented toward this perspective hold the potential of providing causal understanding for various applications, such as protein signaling [266].

Turning back to existing single-cell data, a fundamental question arises: can we attain a general understanding of the “languages” spoken by cells? Significant efforts have recently been dedicated to developing foundation models in the single-cell domain [59, 102]. Employing a pre-training and fine-tuning strategy, these foundation models leverage large-scale datasets and achieve state-of-the-art performance across various downstream computational tasks. However, there is room for improvement in existing foundation models. They primarily center on scRNA-seq data, neglecting the influences from other modalities. Additionally, despite their proficiency in computational tasks,
these models still fall short in providing comprehensive biological understanding. Further efforts are needed to fully unravel the "languages" of cells.

13 CONCLUSION

In this work, we have provided a comprehensive survey of deep learning in single-cell analysis. We started with a thorough introduction to single-cell technologies, followed by an in-depth discussion of popular deep learning concepts. We then presented the common pipeline of single-cell analysis. Then we delved into existing deep learning methods in various single-cell computational tasks. Finally, we highlighted potential directions for future research.

APPENDICES

A ADDITIONAL DETAILS OF DEEP LEARNING CONCEPTS

In this section, we present the details of some other deep learning concepts that are related to those mentioned in Section 3. Particularly, we will cover the Recurrent Neural Network (RNN) and some of its specifications, such as Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU).

A.1 Recurrent Neural Network

RNN [216] is a type of artificial neural network that deals with sequential data or time series data. In traditional neural networks, the input and output are independent of each other, whereas in RNN, the output is dependent on previous elements in the sequence. As shown in Figure 11, a sequence of tokens with length \( n \) is an input of the model. Each token is first transformed into vector form, and then the input sequence can be denoted as \( (x^{(1)}, x^{(2)}, \ldots, x^{(t)}, \ldots, x^{(n)}) \). The RNN model takes in a sequence of data and applies a chunk of neural networks to it one data point at a time. Here, the chunk of neural networks in RNN is a single neural network layer. The inputs of the chunk of neural networks are the output from the preceding position as \( h^{(i-1)} \) and the current element in the sequence as \( x^{(i)} \). The outputs are the output layer as \( y^{(i)} \) and \( h^{(i)} \) to be transmitted to the next position. The formulation for dealing with the \( i \)th element in the sequence can be written as

\[
\begin{align*}
    h^{(i)} &= \sigma_h \left( W_{hh} \cdot h^{(i-1)} + W_{hx} x^{(i)} + b_h \right), \\
    y^{(i)} &= \sigma_y \left( W_{yh} h^{(i)} + b_y \right),
\end{align*}
\]  

(7)  (8)

where \( W_{hh}, W_{hx}, \) and \( W_{yh} \) are learned to perform linear transformations on the output of the previous position, current element in the sequence, and output of the current element processing, respectively; \( b_h \) and \( b_y \) are the bias terms; and \( \sigma_h() \) and \( \sigma_y() \) are the activation functions. \( h^{(0)} \) is usually initialized as vector \( 0 \). We can see that when dealing with the current element, it not only considers the current element but also receives information transmitted from the preceding position in Equation (7). The neural network layer in the chain structure is identical for each element processing in the sequence.

When the RNN processes long sequential data, two problems may be encountered during the training phase: vanishing gradient [122] and exploding gradient [239]. The vanishing gradient problem refers to the issue of disappearing gradients. Gradients contain information used by the RNN, and when they become insignificantly small, parameter updates are rendered insignificantly. This makes it harder to learn long data sequences. If the slope continues to rise exponentially rather than decrease, this phenomenon is commonly referred to as an exploding gradient. This issue emerges when significant error gradients build, resulting in very large modifications to the weights.
Fig. 11. An illustration of recurrent neural networks.

of the neural network model during training. To address these problems, gated RNN models like LSTM networks [123] and GRU [50] have been developed.

A.1.1 Long Short Term Memory. LSTM networks [123] are a special kind of RNN, capable of learning long sequence data. Similar to RNN, LSTM also uses a chain structure, with the same neural network blocks being applied to the sequence’s individual elements. The block neural network in LSTM is much more complicated than that in RNN, where the block neural network is just a simple single neural network layer. Instead, a collection of gating units are used to control the information flow in LSTM, which is the essential distinction. A block of LSTM is shown in Figure 12. The information flowing across successive positions in a sequence consists of the cell state $C^{(t-1)}$ and the hidden state $h^{(t-1)}$ corresponding to the horizontal lines running through the top and bottom of the diagram, respectively. The cell state functions as the information from prior states that is transmitted to the subsequent location, and the hidden state contributes to the determination of how the information should be propagated. The LSTM is capable of removing or adding information to the cell state under the careful control of structures known as gates. Gates are a way to allow information to pass through if desired. A sigmoid activation function and a pointwise multiplication procedure construct them. The sigmoid activation function outputs integers between 0 and 1 that describe the percentage of each component that should be allowed through. A value of 0 indicates "permit nothing to pass through," while a value of 1 indicates "let everything pass through!" To protect and control the cell state, an LSTM has three of these gates, which are forget gates, input gates, and output gates, introduced below:

- **Forget Gate:** The initial phase of the LSTM is determining what information from the prior cell state will be discarded. The choice is determined by a forget gate. The forget gate generates a value between 0 and 1 for each element of the previous cell state $C^{(t-1)}$ based on the prior hidden state $h^{(t-1)}$ and the current input $X^{(t)}$. The forget gate can be defined as

$$f_t = \sigma \left( W_f \cdot x^{(t)} + U_f \cdot h^{(t-1)} + b_f \right), \quad (9)$$

where $W_f$ and $U_f$ are the learned matrices, $b_f$ is the bias term, and $\sigma()$ is the sigmoid activation function.

- **Input Gate:** The following step is to determine what information will be stored in the cell state from new input $x^{(t)}$. In a manner similar to the forget gate, the input gate is designed from the sigmoid activation function to make the decision. It is calculated as below:

$$i_t = \sigma \left( W_i \cdot x^{(t)} + U_i \cdot h^{(t-1)} + b_i \right). \quad (10)$$


Next, a vector of potential new values as $\tilde{C}^{(t)}$ is generated after a tanh layer, and it may be added to the cell state. The process of generating $\tilde{C}^{(t)}$ is as

$$\tilde{C}^{(t)} = \tanh \left( W_c \cdot x^{(t)} + U_c \cdot h^{(t-1)} + b_c \right).$$  \hspace{1cm} (11)

In the next step, we’ll combine these two to create an update $C^{(t)}$ to the cell state as

$$C^{(t)} = f_t \odot C^{(t-1)} + i_t \odot \tilde{C}^{(t)},$$ \hspace{1cm} (12)

where $\odot$ denotes element-wise multiplication.

• **Output Gate:** As the last step, a hidden state $h^{(t)}$ is generated, and it can flow to the position that comes after it and serve as the position’s output. The hidden state is derived from the updated cell state $C^{(t)}$, and an output gate chooses which portions of the cell state are to be maintained. The output gate is defined as

$$o_t = \sigma \left( W_o \cdot x^{(t)} + U_o \cdot h^{(t-1)} + b_o \right).$$ \hspace{1cm} (13)

The new hidden state $h^{(t)}$ is then defined as

$$h^{(t)} = o_t \odot \tanh \left( C^{(t)} \right).$$ \hspace{1cm} (14)

A.1.2 Gated Recurrent Unit. GRU [50] is a variant of LSTM but has fewer parameters than LSTM. In GRU, update gate and reset gate are employed to alleviate the problem of vanishing gradient, which is typical in traditional RNNs. In their most basic form, they are two vectors that control the information that is passed on to the output. A block of GRU is shown in Figure 13, and those two gates are described below:

• **Update Gate:** Similar to forget and input gates in LSTM, the update gate in GRU helps the model estimate how much information from the past (the previous position or time step) must be transmitted to the future. This is extremely potent since the model can choose to reuse all it’s ever learned and avoid the vanishing gradient problem. The update gate is formulated as

$$z_t = \sigma \left( W_z \cdot x^{(t)} + U_z \cdot h^{(t-1)} + b_z \right),$$ \hspace{1cm} (15)

where $\sigma$ is employed to squash the output between 0 and 1 to determine which information to discard from $h^{(t-1)}$ and which information to add from $x^{(t)}$.

• **Reset Gate:** The reset gate is another gate that determines how much historical data to forget. The reset gate is formulated as

$$r_t = \sigma \left( W_r \cdot x^{(t)} + U_r \cdot h^{(t-1)} + b_r \right).$$ \hspace{1cm} (16)
Next, the reset gate would be employed to store the relevant information from the past in a new memory content. The calculation is as follows:

$$\tilde{h}^{(t)} = \tanh \left( W \cdot x^{(t)} + U \cdot (r_t \odot h^{(t-1)}) + b \right).$$ (17)

As the final stage, the network must calculate the final output $h^{(t)}$—a vector that contains information about the current time step and transmits it to the network. To accomplish this, the update gate is required. It selects which information to acquire from the present time step $\tilde{h}^{(t)}$ and which information to collect from the preceding steps $h^{(t-1)}$.

$$h^{(t)} = (1 - z_t) \odot \tilde{h}^{(t)} + z_t \odot h^{(t-1)}. $$ (18)

REFERENCES

[1] 10x Genomics. 2019. Mouse Posterior Brain 10x Visium Data. https://support.10xgenomics.com/spatial-gene-expression/datasets/1.0.0/V1_Mouse_Brain_Sagittal_Posterior

[2] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. TensorFlow: A system for Large-Scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI ’16). 265–283.

[3] Tamim Abdelaal, Lieke Michielsen, Davy Cats, Dylan Hoogduin, Hailliang Mei, Marcel J. T. Reinders, and Ahmed Mahfouz. 2019. A comparison of automatic cell identification methods for single-cell RNA sequencing data. Genome Biology 20, 1 (Dec. 2019), 194. https://doi.org/10.1186/s13059-019-1795-z

[4] Mike Adam, Andrew S. Potter, and S. Steven Potter. 2017. Psychrophilic proteases dramatically reduce single-cell RNA-seq artifacts: A molecular atlas of kidney development. Development 144, 19 (2017), 3625–3632. Data Link: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE94333

[5] Axel A. Almet, Zixuan Cang, Suoqin Jin, and Qing Nie. 2021. The landscape of cell–cell communication through single-cell transcriptomics. Current Opinion in Systems Biology 26 (2021), 12–23.

[6] Jose Alquicira-Hernandez, Anuja Sathe, Hanlee P. Ji, Quan Nguyen, and Joseph E. Powell. 2019. scPred: Accurate supervised method for cell-type classification from single-cell RNA-seq data. Genome Biology 20, 1 (2019), 264. https://doi.org/10.1186/s13059-019-1862-5

[7] Matthew Amodio and Smita Krishnaswamy. 2018. MAGAN: Aligning biological manifolds. In International Conference on Machine Learning. PMLR, 215–223. Code Link: https://github.com/KrishnaswamyLab/MAGAN

[8] Christof Angermueller, Stephen J. Clark, Heather J. Lee, Iain C. Macaulay, Mabel J. Teng, Tim Xiaoming Hu, Felix Krueger, Sébastien A. Smallwood, Chris P. Ponting, Thierry Voet, Gavin Kelsey, Oliver Stegle, and Wolf Reik. 2016. Parallel single-cell sequencing links transcriptional and epigenetic heterogeneity. Nature Methods 13, 3 (Jan. 2016), 229–232 https://doi.org/10.1038/nmeth.3728

[9] Christof Angermueller, Heather J. Lee, Wolf Reik, and Oliver Stegle. 2017. DeepCpG: Accurate prediction of single-cell DNA methylation states using deep learning. Genome Biology 18, 1 (2017), 1–13.

[10] Dvir Aran, Agnieszka P. Looney, Leqian Liu, Esther Wu, Valerie Fong, Austin Hsu, Suzanna Chak, Ram P. Naikawadi, Paul J. Wolters, Adam R. Abate, Atul J. Butte, and Mallar Bhattacharyya. 2019. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. Nature Immunology 2, 2 (2019), 163–172. https://doi.org/10.1038/s41590-018-0276-y
[11] Ricard Argelaguet, Damien Arnol, Danila Bredikhin, Yonatan Deloro, Britta Velten, John C. Marioni, and Oliver Stegle. 2020. MOFA+: A statistical framework for comprehensive integration of multi-modal single-cell data. Genome Biology 21, 1 (2020), 1–17.

[12] Michaela Asp, Joseph Bergensträde, and Joakim Lundeberg. 2020. Spatially resolved transcriptomes-next generation tools for tissue exploration. BioEssays 42, 10 (2020), 1900221.

[13] Md Badsha, Rui Li, Boxiang Liu, Yang I. Li, Min Xian, Nicholas E. Banovich, Audrey Qiuyan Fu, et al. 2020. Imputation of single-cell gene expression with an autoregressive neural network. Quantitative Biology 8, 1 (2020), 78–94.

[14] Mukesh Bansal, Vincenzo Belcastro, Alberto Ambesi-Impiombato, and Diego Di Bernardo. 2007. How to infer gene networks from expression profiles. Molecular Systems Biology 3, 1 (2007), 78.

[15] Maayan Baron and Itai Yanai. 2017. New skin for the old RNA-Seq ceremony: The age of single-cell multi-omics. Genome Biology 18, 1 (2017), 1–3.

[16] Marek Bartosovic, Mukund Kabbe, and Gonçalo Castelo-Branco. 2021. Single-cell cut&tag profiles histone modifications and transcription factors in complex tissues. Nature Biotechnology 39, 7 (2021), 825–835.

[17] Brett K. Beaulieu-Jones, Jason H. Moore, and Pooled Resource Open-Access ALS Clinical Trials Consortium. 2017. Missing data imputation in the electronic health record using deeply learned autoencoders. In Pacific Symposium on Biocomputing 2017. World Scientific. 207–218.

[18] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. 2019. MixMatch: A holistic approach to Semi-Supervised Learning. Advances in Neural Information Processing Systems (2019).

[19] Adrian Bird. 2007. Perceptions of epigenetics. Nature 447, 7143 (2007), 396.

[20] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre. 2008. Fast Unfolding of Community Hierarchies in Large Networks. Code Link: https://github.com/taynaud/python-louvain

[21] A. Sina Boosehaghi, Ingileif B. Hallgrímsdóttir, Ángel Gálvez-Merchán, and Lior Pachter. 2022. Depth normalization for single-cell genomics count data. bioRxiv (2022).

[22] Guillem Boquet, Jose Lopez Vicario, Antoni Morell, and Javier Serrano. 2019. Missing data in traffic estimation: A variational autoencoder imputation method. In 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’19). IEEE, 2882–2886.

[23] Gerard Brady, Mary Barbara, and Norman Iscove. 1990. Representative in vitro cDNA amplification from individual hematopoietic cells and colonies. Methods in Molecular and Cellular Biology 2 (1990), 17–25.

[24] Byron F. Brehm-Stecher and Eric A. Johnson. 1990. Single-cell microbiology: Tools, technologies, and applications. Microbiology and Molecular Biology Reviews 68, 2 (1990), 538–559.

[25] Leo Breiman. 2001. Random forests. Machine Learning 45, 1 (2001), 5–32. https://doi.org/10.1023/A:1010934043242

[26] Jason D. Buenrostro, Paul G. Giresi, Lisa C. Zaba, Howard Y. Chang, and William J. Greenleaf. 2013. Transposition of native chromatin for multimodal regulatory analysis and personal epigenomics. Nature Methods 10, 12 (2013), 1213.

[27] Dylan M. Cable, Evan Murray, Luli S. Zou, Aleksandrina Goeva, Evan Z. Macosko, Fei Chen, and Rafael A. Irizarry. 2022. Robust decomposition of cell type mixtures in spatial transcriptomics. Nature Biotechnology 40, 4 (2022), 517–526. pmid:33603203

[28] Ali Can, Musodiq Bello, Harvey E. Cline, Xiaodong Tao, Paulo Mendonca, and Michael Gerdes. 2009. A unified segmentation method for detecting subcellular compartments in immunofluorescently labeled tissue images. In International Workshop on Microscopic Image Analysis with Applications in Biology.

[29] Junyue Cao, Darren A. Cusanovich, Vijay Ramani, Delasa Aghamirzaie, Hannah A. Pliner, Andrew J. Hill, Riza M. Daza, Jose L. McFaline-Figueroa, Jonathan S Packer, Lena Christiansen, et al. 2018. Joint profiling of chromatin accessibility and gene expression in thousands of single cells. Science 361, 6409 (2018), 1380–1385. Dataset Link: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE117089

[30] Junyue Cao, Jonathan S. Packer, Vijay Ramani, Darren A. Cusanovich, Chau Huynh, Riza Daza, Xiaojie Qiu, Choli Lee, Scott N. Furlan, Frank J. Steemers, et al. 2017. Comprehensive single-cell transcriptional profiling of a multicellular organism. Science 357, 6352 (2017), 661–667. Data Link: http://atlas.gs.washington.edu/worm-rna/docs/

[31] Junyue Cao, Jonathan S. Packer, Vijay Ramani, Darren A. Cusanovich, Chau Huynh, Riza Daza, Xiaojie Qiu, Choli Lee, Scott N. Furlan, Frank J. Steemers, Andrew Adey, Robert H. Waterston, Cole Trapnell, and Jay Shendure. 2017. Comprehensive single-cell transcriptional profiling of a multicellular organism. Science 357, 6352 (Aug. 2017), 661–667. https://doi.org/10.1126/science.aam8940

[32] Kai Cao, Xinxiang Bai, Yaguang Hong, and Lin Wan. 2020. Unsupervised topological alignment for single-cell multi-omics integration. Bioinformatics 36, Supplement_1 (2020), i48–i56.

[33] Kai Cao, Yiguang Hong, and Lin Wan. 2022. Manifold alignment for heterogeneous single-cell multi-omics data integration using Pamona. Bioinformatics 38, 1 (2022), 211–219.

[34] Yinghao Cao, Xiaoyue Wang, and Gongxin Peng. 2020. SCSA: A cell type annotation tool for single-cell RNA-seq data. Frontiers in Genetics 11 (2020).
Deep Learning in Single-cell Analysis

[55] Tabula Muris Consortium et al. 2018. Single-cell transcriptomics of 20 mouse organs creates a tabula muris. Nature 562, 7727 (2018), 367–372.

[56] Tabula Muris Consortium et al. 2018. Single-cell transcriptomics of 20 mouse organs creates a tabula muris. Nature 562, 7727 (2018), 367–372. Data Link: https://tabula-muris.ds.czbiohub.org/

[57] Albert H. Coons, Hugh J. Creech, R. Norman Jones, and Ernst Berliner. 1942. The demonstration of pneumococcal antigen in tissues by the use of fluorescent antibody. Journal of Immunology 45, 3 (1942), 159–170.

[58] Nicola Crosetto, Magda Bienko, and Alexander Van Oudenaarden. 2015. Spatially resolved transcriptomics and beyond. Nature Reviews Genetics 16, 1 (2015), 57–66.

[59] Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, and Bo Wang. 2023. scgpt: Towards building a foundation model for single-cell multi-omics using generative ai. bioRxiv (2023), 2023–04.

[60] Ivan Culjak, David Abram, Tomislav Pribanic, Hrvoje Dzapo, and Mario Cifrek. 2012. A brief introduction to OpenCV. In 2012 Proceedings of the 35th International Convention MIPRO. IEEE, 1725–1730.

[61] Justin Hong Dan D. Erdmann-Pham, Jonathan Fischer and Yun S. Song. 2021. A likelihood-based deconvolution of bulk gene expression data using single-cell references. Genome Research (July 2021).https://doi.org/10.1101/gr.272344.120 Code Link: https://github.com/songlab-cal/rna-sieve

[62] Patrick Danaher, Youngmi Kim, Brenn Nelson, Maddy Griswold, Zhi Yang, Erin Piazza, and Joseph M. Beecham. 2022. Advances in mixed cell deconvolution enable quantification of cell types in spatial transcriptomic data. Nature Communications 13, 1 (Jan. 2022).https://doi.org/10.1038/s41467-022-28020-5Code Link: https://github.com/Nanostring-Biostats/SpatialDecon

[63] Jurrian K. de Kanter, Philip Lijnzaad, Tito Candelli, Thanasis Margaritis, and Frank C. P. Holstege. 2019. CHETAH: A selective, hierarchical cell type identification method for single-cell RNA sequencing. Nucleic Acids Research 47, 16 (2019), e95–e95. https://doi.org/10.1093/nar/gkz543

[64] Yue Deng, Feng Bao, Qionghai Dai, Lani F. Wu, and Steven J. Altschuler. 2018. Massive single-cell RNA-seq analysis and imputation via deep learning. bioRxiv (2018), 315556.

[65] Yue Deng, Feng Bao, Qionghai Dai, Lani F. Wu, and Steven J. Altschuler. 2019. Scalable analysis of cell-type specific features in single-cell transcriptomic data using deep recurrent learning. Nature Methods 16, 4 (2019), 311–314.

[66] Jiayuan Ding, Hongzhi Wen, Wenzhuo Tang, Renming Liu, Zhaoheng Li, Julian Venegas, Runze Su, Dylan Molho, Wei Jin, Wangyang Zuo, et al. 2022. DANCE: A deep learning library and benchmark for single-cell analysis. bioRxiv (2022). Code Link: https://github.com/OmicsML/dance

[67] C. Dominguez Conde, C. Xu, L. B. Jarvis, D. B. Rainbow, S. B. Wells, T. Gomes, S. K. Howlett, O. Suchanek, K. Polanski, H. W. King, L. Mamanova, N. Huang, P. A. Szabo, L. Richardson, L. Bolt, E. S. Fasouli, K. T. Mabhubani, M. Prete, L. Tuck, N. Richoz, Z. K. Tuong, L. Campos, H. S. Moussa, E. J. Needham, S. Pritchard, T. Li, R. Elmentaite, J. Park, E. Rahman, D. Chen, D. K. Menon, O. A. Bayraktar, L. K. James, K. B. Meyer, N. Yosef, M. R. Clatworthy, P. A. Sims, D. L. Farber, K. Saeb-Parsy, J. L. Jones, and S. A. Teichmann. 2022. Cross-tissue immune cell analysis reveals tissue-specific features in humans. Science 376, 6594 (2022), eabl5197. https://www.science.org/doi/10.1126/science.abl5197

[68] Kangning Dong and Shihua Zhang. 2022. Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder. Nature Communications 13, 1 (2022), 1–12. Code Link: https://github.com/zhanglabtools/STAGATE.

[69] R. Dong and G. Yuan. 2021. SpatialDWLS: Accurate deconvolution of spatial transcriptomic data. Genome Biology 22, 1 (May 2021). https://doi.org/10.1186/s13059-021-02362-7 Code Link: https://github.com/RubD/Giotto

[70] Shi Dong, Ping Wang, and Khushnood Abbas. 2021. A survey on deep learning and its applications. Computer Science Review 46 (2021), 100379.

[71] Jinzhua Dou, Shaoheng Liang, Vakul Mohanty, Xuesen Cheng, Sangbae Kim, Jongsu Choi, Yumei Li, Katayoun Rezvani, Rui Chen, and Ken Chen. 2020. Unbiased integration of single cell multi-omics data. bioRxiv (2020).

[72] Ruben Dries, Qian Zhu, Rui Dong, Chee-Huat Linus Eng, Huipeng Li, Kan Liu, Yuntian Fu, Tianxiao Zhao, Arpan Sarkar, Feng Bao, et al. 2021. Giotto: A toolbox for integrative analysis and visualization of spatial expression data. Genome Biology 22, 1 (2021), 1–31. Code Link: https://github.com/RubD/Giotto.

[73] Zhana Duren, Xi Chen, Rui Jiang, Yong Wang, and Wing Hung Wong. 2017. Modeling gene regulation from paired expression and chromatin accessibility data. Proceedings of the National Academy of Sciences 114, 25 (2017), E4914–E4923.

[74] Zhana Duren, Xi Chen, Mahdi Zamanighomi, Wanwen Zeng, Ansuman T. Satpathy, Howard Y. Chang, Yong Wang, and Wing Hung Wong. 2018. Integrative analysis of single-cell genomics data by coupled nonnegative matrix factorizations. Proceedings of the National Academy of Sciences 115, 30 (2018), 7723–7728.

[75] James Eberwine, Hermes Yeh, Kevin Miyashiro, Yanxiang Cao, Suresh Nair, Richard Finnell, Martha Zettel, and Paul Coleman. 1992. Analysis of gene expression in single live neurons. Proceedings of the National Academy of the United States of America 89 (1992), 3010–3014.
Deep Learning in Single-cell Analysis

Knauumalle, Kevin Kenny, Melinda Larsen, Michael Lazare, Qing Li, Christina Lowes, Colin C. McCulloch, Elizabeth McDonough, Michael C. Montalto, Zhengyu Fang, Jens Rittscher, Alberto Santamaria-Pang, Brion D. Sarachan, Maximilian L. Seel, Antti Seppo, Kashan Shaikh, Yunxia Sui, Jingyu Zhang, and Fiona Ginty. 2013. Highly multiplexed single-cell analysis of formalin-fixed, paraffin-embedded cancer tissue. *Proceedings of the National Academy of Sciences* 110, 29 (2013), 11982–11987. https://doi.org/10.1073/pnas.1300136110

[95] Charlotte Giesen, Hao A. O. Wang, Denis Schapiro, Nevena Zivanovic, Andrea Jacobs, Bodo Hattendorf, Peter J. Schüßler, Daniel Grolimund, Joachim M. Buhmann, Simone Brandt, et al. 2014. Highly multiplexed imaging of tumor tissues with subcellular resolution by mass cytometry. *Nature Methods* 11, 4 (2014), 417–422.

[96] Amir Giladi and Ido Amit. 2018. Single-cell genomics: A stepping stone for future immunology discoveries. *Cell* 172, 1–2 (2018), 14–21.

[97] Samantha L. Goldman, Matthew MacKay, Ebrahim Afshinnekoo, Ari M. Melnick, Shuxiu Wu, and Christopher E. Mason. 2019. The impact of heterogeneity on single-cell sequencing. *Frontiers in Genetics* 10 (March 2019), 8.

[98] Yury Goltsev, Nikolay Samusik, Julia Kennedy-Darling, Salil Bhide, Matthew Hale, Gustavo Vazquez, Sarah Black, and Garry P. Nolan. 2018. Deep profiling of mouse splenic architecture with CODEX multiplexed imaging. *Cell* 174, 4 (2018), 968–981.

[99] Lovedeep Gondara and Ke Wang. 2017. Multiple imputation using deep denoising autoencoders. *arXiv preprint arXiv:1705.02737* 280 (2017).

[100] Lovedeep Gondara and Ke Wang. 2018. Mida: Multiple imputation using denoising autoencoders. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 260–272.

[101] Boying Gong, Yun Zhou, and Elizabeth Purdom. 2021. Cobolt: Integrative analysis of multimodal single-cell sequencing data. *Genome Biology* 22, 1 (2021), 1–21. Code Link: https://github.com/epurdom/cobolt_manuscript

[102] Jing Song, Minsheng Hao, Xin Zeng, Chiming Liu, Jianzhu Ma, Xingyi Cheng, Taifeng Wang, Xuegong Zhang, and Le Song. 2023. xTrimoGene: An efficient and scalable representation learner for single-cell RNA-seq data. *bioRxiv* (2023), 2023–03.

[103] Wuming Gong, Il-Youp Kwak, Pruthvi Kota, Naoko Koyano-Nakagawa, and Daniel J. Garry. 2018. DrImpute: Imputing dropout events in single cell RNA sequencing data. *BMC Bioinformatics* 19, 1 (2018), 1–10.

[104] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in Neural Information Processing Systems* 27 (2014).

[105] Noah Greenwald, Geneva Miller, Erick Moen, Alex Kong, Adam Kagel, Thomas Dougherty, Christine Fullaway, Brianna McIntosh, Ke Leow, Morgan Schwartz, Cole Pavelchek, Sunny Cui, Isabella Campilloss, Omer Bar-Tal, Jaiveer Singh, Mara Fong, Gautam Chaudhry, Zion Abraham, Jackson Moseley, and David Valen. 2022. Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nature Biotechnology* 40 (April 2022). https://doi.org/10.1038/s41587-021-01094-0 Code Link: https://github.com/voelkel/deepcell-tf

[106] Hongyu Guo and Jun Li. 2021. scSorter: Assigning cells to known cell types according to marker genes. *Genome Biology* 22, 1 (2021), 69. https://doi.org/10.1186/s13059-021-02281-7

[107] Hongshun Guo, Ping Zhu, Xinglong Wu, Xianlong Li, Lu Wen, and Fuchou Tang. 2013. Single-cell methylome landscapes of mouse embryonic stem cells and early embryos analyzed using reduced representation bisulfite sequencing. *Genome Research* 23, 12 (Oct. 2013), 2126–2135. https://doi.org/10.1101/gr.161679.113

[108] Minzhe Guo, Hui Wang, S. Steven Potter, Jeffrey A. Whitsett, and Yan Xu. 2015. SINCERA: A pipeline for single-cell RNA-Seq profiling analysis. *PLoS Computational Biology* 11, 11 (2015), e1004575.

[109] Aric Hagberg, Pieter Swart, and Daniel S. Chult. 2008. *Exploring Network Structure, Dynamics, and Function Using Networkx*. Technical Report. Los Alamos National Lab (LANL), Los Alamos, NM, USA.

[110] Laleh Haghverdi, Aaron T. L. Lun, Michael D. Morgan, and John C. Marioni. 2018. Batch effects in single-cell RNA-Seq profiling analysis. *Technical Report*. Los Alamos National Lab (LANL), Los Alamos, NM, USA.

[111] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems* (2017).https://arxiv.org/abs/1706.02216

[112] Xiaoping Han, Renying Wang, Yuming Zhou, Lijiang Fei, Huiyu Sun, Shujing Lai, Assieh Saadatpour, Ziming Zhou, Haide Chen, Song Ye, et al. 2018. Mapping the mouse cell atlas by microwell-seq. *Cell* 172, 5 (2018), 1091–1107. Data Link: https://figshare.com/s/865e6594d0d5857db4

[113] Xiaoping Han, Renying Wang, Yiming Zhou, Lijiang Fei, Huiyu Sun, Shujing Lai, Assieh Saadatpour, Ziming Zhou, Haide Chen, Song Ye, Daosheng Huang, Yang Xu, Wentao Huang, Mengmeng Jiang, Xinyi Jiang, Jie Mao, Yao Chen, Chenyu Lu, Jin Xie, Qun Fang, Yibin Wang, Rui Yue, Tiefeng Li, He Huang, Stuart H. Orkin, Guo-Cheng Yuan, Ming Chen, and Guoji Guo. 2018. Mapping the mouse cell atlas by microwell-seq. *Cell* 172, 5 (Feb. 2018), 1091–1107.e17. https://doi.org/10.1016/j.cell.2018.02.001
[114] Xiaoping Han, Ziming Zhou, Lijiang Fei, Huiyu Sun, Renying Wang, Yao Chen, Haide Chen, Jingjing Wang, Huanna Tang, Wenhao Ge, Yingcong Zhou, Fang Ye, Mengmeng Jiang, Junqing Wu, Yanyu Xiao, Xiaoning Jia, Tingyue Zhang, Xiaojie Ma, Qi Zhang, Xueli Bai, Shujing Lai, Chengxuan Yu, Lijun Zhu, Rui Lin, Yuchi Gao, Min Wang, Yiqing Wu, Jianming Zhang, Renya Zhan, Saiyong Zhu, Haiyan Hu, Changchun Wang, Ming Chen, He Huang, Tingbo Liang, Jianghua Chen, Weilin Wang, Dan Zhang, and Guoji Guo. 2020. Construction of a human cell landscape at single-cell level. *Nature* 581, 7808 (May 2020), 303–309. https://doi.org/10.1038/s41586-020-2157-4

[115] Yuhua Hao, Stephanie Hao, Erica Andersen-Nissen, William M. Mauck III, Shiewei Zheng, Andrew Butler, Maddie J. Lee, Aaron J. Wilk, Charlotte Darby, Michael Zager, et al. 2021. Integrated analysis of multimodal single-cell data. *Cell* 184, 13 (2021), 3573–3587.

[116] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*. 2961–2969.

[117] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[118] Nicholas Heller, Fabian Isensee, Klaus H. Maier-Hein, Xiaoshuai Hou, Chunmei Xie, Fengyi Li, Yang Nan, Guangrui Mu, Zhiyong Lin, Miofei Han, et al. 2021. The state of the art in kidney and kidney tumor segmentation in contrast-enhanced CT imaging: Results of the KITS19 challenge. *Medical Image Analysis* 67 (2021), 101821.

[119] Mikael Henaff, Joan Bruna, and Yann LeCun. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163* (2015).

[120] David Hendrickson, Ilya Sosifer, Bernd Wranik, David Botstein, and Scott McIsaac. 2018. Simultaneous profiling of DNA accessibility and gene expression dynamics with ATAC-seq and RNA-seq. *Methods in Molecular Biology* 1819 (2018), 317–333.

[121] Leon Hetzel, Simon Boehm, Niki Kilbertus, Stephan Günemann, Fabian Theis, et al. 2022. Predicting cellular responses to novel drug perturbations at a single-cell resolution. *Advances in Neural Information Processing Systems* 35 (2022), 26711–26722.

[122] Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural networks and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems* 6, 2 (1998), 107–116.

[123] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[124] Rui Hou, Elena Denisenko, and Alistair R. R. Forrest. 2019. scMatch: A single-cell gene expression profile annotation tool using reference datasets. *Bioinformatics* 35, 22 (2019), 4688–4695. https://doi.org/10.1093/bioinformatics/btz292

[125] Yu Hou, Huahu Guo, Chen Cao, Xianlong Li, Boqiang Hu, Ping Zhu, Xinglong Wu, Lu Wen, Fuchou Tang, Yanyi Huang, et al. 2016. Single-cell triple omics sequencing reveals genetic, epigenetic, and transcriptomic heterogeneity in hepatocellular carcinomas. *Cell Research* 26, 3 (2016), 304–319.

[126] David Houle, Diddahally R. Govindaraju, and Stig Omholt. 2010. Phenomics: The next challenge. *Nature Reviews Genetics* 11, 12 (Nov. 2010), 855–866. https://doi.org/10.1038/nrg2897

[127] Jian Hu, Xiangjie Li, Kyle Coleman, Amelia Schroeder, Nan Ma, David J. Lee, Russell T. Shinozaka, and Mingyao Li. 2021. SpaGCN: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nature Methods* 18, 11 (2021), 1342–1351. Code Link: https://github.com/jianhuupenn/SpaGCN

[128] Jinghan Huang, Yiqing Shen, Dinggang Shen, and Jing Ke. 2021. CA 2.5-Net nuclei segmentation framework with assisted Intervention. *Springer*, 445–454.

[129] Mo Huang, Jingdhu Wang, Eduardo Torre, Hannah Dueck, Sydney Shaffer, Roberto Bonasio, John I. Murray, Arjun Raj, Mingyao Li, and Nancy R. Zhang. 2018. SAVER: Gene expression recovery for single-cell RNA sequencing. *Nature Methods* 15, 7 (2018), 539–542.

[130] Qianhui Huang, Yu Liu, Yuheng Du, and Lana X. Garmire. 2021. Evaluation of cell type annotation R packages on single-cell RNA-seq data. *Genomics, Proteomics & Bioinformatics* 19, 2 (April 2021), 267–281. https://doi.org/10.1016/j.gpb.2020.07.004

[131] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. 2018. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Experimental & Molecular Medicine* 50, 8 (2018), 1–14.

[132] Dong hyun Lee. 2013. Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks.

[133] Saiful Islam, Una Kjällquist, Annalena Moliner, Pawel Zajac, Jian-Bing Fan, Peter Lönnberg, and Sten Linnarsson. 2011. Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq. *Genome Research* 21, 7 (2011), 1160–1167.

ACM Trans. Intell. Syst. Technol., Vol. 15, No. 3, Article 40. Publication date: March 2024.
[134] Mika Sarkin Jain, Krzysztof Polanski, Cecilia Dominguez Conde, Xi Chen, Jongeun Park, Lira Mamanova, Andrew Knights, Rachel A. Botting, Emily Stephenson, Mazilah Haniffa, et al. 2021. MultiMAP: Dimensionality reduction and integration of multimodal data. Genome Biology 22, 1 (2021), 1–26.

[135] Ayush Jaiswal, Dong Guo, Cauigi S. Raghavendra, and Paul Thompson. 2018. Large-scale unsupervised deep representation learning for brain structure. arXiv preprint arXiv:1805.01949 (2018).

[136] Camden Jansen, Ricardo N. Ramirez, Nicole C. El-All, David Gomez-Cabrero, Jesper Tegner, Matthias Merkenschlager, Ana Comesa, and Ali Mortazavi. 2019. Building gene regulatory networks from scATAC-seq and scRNA-seq using linked self organizing maps. PLoS Computational Biology 15, 11 (2019), e1006555. Code Link: https://github.com/esijsen/SOMatic

[137] Raymond Austin Jarvis and Edward A. Patrick. 1973. Clustering using a similarity measure based on shared near neighbors. IEEE Transactions on Pattern Analysis and Machine Intelligence 1, 7 (2014), 741–742.

[138] Hao Jiang, Lydia L. Sohn, Haiyan Huang, and Luonan Chen. 2018. Single cell clustering based on cell-pair differentiability correlation and variance analysis. Bioinformatics 34, 21 (2018), 3684–3694.

[139] Suoqin Jin, Luhua Zhang, and Qing Nie. 2020. scAI: An unsupervised approach for the integrative analysis of parallel single-cell transcriptomic and epigenomic profiles. Genome Biology 21, 1 (2020), 1–19.

[140] Stephen C. Johnson. 1967. Hierarchical clustering schemes. Psychometrika 32, 3 (1967), 241–254.

[141] Lee Kamentsky, Thouis R. Jones, Adam Fraser, Mark-Anthony Bray, David J. Logan, Katherine L. Madden, Vebjorn Ljosa, Curtis Ruiden, Kevin W. Eliceiri, and Anne E. Carpenter. 2011. Improved structure, function and compatibility for CellProfiler: modular high-throughput image analysis software. Bioinformatics 27, 8 (Feb. 2011), 1179–1180. https://doi.org/10.1093/bioinformatics/btr095 arXiv:https://academic.oup.com/bioinformatics/article-pdf/27/8/1179/17103761/btr095.pdf Code Link: https://github.com/CellProfiler/CellProfiler

[142] Konstantinos Kamnitas, Christian Ledig, Virginia F. J. Newcombe, Joanna P. Simpson, Andrew D. Kane, David K. Menon, Daniel Rueckert, and Ben Glover. 2017. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Medical Image Analysis 36 (2017), 61–78.

[143] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Kiat Hui Agam, David M. Mount, Nathan S. Netanyahu, Kiat Hui Agam, Christine D. Piatko, Ruth Silverman, and Ann Y. Wu. 2002. An efficient k-means clustering algorithm: Algorithm and Implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 7 (2002), 881–892. Code Link: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

[144] Chantriolnt-Andreas Kaporouani and Guido Sanguinetti. 2019. Melissa: Bayesian clustering and imputation of single-cell methylomes. Genome Biology 20, 1 (2019), 1–15.

[145] Jakob Nikolos Kather, Johannes Krisam, Pornpimol Charoenpong, Tom Luedde, Esther Herpel, Cleo-Aron Weis, Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. 2002. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 7 (2002), 881–892. Code Link: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

[146] Peter V. Kharchenko, Lev Silberstein, and David T. Scadden. 2014. Bayesian approach to single-cell differential expression analysis. Nature Methods 11, 7 (2014), 740–742.

[147] Jacob C. Kimmeland David R. Kelley. 2021. Semisupervised adversarial neural networks for single-cell classification. Genome Biology 20, 1 (2019), 1–15.

[148] Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114 (2013).

[149] Diederik P. Kingma, Max Welling, et al. 2019. An introduction to variational autoencoders. Foundations and Trends® in Machine Learning 12, 4 (2019), 307–392.

[150] Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02706 (2016).

[151] Vladimir Yu Kiselev, Kristina Kirschn, Michael T. Schaurb, Tallullah Andrews, Andrew Yiuy, Tamir Chandra, Kedar N. Natarajan, Wolf Reik, Mauricio Barahona, Anthony R. Green, et al. 2017. SC3: Consensus clustering of single-cell RNA-seq data. Nature Methods 14, 5 (2017), 483–486. Code Link: https://github.com/hemberg-lab/sc3

[152] Vladimir Yu Kiselev, Andrew Yiuy, and Martin Hemberg. 2018. Scmap: Projection of single-cell RNA-seq data across data sets. Nature Methods 15, 5 (2018), 359–362. https://doi.org/10.1038/nmeth.4644

[153] Hiroaki Kitano. 2002. Systems biology: A brief overview. Science 295, 5560 (2002), 1662–1664.

[154] Allon M. Klein, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A. Weitz, and Marc W. Kirschner. 2015. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. Cell 161, 5 (2015), 1187–1201. Data Link: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE65525

ACM Trans. Intell. Syst. Technol., Vol. 15, No. 3, Article 40. Publication date: March 2024.
[157] Vitalii Kleshchevnikov, Artem Shmatko, Emma Dann, Alexander Aivazidis, Hamish W. King, Tong Li, Rasa Elmentaita, Artem Lomakin, Veronika Kedlian, Adam Gayoso, Mika Sarkin Jain, Jun Sung Park, Lauma Ramona, Elizabeth Tuck, Anna Arutyunyan, Roser Vento-Tormo, Mortiz Gerstung, Louisa James, Oliver Stegle, and Omer Ali Bayraktar. 2022. Cell2location maps fine-grained cell types in spatial transcriptomics. Nature Biotechnology 40, 5 (2022), 661–671. Code Link: https://github.com/BayraktarLab/cell2location

[158] Teuvo Kohonen. 1982. Self-organized formation of topologically correct feature maps. Biological Cybernetics 43, 1 (1982), 59–69.

[159] Aleksandra A. Kolodziejczyk, Jong Kyoun Kim, Valentine Svensson, John C. Marioni, and Sarah A. Teichmann. 2015. The technology and biology of single-cell RNA sequencing. Molecular Cell 58, 4 (2015), 610–620.

[160] Roger D. Kornberg. 1974. Chromatin structure: A repeating unit of histones and DNA. Science 184, 4139 (1974), 868–871.

[161] Roger D. Kornberg and Yahli Lorch. 1974. Chromatin structure and transcription. Annual Review of Cell Biology 8 (1974), 563–587.

[162] Can Fahrettin Koyuncu, Gozde Nur Gunesli, Rengul Cetin-Atalay, and Cigdem Gunduz-Demir. 2020. DeepDistance: A multi-task deep regression model for cell detection in inverted microscopy images. Medical Image Analysis 63 (2020), 101720. https://doi.org/10.1016/j.media.2020.101720

[163] April R. Kriebel and Joshua D. Welch. 2022. UINMF performs mosaic integration of single-cell multi-omic datasets using nonnegative matrix factorization. Nature Communications 13, 1 (2022), 1–17.

[164] Daniëlle Krijgsman, Neeraj Sinha, Matthijs J. D. Baars, Stephanie van Dam, Mojtaba Amini, and Yvonne Vercoulen. 2022. MATISSE: An analysis protocol for combining imaging mass cytometry with fluorescence microscopy to generate single-cell data. STAR Protocols 3, 1 (2022), 101034. https://doi.org/10.1016/j.xpro.2021.101034

[165] Ashwinikumar Kulkarni, Ashley G. Anderson, Devin P. Merullo, and Genevieve Konopka. 2019. Beyond bulk: A review of single cell transcriptomics methodologies and applications. Current Opinion in Biotechnology 58 (Aug. 2019), 129–136.

[166] Mahmoud Labib and Shana O. Kelley. 2020. Single-cell analysis targeting the proteome. Nature Reviews Chemistry 4, 3 (2020), 143–158.

[167] David Lähnemann et al. 2020. Eleven grand challenges in single-cell data science. Genome Biology 21, 1 (2020).

[168] Blue B. Lake, Song Chen, Brandon C. SOS, Jean Fan, Gwendolyn E. Kaeser, Yun C. Yung, Thu E. Duong, Derek Gao, Jerold Chun, Peter V. Kharchenko, et al. 2018. Integrative single-cell analysis of transcriptional and epigenetic states in the human adult brain. Nature Biotechnology 36, 1 (2018), 70–80.

[169] Yann LeCun, Yoshua Bengio, et al. 1995. Convolutional networks for images, speech, and time series. Handbook of Brain Theory and Neural Networks 3361, 10 (1995), 1995.

[170] David Legland, Ignacio Arganda-Carreras, and Philippe Andrey. 2016. MorphoLibJ: Integrated library and plugins for mathematical morphology with ImageJ. Bioinformatics 32, 22 (2016), 3532–3534. Code Link: https://github.com/ijpb/MorphoLibJ

[171] Zachary R. Lewis, Tien Phan-Everson, Gary Geiss, Mithra Korukonda, Ruchir Bhatt, Carl Brown, Dwayne Dunaway, Joseph Phan, Alyssa Rosenbloom, Brian Filanoski, et al. 2022. Subcellular characterization of over 100 proteins in FFPE tumor biopsies with CosMx spatial molecular imager. Cancer Research 82, 12_Supplement (2022), 3878–3878.

[172] Chenwei Li, Baolin Liu, Boxi Kang, Zedao Liu, Yedan Liu, Changya Chen, Xianwen Ren, and Zemin Zhang. 2020. SciBet as a portable and fast single cell type identifier. Nature Communications 11, 1 (2020), 1818. https://doi.org/10.1038/s41467-020-15523-2

[173] Haochen Li, Tianxing Ma, Minsheng Hao, Lei Wei, and Xuegong Zhang. 2022. Decoding functional cell-cell communication events by multi-view graph learning on spatial transcriptomics. bioRxiv (2022).

[174] Jiachen Li, Siheng Chen, Xiaoyong Pan, Ye Yuan, and Hong-Bin Shen. 2022. Cell clustering for spatial transcriptomics data with graph neural networks. Nature Computational Science 2, 6 (2022), 399–408. Code Link: https://github.com/xiaoyeye/CCST

[175] Wei Vivian Li and Jingyi Jessica Li. 2018. An accurate and robust imputation method scImpute for single-cell RNA-seq data. Nature Communications 9, 1 (2018), 1–9.

[176] Xiaomeng Li, Hao Chen, Xiaojuan Qi, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. 2018. H-DenseUNet: Hybrid densely connected UNet for liver and tumor segmentation from CT volumes. IEEE Transactions on Medical Imaging 37, 12 (2018), 2663–2674.

[177] Xinmin Li and Cun-Yu Wang. 2021. From bulk, single-cell to spatial RNA sequencing. International Journal of Oral Science 13, 1 (2021), 1–6.

[178] Xiangjie Li, Kui Wang, Yafei Lyu, Huize Pan, Jingxiao Zhang, Dwight Stambolian, Katalin Susztak, Muredadh P. Reilly, Gang Hu, and Mingyao Li. 2020. Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis. Nature Communications 11, 1 (2020), 1–14. Code Link: https://eleozzr.github.io/desc/
Deep Learning in Single-cell Analysis

[179] Zhijian Li, Christoph Kuppe, Susanne Ziegler, Mingbo Cheng, Nazanin Khabgani, Sylvia Menzel, Martin Zenke, Rafael Kramann, and Ivan G. Costa. 2021. Chromatin-accessibility estimation from single-cell ATAC-seq data with scOpen. Nature Communications 12, 1 (2021), 1–14.

[180] Jialong Liang, Wanshi Cai, and Zhongsheng Sun. 2014. Single-cell sequencing technologies: Current and future. Journal of Genetics and Genomics 41, 10 (2014), 513–528.

[181] Jia-Ren Lin, Mohammad Fallah-Sichani, and Peter K. Sorger. 2015. Highly multiplexed imaging of single cells using a high-throughput cyclic immunofluorescence method. Nature Communications 6, 1 (2015), 1–7.

[182] Jia-Ren Lin, Benjamin Izar, Shu Wang, Clarence Yapp, Shaolin Mei, Parin M. Shah, Sandro Santagata, and Peter K. Sorger. 2018. Highly multiplexed immunofluorescence imaging of human tissues and tumors using t-CyCIF and conventional optical microscopes. Elife 7 (2018).

[183] Peijie Lin, Michael Troup, and Joshua W. K. Ho. 2017. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. Genome Biology 18, 1 (2017), 1–11.

[184] Peijie Lin, Michael Troup, and Joshua W. K. Ho. 2017. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. Genome Biology 18, 1 (2017), 1–11. Code Link: https://github.com/VCCRI/CIDR

[185] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2117–2125.

[186] Yingxin Lin, Tung-Yu Wu, Sheng Wan, Jean Y. H. Yang, Wing H. Wong, and Y. X. Wang. 2022. scJoint integrates atlas-scale single-cell RNA-seq and ATAC-seq data with transfer learning. Nature Biotechnology 40, 5 (2022), 703–710. Code Link: https://github.com/SydneyBioX/scJoint

[187] Zhixiang Lin, Mahdi Zamanighomi, Timothy Daley, Shining Ma, and Wing Hung Wong. 2020. Model-based approach to the joint analysis of single-cell data on chromatin accessibility and gene expression. Statistical Science 35, 1 (2020), 2–13.

[188] George C. Linderman, Jun Zhao, Manolis Roulis, Piotr Bielecki, Richard A. Flavell, Boaz Nadler, and Yuval Kluger. 2022. Zero-preserving imputation of single-cell RNA-seq data. Nature Communications 13, 1 (2022), 1–11.

[189] Russell Littman, Zachary Hemminger, Robert Foreman, Douglas Arneson, Guanglin Zhang, Fernando Gómez-Pinilla, Xia Yang, and Roy Wollman. 2021. Joint cell segmentation and cell type annotation for spatial transcriptomics. Molecular Systems Biology 17, 6 (2021), e10108. Code Link: https://github.com/WollmanLab/JSTA

[190] Jialin Liu, Chao Gao, Joshua Sodicoff, Velina Kozareva, Evan Z. Macrosko, and Joshua D. Welch. 2020. Jointly defining cell types from multiple single-cell datasets using LIGER. Nature Protocols 15, 11 (2020), 3632–3662.

[191] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 3431–3440.

[192] Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. 2018. Deep generative modeling for single-cell transcriptomics. Nature Methods 15, 12 (2018), 1053–1058. Code Link: https://github.com/scverse/scvi-tools

[193] Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L. Ibarra, Sanjay R. Srivatsan, Mogens Naghipourfar, Riza M. Daza, Beth Martin, et al. 2023. Predicting cellular responses to complex perturbations in high-throughput screens. Molecular Systems Biology (2023), e11517.

[194] Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. 2019. scGen predicts single-cell perturbation responses. Nature Methods 16, 8 (2019), 715–721.

[195] Zhi Lu, Gustavo Carneiro, Andrew P. Bradley, Daniela Ushizima, Masoud S. Nosrati, Andrea G. C. Bianchi, Claudia M. Carneiro, and Ghassan Hamarneh. 2016. Evaluation of three algorithms for the segmentation of overlapping cervical cells. IEEE Journal of Biomedical and Health Informatics 21, 2 (2016), 441–450.

[196] Eric Lubeck, Ahmet F. Coskun, Timur Zhiyentayev, Mubhij Ahmad, and Long Cai. 2014. Single-cell in situ RNA profiling by sequential hybridization. Nature Methods 11, 4 (March 2014), 360–361. https://doi.org/10.1038/nmeth.2892

[197] Feiyang Ma and Matteo Pellegrini. 2019. ACTINN: Automated identification of cell types in single cell RNA sequencing. Bioinformatics (2019), btz592. https://doi.org/10.1093/bioinformatics/btz592

[200] Jun Ma, Yixin Wang, Xingle An, Cheng Ge, Ziqi Yu, Jianan Chen, Qingjie Zhu, Guoqiang Dong, Jian He, Zhiqiang He, et al. 2021. Toward data-efficient learning: A benchmark for COVID-19 CT lung and infection segmentation. Medical Physics 48, 3 (2021), 1197–1210.
[201] Qin Ma and Dong Xu. 2022. Deep learning shapes single-cell data analysis. *Nature Reviews Molecular Cellular Biology* 23, 5 (2022), 303–304.

[202] Sai Ma, Bing Zhang, Lindsay M. LaFave, Andrew S. Earl, Zachary Chiang, Yan Hu, Jiarui Ding, Alison Brack, Vinay K. Kartha, Tristan Tay, et al. 2020. Chromatin potential identified by shared single-cell profiling of RNA and chromatin. *Cell* 183, 4 (2020), 1103–1116. Dataset Link: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE140205

[203] Iain C. Macaulay, Wilfried Haerty, Farveen Kumar, Yang I. Li, Tim Xiaoming Hu, Mabel J. Teng, Mubeen Goolam, Nathalie Saurat, Paul Coupland, Lesley M. Shirley, Miriam Smith, Niels Van der Aa, Ruby Banerjee, Peter D. Ellis, Michael A. Quail, Harold P. Swerdlov, Magdalena Zernicka-Goetz, Frederick J Livesey, Chris P. Ponting, and Thierry Voel. 2015. G&T-seq: Parallel sequencing of single-cell genomes and transcriptomes. *Nature Methods* 12, 6 (April 2015), 519–522. doi.org/10.1038/nmeth.3370

[204] Yao Ma and Jiliang Tang. 2021. Deep Learning on Graphs. Cambridge University Press.

[205] Y. Ma and X. Zhou. 2022. Spatially informed cell-type deconvolution for spatial transcriptomics. *Nature Biotechnology* (May 2022). https://doi.org/10.1038/s41587-020-0472-9

[206] Iain C. Macaulay, Wilfried Haerty, Farveen Kumar, Yang I. Li, Tim Xiaoming Hu, Mabel J. Teng, Mubeen Goolam, Nathalie Saurat, Paul Coupland, Lesley M. Shirley, Miriam Smith, Niels Van der Aa, Ruby Banerjee, Peter D. Ellis, Michael A. Quail, Harold P. Swerdlov, Magdalena Zernicka-Goetz, Frederick J Livesey, Chris P. Ponting, and Thierry Voel. 2015. G&T-seq: Parallel sequencing of single-cell genomes and transcriptomes. *Nature Methods* 12, 6 (April 2015), 519–522. doi.org/10.1038/nmeth.3370

[207] J. MacQueen. 1967. Classification and analysis of multivariate observations. In *5th Berkeley Symposium on Mathematical Statistics and Probability*. 281–297.

[208] Hassaan Majeed, Adib Keikhosravi, Mikhail E. Kandel, Tan H. Nguyen, Yuming Liu, Andre Kajdacsy-Balla, Krishnarao Tanglell, Kevin W. Eliceiri, and Gabriel Popescu. 2019. Quantitative histopathology of stained tissues using color spatial light interference microscopy (cSLIM). *Scientific Reports* 9, 1 (2019), 1–14.

[209] Xian F. Mallory, Mohammadamin Ebris, Nicholas Navin, and Luay Nakhl. 2020. Methods for copy number aberration detection from single-cell DNA-sequencing data. *Genome Biology* 21, 1 (2020), 1–22.

[210] Shunfu Mao, Yue Zhang, Georg Seelig, and Sreeram Kannan. 2022. CellMeSH: Probabilistic cell-type identification using indexed literature. *Bioinformatics* 38, 5 (2022), 1393–1402. doi.org/10.1093/bioinformatics/btabb53

[211] Vivien Marx. 2021. Method of the year: Spatially resolved transcriptomics. *Nature Methods* 18, 1 (2021), 9–14.

[212] Warren Mcculloch and Walter Pitts. 1943. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5 (1943), 127–147.

[213] Eliot T. McKinley, Justin Shao, Samuel T. Ellis, Cody N. Heiser, Joseph T. Roland, Mary C. Macedonia, Paige N. Vega, Susie Shin, Robert J. Coffey, and Ken S. Lau. 2022. MIRIAM: A machine and deep learning single-cell segmentation and quantification pipeline for multi-dimensional tissue images. *Cytometry Part A* (2022). Code Link: https://github.com/Coffey-Lab/MIRIAM.

[214] Cory Y. McLean, Dave Bristor, Michael Hiler, Shoa L. Clarke, Bruce T. Schaar, Craig B. Low, Aaron M. Wenger, and Gill Bejerman. 2010. GREAT improves functional interpretation of cis-regulatory regions. *Nature Biotechnology* 28, 5 (2010), 495–501.

[215] Joel McManus, Zhe Cheng, and Christine Vogel. 2015. Next-generation analysis of gene expression regulation–comparing the roles of synthesis and degradation. *Molecular Biosystems* 11, 10 (2015), 2680–2689.

[216] Larry R. Medsker and L. C. Jain. 2001. Recurrent neural networks. *IEEE Trans. Neural Networks* 12, 6 (2001), 2680–2689.

[217] Kevin Menden, Mohamed Marouf, Sergio Oller, Anupriya Dalmia, Daniel Sumner Magruder, Karin Kloiber, Peter Heutink, and Stefan Bonn. 2020. Deep learning-based cell composition analysis from tissue expression profiles. *Science Advances* 6, 30 (2020), eaba2619. https://doi.org/10.1126/sciadv.aba2619 Code Link: https://github.com/KevinMenden/scaden

[218] Christopher R. Merritt, Giang T. Ong, Sarah E. Church, Kristi Barker, Patrick Danaher, Gary Geiss, Margaret Hoang, Jaemyeong Jung, Yan Liang, Jill McKay-Fleisch, et al. 2020. Multiplex digital spatial profiling of proteins and RNA in fixed tissue. *Nature Biotechnology* (May 2020). https://doi.org/10.1038/s41587-020-0472-9

[219] Bettina Mieth, James R. F. Hockley, Nico Görnitz, Marina M.-C. Vidovic, Klaus-Robert Müller, Alex Gutteridge, and Christopher R. Merritt, Giang T. Ong, Sarah E. Church, Kristi Barker, Patrick Danaher, Gary Geiss, Margaret Hoang, Jaemyeong Jung, Yan Liang, Jill McKay-Fleisch, et al. 2020. Multiplex digital spatial profiling of proteins and RNA in fixed tissue. *Nature Biotechnology* (May 2020). https://doi.org/10.1038/s41587-020-0472-9

[220] Eleni P. Mimitou, Caleb A. Lareau, Kelvin Y. Chen, Andre L. Zorzetto-Fernandes, Yuhan Hao, Yusuke Takeshima, Wendy Luo, Tse-Shun Huang, Bertrand Z. Yeung, Efthymia Papalexi, et al. 2021. Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells. *Nature Biotechnology* 39, 10 (2021), 1246–1258. Dataset Link: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE156478

[221] Kodai Minoura, Ko Abe, Hyunha Nam, Hiroyoshi Nishikawa, and Teppei Shimamura. 2021. A mixture-of-experts deep generative model for integrated analysis of single-cell multiomics data. *Cell Reports Methods* 1, 5 (2021), 100071. Code Link: https://github.com/kodaim1115/scMM
[222] J. R. Moffitt and X. Zhuang. 2016. Chapter 1: RNA imaging with multiplexed error-robust fluorescence in situ hybridization (MERFISH). In Visualizing RNA Dynamics in the Cell, Grigory S. Filonov and Samie R. Jaffrey (Eds.). Methods in Enzymology, Vol. 572. Academic Press, 1–49. https://doi.org/10.1016/bs.mie.2016.03.020; ISSN: 0076-6879.

[223] Reuben Moncada et al. 2020. Integrating microarray-based spatial transcriptomics and single-cell RNA-seq reveals tissue architecture in pancreatic ductal adenocarcinomas. Nature Biotechnology 38, 3 (2020), 333–342.

[224] Aanchal Mongia, Debarka Sengupta, and Angshul Majumdar. 2019. Mcimpute: Matrix completion based imputation for single cell RNA-seq data. Frontiers in Genetics 10 (2019), 9.

[225] Andreas E. Moor and Shalev Itzkovitz. 2017. Spatial transcriptomics: Paving the way for tissue-level systems biology. Current Opinion in Biotechnology 46 (2017), 126–133.

[226] Lisa D. Moore, Thuc Le, and Guoping Fan. 2013. DNA methylation and its basic function. Neuropsychopharmacology 38, 1 (2013), 23–38.

[227] Lambda Moses and Lior Pachter. 2022. Museum of spatial transcriptomics. Nature Methods 19, 5 (May 2022), 534–546. https://doi.org/10.1038/s41592-022-01409-2

[228] Mauro J. Muraro, Gitanjali Dharmadhikari, Dominic Grün, Nathalie Groen, Tim Dielen, Erik Jansen, Leon Van Gurp, Marten A. Engelse, Francoise Carlotti, Elco Jp De Koning, et al. 2016. A single-cell transcriptome atlas of the human pancreas. Cell Systems 3, 4 (2016), 385–394. Data Link: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE85241

[229] Giulia Muzio, Leslie O’Bray, and Karsten Borgwardt. 2021. Biological network analysis with deep learning. Briefings in Bioinformatics 22, 2 (2021), 1515–1530.

[230] Kedar Nath Natarajan. 2019. Single-cell tagged reverse transcription (STRT-Seq). In Single Cell Methods. Springer, 133–153.

[231] Nicholas E. Navin. 2015. The first five years of single-cell cancer genomics and beyond. Genome Research 25, 10 (2015), 1499–1507.

[232] Quy H. Nguyen, Nicholas Pervolarakis, Kevin Nee, and Kai Kessenbrock. 2018. Experimental considerations for single-cell RNA sequencing approaches. Frontiers in Cell and Developmental Biology 6 (2018), 108. https://doi.org/10.3389/fcell.2018.00108

[233] Ozan Oktay, Enzo Ferrante, Konstantinos Kamnitas, Mattias Heinrich, Wenjia Bai, Jose Caballero, Stuart A. Cook, Antonio De Marvao, Timothy Dawes, Declan P. O’Regan, et al. 2017. Anatomically constrained neural networks (ACNNs): Application to cardiac image enhancement and segmentation. IEEE Transactions on Medical Imaging 37, 2 (2017), 384–395.

[234] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Wenjia Bai, Jose Caballero, Stuart A. Cook, Antonio De Marvao, Timothy Dawes, Declan P. O’Regan, et al. 2017. Anatomically constrained neural networks (ACNNs): Application to cardiac image enhancement and segmentation. IEEE Transactions on Medical Imaging 37, 2 (2017), 384–395.

[235] Keiron O’Shea and Ryan Nash. 2015. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.04858 (2015).

[236] Ahmad Paintdakhi, Bradley Parry, Manuel Campos, Irinov Irov, Johan Elf, Ivan Surovtsev, and Christine Jacobs-Wagner. 2016. Outfi: An integrated software package for high accuracy, high-throughput quantitative microscopy analysis. Molecular Microbiology 99, 4 (2016), 767–777. Website Link: http://www.outfi.org/

[237] Brenda Pardo, Abby Spangler, Lukas M. Weber, Stephanie C. Hicks, Andrew E. Jaffe, Keri Martinowich, Kris K. Maynard, and Leonardo Collado-Torres. 2022. spatialLIBD: An R/Bioconductor package to visualize spatially-resolved transcriptomics data. BMC Genomics (2022). https://doi.org/10.1186/s12864-022-08601-w

[238] Sapun Parekh and Schwendy Mischa. 2019. EVICAN Dataset. https://doi.org/10.17617/3.AJBV1S

[239] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In International Conference on Machine Learning. PMLR, 1310–1318.

[240] Giovanni Pasquini, Jesus Eduardo Rojo Arias, Patrick Schäfer, and Volker Busskamp. 2021. Automated methods for cell type annotation on scRNA-seq data. Computational and Structural Biotechnology Journal 19 (2021), 961–969. https://doi.org/10.1016/j.csbj.2021.01.015

[241] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems 32 (2019).

[242] Scott Patterson and Ruedi Aebersold. 2003. Proteomics: The first decade and beyond. Nature Genetics 33 (2003).

[243] Ricardo Cardoso Pereira, Miriam Seoane Santos, Pedro Pereira Rodrigues, and Pedro Henriques Abreu. 2020. Reviewing autoencoders for missing data imputation: Technical trends, applications and outcomes. Journal of Artificial Intelligence Research 69 (2020), 1255–1285.

[244] Vanessa M. Peterson, Kelvin Xi Zhang, Namit Kumar, Jerelyn Wong, Lixia Li, Douglas C. Wilson, Renee Moore, Terrill K. McLanahan, Svetlana Sadekova, and Joel A. Klappenbach. 2017. Multiplexed quantification of proteins and transcripts in single cells. Nature Biotechnology 35, 10 (2017), 936–939.

ACM Trans. Intell. Syst. Technol., Vol. 15, No. 3, Article 40. Publication date: March 2024.
[245] Viktor Petukhov, Rosalind Xu, Ruslan Soldatov, Paolo Cadinu, Konstantin Khodosevich, Jeffrey Moffitt, and Peter Kharchenko. 2021. Cell segmentation in imaging-based spatial transcriptomics. Nature Biotechnology 40 (Oct. 2021), 1–10. https://doi.org/10.1038/s41587-021-01044-w Code Link: https://github.com/kharchenkolab/Baysor

[246] Duy Pham, Xiao Tan, Jun Xu, Laura F. Grice, Pui Yeng Lam, Arti Raghubar, Jana Vukovic, Marc J. Ruitenbeek, and Quan Nguyen. 2020. stLearn: Integrating spatial location, tissue morphology and gene expression to find cell types, cell-cell interactions and spatial trajectories within undissociated tissues. bioRxiv (2020). Code Link: https://github.com/BiomedicalMachineLearning/stLearn

[247] Hannah A. Pliner, Jay Shendure, and Cole Trapnell. 2019. Supervised classification enables rapid annotation of cell atlases. Nature Methods 16, 10 (2019), 983–986. https://doi.org/10.1038/s41592-019-0535-3 pmid:31501545

[248] Marius-Constantin Popescu, Valentina E. Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. 2020. Multilayer perceptron and neural networks. WSEAS Transactions on Circuits and Systems 8, 7 (2009), 579–588.

[249] Samira Pouryanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S. Iyengar. 2018. A survey on deep learning: Algorithms, techniques, and applications. ACM Computing Surveys (CSUR) 51, 5 (2018), 1–36.

[250] Sandhya Prabhakaran, Elham Azizi, Ambrose Carr, and Dana Pe’er. 2016. Dirichlet process mixture model for correcting technical variation in single-cell gene expression data. In International Conference on Machine Learning. PMLR, 1070–1079.

[251] Peng Qiu. 2020. Embracing the dropouts in single-cell RNA-seq analysis.66x66

[252] Anjali Rao, Dalia Barkley, Gustavo S. França, and Itai Yanai. 2021. Exploring tissue architecture using spatial transcriptomics. Nature 587, 7871 (Aug. 2021), 211–220. https://doi.org/10.1038/s41586-021-03634-9

[253] Jiahua Rao, Xiang Zhou, Yutong Lu, Huiying Zhao, and Yuedong Yang. 2021. Imputing single-cell RNA-seq data by combining graph convolution and autoencoder neural networks. Iiscience 24, 5 (2021), 102393.

[254] Pia Rautenstrauch, Anna Hendriks Cornelia Vlot, Sepideh Saran, and Uwe Ohler. 2021. Intricacies of single-cell multi-omics data integration. Trends in Genetics (2021).

[255] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 779–788.

[256] Aviv Regev, Sarah A. Teichmann, Eric S. Lander, Ido Amit, Christophe Benoist, Ewan Birney, Bernd Bodenmiller, Peter Campbell, Piero Carninci, Menna Clatworthy, et al. 2017. Science forum: The human cell atlas. Elife 6 (2017), e27041.

[257] Aviv Regev, Sarah A. Teichmann, Eric S. Lander, Ido Amit, Christophe Benoist, Ewan Birney, Bernd Bodenmiller, Peter Campbell, Piero Carninci, Menna Clatworthy, Hans Clevers, Bart Deplancke, Ian Dunham, James Eberwine, Roland Eils, Wolfgang Enard, Andrew Farmer, Lars Fugger, Berthold Göttgens, Nir Hacohen, Muzifah Haniffa, Martin Hemberg, Seung Kim, Paul Klenerman, Arnold Kriegstein, Ed Lein, Sten Linnarsson, Emma Lundberg, Joakim Lundeberg, Partha Majumder, John C. Marioni, Miriam Merad, Musa Mhlanga, Martijn Nawijn, Garry Nolan, Dana Pe’er, Anthony Phillipakis, Chris P. Ponting, Stephen Quake, Wolf Reik, Orit Rozenblatt-Rosen, Joshua Sanes, Rahul Satija, Ton N. Schubacher, Alex Shalek, Ehud Shapiro, Padmanee Sharma, Jay W. Shin, Oliver Stegle, Michael Stratton, Michael J. T. Stubbington, Fabian J. Theis, Matthias Uhlen, Alexander van Oudenaarden, Allon Wagner, Fiona Watt, Jonathan Weissman, Barbara Wold, Ramnik Xavier, Nir Yosef, and Human Cell Atlas Meeting Participants. 2017. The human cell atlas. Cell 170 (2017), 971–983.

[258] Samuel G. Rodrigues, Robert R. Stickels, Aleksandrina Goeva, Carly A. Martin, Evan Murray, Charles R. Vanderburg, Joshua Welch, Linlin M. Chen, Fei Chen, and Evan Z. Macosko. 2019. Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. Science 363, 6434 (2019), 1463–1467. https://doi.org/10.1126/science.aaw1219 Code Link: https://github.com/broadchenf/Slideseq

[259] Roman A. Romanov, Amit Zeisel, Joanne Bakker, Fatima Girach, Arash Halloyaz, Raju Tomer, Alan Alpar, Jan Mulder, Frederic Clotman, Erik Keimpema, et al. 2017. Molecular interrogation of hypothalamic organization reveals distinct dopamine neuronal subtypes. Nature Neuroscience 20, 2 (2017), 176–188. Data Link: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE74672

[260] Jonathan Ronen and Altunra Akalin. 2018. netSmooth: Network-smoothing based imputation for single cell RNA-seq. F1000Research 7 (2018).

[261] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 234–241.

[262] Yusuf Roohani, Kexin Huang, and Jure Leskovec. 2023. Predicting transcriptional outcomes of novel multigene perturbations with GEARs. Nature Biotechnology (2023), 1–9.

[263] Alexander B. Rosenberg, Charles M. Roco, Richard A. Muscat, Anna Kuchina, Paul Sample, Zizhen Yao, Lucas T. Graybuck, David J. Peeler, Sumit Mukherjee, Wei Chen, Suzie H. Pun, Drew L. Sellers, Bosiljka Tasic, and Georg Seelig. 2018. Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. Science 360, 6385 (April 2018), 176–182. https://doi.org/10.1126/science.aam8999
[264] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1985. Learning Internal Representations by Error Propagation. Technical Report. California Institute, San Diego, La Jolla Institute for Cognitive Science.

[265] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. Nature 323, 6088 (1986), 533–536.

[266] Karen Sachs, Omar Perez, Dana Pe’er, Douglas A. Lauffenburger, and Garry P. Nolan. 2005. Causal protein-signaling networks derived from multiparameter single-cell data. Science 308, 5721 (2005), 523–529.

[267] Alberto Santamaria-Pang, Jens Rittscher, Michael Gerdes, and Dirk Padfield. 2015. Cell segmentation and classification by hierarchical supervised shape ranking. In 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI ’15). 1296–1299. https://doi.org/10.1109/ISBI.2015.7164112

[268] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. IEEE Transactions on Neural Networks 20, 1 (2008), 61–80.

[269] Christian M. Schürch, Sial S. Bhaite, Graham L. Barlow, Darci J. Phillips, Luca Noti, Inti Zlobec, Pauline Chu, Sarah Black, Janos Demeter, David R. McIlwain, et al. 2020. Coordinated cellular neighborhoods orchestrate antitumoral immunity at the colorectal cancer invasive front. Cell 182, 5 (2020), 1341–1359.

[270] Peter J. Schüffler, Denis Schapiro, Charlotte Giesen, Hao A. O. Wang, Bernd Bodenmüller, and Joachim M. Buhmann. 2015. Automatic single cell segmentation on highly multiplexed tissue images. Cytoometry Part A 87, 10 (2015), 936–942. https://doi.org/10.1002/cyto.a.22702 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cyto.a.22702 Website Link: http://www.comp-path.inf.ethz.ch/

[271] Sheel Shah, Eric Lubeck, Wen Zhou, and Long Cai. 2016. In situ transcription profiling of single cells reveals spatial organization of cells in the mouse hippocampus. Neuron 92, 2 (Oct. 2016), 342–357. https://doi.org/10.1016/j.neuron.2016.10.001

[272] Xin Shao, Jie Liao, Xiaoyan Lu, Rui Xue, Ni Ai, and Xiaohui Fan. 2020. scCATCH: Automatic annotation on cell types of clusters from single-cell RNA sequencing data. Science 23, 3 (2020), 100882.

[273] Xin Shao, Jie Liao, Xiaoyan Lu, Rui Xue, Ni Ai, and Xiaohui Fan. 2020. scCATCH: Automatic annotation on cell types of clusters from single-cell RNA sequencing data. Science 23, 3 (2020).

[274] Xin Shao, Haihong Yang, Xiang Zhuang, Jie Liao, Penghui Yang, Junyun Cheng, Xiaoyan Lu, Huanjun Chen, and Xiaohui Fan. 2020. scDeepSort: A pre-trained cell-type annotation method for single-cell transcriptomics using deep learning with a weighted graph neural network. Nucleic Acids Research 49, 21 (2021), e122–e122.

[275] Rakesh Singal and Gordon D. Ginder. 1999. DNA methylation. Blood the Journal of the American Society of Hematology 93, 12 (1999), 4059–4070.

[276] Ritambhara Singh, Pinar Demeti, Giancarlo Bonora, Vijay Ramani, Choli Lee, He Fang, Zhijun Duan, Xinxian Deng, Sheel Shah, Eric Lubeck, Wen Zhou, and Long Cai. 2016. In situ transcription profiling of single cells reveals spatial organization of cells in the mouse hippocampus. Neuron 92, 2 (Oct. 2016), 342–357. https://doi.org/10.1016/j.neuron.2016.10.001

[277] VIPUL SINGHAL, NIGEL CHOU, JOSEF LANGER, LYTSPHIN, et al. 2020. A single-cell spatial transcriptomics pipeline for performing spatio-temporal analyses of biological processes. bioRxiv (2020). https://doi.org/10.1101/2020.06.12.199283

[278] Christian M. Schürch, Salil S. Bhaite, Graham L. Barlow, Darci J. Phillips, Luca Noti, Inti Zlobec, Pauline Chu, Sarah Black, Janos Demeter, David R. McIlwain, et al. 2020. Coordinated cellular neighborhoods orchestrate antitumoral immunity at the colorectal cancer invasive front. Cell 182, 5 (2020), 1341–1359.

[279] Peter J. Schüffler, Denis Schapiro, Charlotte Giesen, Hao A. O. Wang, Bernd Bodenmüller, and Joachim M. Buhmann. 2015. Automatic single cell segmentation on highly multiplexed tissue images. Cytoometry Part A 87, 10 (2015), 936–942. https://doi.org/10.1002/cyto.a.22702 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cyto.a.22702 Website Link: http://www.comp-path.inf.ethz.ch/

[280] Sheel Shah, Eric Lubeck, Wen Zhou, and Long Cai. 2016. In situ transcription profiling of single cells reveals spatial organization of cells in the mouse hippocampus. Neuron 92, 2 (Oct. 2016), 342–357. https://doi.org/10.1016/j.neuron.2016.10.001

[281] Charlotte Soneson and Mark D. Robinson. 2018. Bias, robustness and scalability in single-cell differential expression analysis. Nature Methods 15, 4 (2018), 255–261.

[282] Xiaohui Fan. 2021. scDeepSort: A pre-trained cell-type annotation method for single-cell transcriptomics using deep learning with a weighted graph neural network. Nucleic Acids Research 49, 21 (2021), e122–e122.

[283] Qianqian Song and Jing Su. 2021. DSTG: Deconvoluting spatial transcriptomics data through graph-based artificial intelligence. Briefings in Bioinformatics 22, 3 (2021), 1–13. Code Link: https://github.com/Su-informatics-lab/DSTG

[284] Sheel Shah, Eric Lubeck, Wen Zhou, and Long Cai. 2016. In situ transcription profiling of single cells reveals spatial organization of cells in the mouse hippocampus. Neuron 92, 2 (Oct. 2016), 342–357.https://doi.org/10.1016/j.neuron.2016.10.001

[285] Madhav V. Bhat, Lawrence A. Cafferty, William J. Baker, et al. 2019. Computational analysis of mammalian cell lines for single-cell analysis. Cell 87, 4 (Nov. 2019), 722–735. Code Link: https://github.com/b癌症项目/CellLineAnalysis

[286] Sheel Shah, Eric Lubeck, Wen Zhou, and Long Cai. 2016. In situ transcription profiling of single cells reveals spatial organization of cells in the mouse hippocampus. Neuron 92, 2 (Oct. 2016), 342–357. https://doi.org/10.1016/j.neuron.2016.10.001

[287] Charlotte Soneson and Mark D. Robinson. 2018. Bias, robustness and scalability in single-cell differential expression analysis. Nature Methods 15, 4 (2018), 255–261.

[288] Sanjay R. Srivatsan, José L. McFaline-Figueroa, Vijay Ramani, Lauren Saunders, Junyue Cao, Jonathan Packer, Hannah A. Pliner, Dana L. Jackson, Riza M. Daza, Lena Christiansen, et al. 2020. Massively multiplex chemical transcriptomics at single-cell resolution. Science 367, 6473 (2020), 45–51.

[289] Patrik L. Stähli, Fredrik Salmen, Sanja Wickovic, Anna Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacometto, Michaela Asp, Jakub O. Westholm, Mikael Huss, Annelle Mollbrink, Sten Linnarsson, Simone Codeluppi, Åke Borg, Fredrik Pontén, Paul Igor Costea, Pelin Sahlén, Jan Mulder, Olaf Bergmann, Joakim Lundeberg, and Jonas Frisen. 2016. Visualization and analysis of tissue expression in single sections by spatial transcriptomics. Science 353, 6294 (July 2016), 78–82. https://doi.org/10.1126/science.aaf2403
[285] Jay S. Stanley III, Scott Gigante, Guy Wolf, and Smita Krishnaswamy. 2020. Harmonic alignment. In Proceedings of the 2020 SIAM International Conference on Data Mining. SIAM, 316–324.

[286] Stefan G. Stark, Joanna Ficek, Francesco Locatello, Ximena Bonilla, Stéphane Chevrier, Franziska Singer, Gunnar Rättsch, and Kjørg- van Lehmann. 2020. SCIm: Universal single-cell matching with unpaired feature sets. Bioinformatics 36, Supplement_2 (2020), i919–i927. Code Link: https://github.com/ratschlab/scim

[287] Oliver Stegle, Sarah A. Teichmann, and John C. Marioni. 2015. Computational and analytical challenges in single-cell transcriptomics. Nature Reviews Genetics 16, 3 (March 2015), 133–145.

[288] Robert R. Stickells, Evan Murray, Pawan Kumar, Jilong Li, Jamie L. Marshall, Daniela J. Di Bella, Paola Arlotta, Evan Z. Macosko, and Fei Chen. 2020. Highly sensitive spatial transcriptomics at near-cellular resolution with Slide-seqV2. Nature Biotechnology 39, 3 (Dec. 2020), 313–319. https://doi.org/10.1038/s41587-020-0739-1

[289] Marlon Stoeckius, Christoph Hafemeister, William Stephenson, Brian Houck-Loomis, Pratip K. Chattopadhyay, Harold Swedlow, Rahul Satija, and Peter Smibert. 2017. Large-scale simultaneous measurement of epifluorescence and transcriptomes in single cells. Nature Methods 14, 9 (2017), 865–868.

[290] Marlon Stoeckius, Christoph Hafemeister, William Stephenson, Brian Houck-Loomis, Pratip K. Chattopadhyay, Harold Swedlow, Rahul Satija, and Peter Smibert. 2017. Simultaneous epifluorescence and transcriptome measurement in single cells. Nature Methods 14, 9 (2017), 865–868.

[291] Carsen Stringer and Marius Pachitariu. 2022. Cellpose 2.0: How to train your own model. bioRxiv (2022). https://doi.org/10.1101/2022.04.01.486764 arXiv:https://www.biorxiv.org/content/early/2022/04/02/2022.04.01.486764.full.pdf

[292] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. 2021. Cellpose: A generalist algorithm for cellular segmentation. Nature Methods 18, 1 (Jan. 2021), 100–106. https://doi.org/10.1038/s41592-020-01018-x Website Link: https://github.com/MouseLand/cellpose

[293] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Eftymia Papalexi, William M. Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. 2019. Comprehensive integration of single-cell data. Cell 177, 7 (2019), 1888–1902. Dataset Link: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE128639

[294] Patrik L. Stähl, Fredrik Salmén, Sanja Vickovic, Anna Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacomello, Michaella Asp, Jakub O. Westholm, Mikael Huss, Annelie Mollbrink, Sten Linnarsson, Simone Codeluppi, Åke Borg, Fredrik Pontén, Paul Igor Costea, Pelin Sahlin, Jan Mulder, Olaf Bergmann, Joakim Lundeberg, and Jonas Frisén. 2016. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. Science 353, 6294 (2016), 78–82. https://doi.org/10.1126/science.aaf2403

[295] A. K. Subramanian. 2020. PyTorch-VAE. https://github.com/AntixK/PyTorch-VAE

[296] Valentine Svensson, Kedar Nath Natarajan, Lam-Ha Ly, Ricardo J. Mirzaiga, Charlotte Labalette, Iain C. Macaulay, Ana Cvejic, and Sarah Teichmann. 2017. Power analysis of single-cell RNA-sequencing experiments. Nature Methods 14, 4 (2017), 381–387.

[297] Valentine Svensson, Sarah A. Teichmann, and Oliver Stegle. 2018. SpatialDE: Identification of spatially variable genes. Nature Methods 15, 5 (2018), 343–346. Code Link: https://github.com/Teichlab/SpatialDE

[298] Valentine Svensson, Roser Vento-Tormo, and Sarah A Teichmann. 2018. Exponential scaling of single-cell RNA-seq in the past decade. Nature Protocols 13, 4 (2018), 599–604.

[299] Damian Szklarczyk, Annika L. Gable, David Lyon, Alexander Junge, Stefan Wyder, Annelie Mollbrink, Sten Linnarsson, Simone Codeluppi, Åke Borg, Fredrik Pontén, Paul Igor Costea, Pelin Sahlin, Jan Mulder, Olaf Bergmann, Joakim Lundeberg, and Jonas Frisén. 2016. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. Science 353, 6294 (2016), 78–82. https://doi.org/10.1126/science.aaf2403

[300] Divyanshu Talwar, Aanchal Mongia, Debarka Sengupta, and Angshul Majumdar. 2018. AutoImpute: Autoencoder based imputation of single-cell RNA-seq data. Scientific Reports 8, 1 (2018), 1–11.

[301] Yuqi Tan and Patrick Cahan. 2019. SingleCellNet: A computational tool to classify single cell RNA-seq data across platforms and Across Species. Cell Systems 9, 2 (2019), 207–216.e2. https://doi.org/10.1016/j.cels.2019.06.004

[302] Fuchou Tang, Catalin Barbaciou, Yangzhou Wang, Ellen Nordman, Clarence Lee, Nanlan Xu, Xiaohui Wang, John Bodeau, Brian B. Tuch, Asim Siddiqui, Kaiqin Lao, and M. Azim Surani. 2009. mRNA-Seq: whole-transcriptome analysis of a single cell. Nature Methods 6, 5 (2009), 377–382.

[303] Jianxiang Tang, Jianxiao Zou, Mei Fan, Qi Tian, Jiayang Zhang, and Shichai Fan. 2021. Camelia: Imputation in single-cell methylomes based on local similarities between cells. Bioinformatics 37, 13 (2021), 1814–1820.

[304] Wenhao Tang, François Beriaux, Philipp Thomas, Claire Stefanelli, Malik Saint, Samuel Marguerat, and Vahid Shahrezaei. 2020. bayNorm: Bayesian gene expression recovery, imputation and normalization for single-cell RNA-sequencing data. Bioinformatics 36, 4 (2020), 1174–1181.

[305] Wenzhuo Tang, Hongzhi Wen, Renming Liu, Jiayuan Ding, Wei Jin, Yuying Xie, Hui Liu, and Jiliang Tang. 2023. Single-cell multimodal prediction via transformers. arXiv preprint arXiv:2303.00233 (2023).

[306] Joji Marie Teves and Kyoung Jae Won. 2020. Mapping cellular coordinates through advances in spatial transcriptomics technology. Molecules and Cells 43, 7 (2020), 591.
Deep Learning in Single-cell Analysis

[307] The Tabula Muris Consortium, Overall coordination, Logistical coordination, Organ collection and processing, Library preparation and sequencing, Computational data analysis, Cell type annotation, Writing group, Supplemental text writing group, and Principal investigators. 2018. Single-cell transcriptomics of 20 mouse organs creates a tabula muris. Nature 562, 7727 (Oct. 2018), 367–372. https://doi.org/10.1038/s41586-018-0590-4

[308] The Tabula Sapiens Consortium, Robert C. Jones, Jim Karkanias, Mark A. Krasnow, Angela Oliveira Pisco, Stephen R. Quake, Julia Salzman, Nir Yosef, Bryan Buithaup, Phillip Brown, William Harper, Marisa Hemezen, Ravikumar Fon-nusamy, Ahmad Salehi, Bhavani A. Sanagavarapu, Eileen Spallino, Ksenia A. Aaron, Waldo Concepcion, James M. Gardner, Burnett Kelly, Nikole Neidlinger, Zifa Wang, Sheela Crasta, Sarojo Kolluru, Maurizio Morri, Angela Oliveira Pisco, Serena Y. Tan, Kyle J. Travaglini, Chenling Xu, Marcela Alcântara-Hernández, Nicole Almanzar, Jane Antony, Benjamin Beyerders, Deviana Burhan, Kruti Calcuttawala, Matthew M. Carter, Charles K. F. Chan, Charles A. Chang, Stephen Chang, Alex Colville, Sheela Crasta, Rebecca N. Culver, Ivana Cvijović, Gaetano D’Amato, Camille Ezran, Francisco X. Galdos, Astrid Gillich, William R. Goodyer, Yan Hang, Alyssa Hayashi, Sahar Houshdaran, Xianxi Huang, Juan C. Irwin, SoRi Jang, Julia Valle Juanico, Aaron M. Kershner, SooChi Kim, Bernhard Kiss, Sarojo Kolluru, William Kong, Maya E. Kumar, Angera H. Kuo, Rebecca Leylek, Baoxiang Li, Gabriel B. Loeb, Wan-Jin Lu, Sruthi Mantri, Maxim Markovic, Patrick L. McAlpine, Antoine de Morree, Maurizio Morri, Karim Mrouj, Shrvani Mukherjee, Tyler Muser, Patrick Neuhofer, Thi D. Nguyen, Kimberly Perez, Ragini Phansalkar, Angela Oliveira Pisco, Nazan Puluca, Zhen Qi, Poorvi Rao, Hayley Raquer-McKay, Nicholas Schaum, Bronwyn Scott, Bobak Sedighzadeh, Joe Segal, Sushimita Sen, Shaheen Sikandar, Sean P. Spencer, Lea C. Steffes, Varun R. Subramanium, Aditi Sarup, Michael Swift, Kyle J. Travaglini, Will Van Treuren, Emily Trimmin, Stefan Veizades, Sivakamasundari Vijayakumar, Kim Chi Yo, Sevahn K. Vorperian, Wanxin Wang, Hannah N. W. Weinstein, Juliane Winkler, Timothy T. H. Wu, Jamie Xie, Andrea R. Yung, Yue Zhang, Angela M. Detweiler, Honey Mekonen, Norma F. Neff, Rene V. Sit, Michelle Tan, Jia Yan, Gregory R. Bean, Vivek Charu, Erna Forgö, Brock A. Martin, Michael G. Ozawa, Oscar Silva, Serena Y. Tan, Angus Toland, Venkata P. Vemuri, Shaked Afik, Kyle Awayan, Olga Borisovna Botvinnik, Ashley Byrne, Michelle Chen, Roobeh Dehghannisar, Angela M. Detweiler, Adam Gayoso, Alejandro A. Granados, Qingqi Li, Gita Mahmoudabadi, Aaron McGeever, Antoine de Morree, Julia Eve Olivieri, Madeline Park, Angela Oliveira Pisco, Neha Ravikumar, Julia Salzman, Geoffrey Stanley, Michael Swift, Michelle Tan, Weilun Tan, Alexander J. Tarashansky, Rohan Vanheusden, Sevahn K. Vorperian, Peter Wang, Sheng Wang, Galen Xing, Chenling Xu, Nir Yosef, Marcela Alcântara-Hernández, Jane Antony, Charles K. F. Chan, Charles A. Chang, Alex Colville, Sheela Crasta, Rebecca Culver, Les Dethlefsen, Camille Ezran, Astrid Gillich, Yan Hang, Yo-I Ho, Juan C. Irwin, SoRi Jang, Aaron M. Kershner, William Kong, Maya E. Kumar, Angera H. Kuo, Rebecca Leylek, Shixuan Liu, Gabriel B. Loeb, Wan-Jin Lu, Jonathan S. Maltzman, Ross J. Metzger, Antoine de Morree, Patrick Neuhofer, Kimberly Perez, Ragini Phansalkar, Zhen Qi, Poorvi Rao, Hayley Raquer-McKay, Koki Sasa3awa, Bronwyn Scott, Rahul Sinha, Hanbing Song, Sean P. Spencer, Aditi Sarup, Michael Swift, Kyle J. Travaglini, Emily Trimmin, Stefan Veizades, Sivakamasundari Vijayakumar, Bruce Wang, Wanxin Wang, Juliane Winkler, Jamie Xie, Andrea R. Yung, Steven E. Artandi, Philip A. Beachy, Michael F. Clarke, Linda C. Giudice, Franklin W. Huang, Kerwyn Casey Huang, Juliane Idoya, Seung K. Kim, Mark Krasnow, Christin S. Kuo, Patricia Nguyen, Stephen R. Quake, Thomas A. Rando, Kristy Red-Horse, Jeremy Reiter, David A. Relman, Justin L. Sonnenburg, Bruce Wang, Albert Wu, Sean M. Wu, and Tony Wyss-Coray. 2022. The tabula sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. Science 376, 6594 (May 2022), eabl4896. https://doi.org/10.1126/science.abl4896

[309] Todd Thorson, Richard W. Roberts, Frances H. Arnold, and Stephen R. Quake. 2001. Dynamic pattern formation in a vesicle-generating microfluidic device. Physical Review Letters 86, 18 (2001), 4163–4166.

[310] Robert Thurman et al. 2012. The accessible chromatin landscape of the human genome. Nature 489, 7414 (2012), 75–82.

[311] Tian Tian, Jie Zhang, Xiang Lin, Zhi Wei, and Hakon Hakonarson. 2021. Model-based deep embedding for constrained clustering analysis of single cell RNA-seq data. Nature Communications 12, 1 (2021), 1–12. Code Link: https://github.com/ttgump/scDeepCluster

[312] Duc Tran, Hung Nguyen, Bang Tran, Carlo La Vecchia, Hung N. Luu, and Tin Nguyen. 2021. Fast and precise single-cell data analysis using a hierarchical autoencoder. Nature Communications 12, 1 (2021), 1–10. Code Link: https://github.com/duc317/scDHA

[313] Phi Vu Tran. 2018. Learning to make predictions on graphs with autoencoders. In 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA ‘18). IEEE, 237–245.

[314] Kyle J. Travaglini, Ahmad N. Nabhan, Lolita Penland, Rahul Sinha, Astrid Gillich, Rene V. Sit, Stephen Chang, Stephanie D. Conley, Yasuo Mori, Jun Seita, et al. 2020. A molecular cell atlas of the human lung from single-cell RNA sequencing. Nature 578, 7835 (2020), 619–625.
[316] Sophie T Rivas, Marlen Buitzn, David S. Fischer, Marius Lange, Volker Bergen, Heiko Hickert, and Fabian J. Theis. 2019. Concepts and limitations for learning developmental trajectories from single cell genomics. Development 146, 12 (2019), dev170506.

[317] Daphne Tsoucas, Rui Dong, Haide Chen, Qian Zhu, Guoji Guo, and Guo-Cheng Yuan. 2019. Accurate estimation of cell-type composition from gene expression data. Nature Communications 10, 1 (July 2019). https://doi.org/10.1038/s41467-019-10802-z Code Link: https://github.com/dtsoucas/DWLS

[318] Fatima Valdes-Mora, Marisa Handler, Andrew M. K. Law, Robert Salomon, Samantha R. Oakes, Christopher J. Ormandy, and David Gallego-Ortega. 2018. Single-cell transcriptomics in cancer immunobiology: the future of precision oncology. Frontiers in Immunology 9 (2018), 2582.

[319] Koen Van den Berge, Hector Roux de Bézieux, Kelly Street, Wouter Saelens, Robrecht Cannoodt, Yvan Saey, Sandrine Dudoit, and Lieven Clement. 2020. Trajectory-based differential expression analysis for single-cell sequencing data. Nature Communications 11, 1 (2020), 1–13.

[320] David van Dijk, Juzonas Nainys, Roshan Sharma, Pooja Kaithail, Ambrose J. Carr, Kevin R. Moon, Linas Mazutis, Guy Wolf, Smita Krishnaswamy, and Dana Pe’er. 2017. MAGIC: A diffusion-based imputation method reveals gene-gene interactions in single-cell RNA-seq data. bioRxiv (2017), 111591.

[321] David A. Van Valen, Takamasa Kudo, Keara M. Lane, Derek N. Macklin, Nicolas T. Quach, MiaLy M. DeFelice, Inbal Maayan, Yu Tanouchi, Euan A. Ashley, and Markus W. Covert. 2016. Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. PLoS Computational Biology 12, 11 (2016), e1005177.

[322] Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Arno Solin, Yoshua Bengio, and David Lopez-Paz. 2022. Interpolation consistency training for semi-supervised learning. Neural Networks 145 (2022), 90–106.

[323] Luc Vincent and Pierre Soille. 1991. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis & Machine Intelligence 13, 6 (1991), 583–598. Code Link: https://github.com/BatzoglouLabSU/SIMLR

[324] Luke F. Vistain and Savaş Tay. 2021. Single-cell proteomics. Trends in Biochemical Sciences 46, 8 (2021), 661–672.

[325] Johanna Wagner, Maria Anna Rapsomaniki, Stéphane Chevrier, Tobias Anzeneder, Claus Langwieder, August Dykgers, Martin Rees, Annette Ramsawamy, Simone Muenst, Savas Deniz Soysal, et al. 2019. A single-cell atlas of the tumor and immune ecosystem of human breast cancer. Cell 177, 5 (2019), 1330–1345.

[326] Bo Wang, Daniele Ramazzotti, Luca De Sano, Junjie Zhu, Emma Pierson, and Serafim Batzoglou. 2018. SIMLR: A tool for large-scale genomic analyses by multi-kernel learning. Proteomics 18, 2 (2018), 1700232. Code Link: https://github.com/juexinwang/scGNN

[327] Guiping Wang, Jeffrey R. Moffitt, and Xiaowei Zhuang. 2018. Multiplexed imaging of high-density libraries of RNAs with MERFISH and expansion microscopy. Scientific Reports 8, 1 (2018), 1–13.

[328] Juexin Wang, Anjun Ma, Yuzhou Chang, Jiating Gong, Yuexu Jiang, Ren Qi, Cankun Wang, Hongjun Fu, Qin Ma, and Dong Xu. 2021. scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses. Nature Communications 12, 1 (2021), 1–11. Code Link: https://github.com/juexinwang/scGNN

[329] Minjie Wang, Da Zheng, Zhihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, et al. 2019. Deep graph library: A graph-centric, highly-performant package for graph neural networks. arXiv preprint arXiv:1909.01315 (2019).

[330] Tianyu Wang, Jun Bai, and Sheida Nabavi. 2021. Single-cell classification using graph convolutional networks. BMC Bioinformatics 22, 1 (2021), 364. https://doi.org/10.1186/s12859-021-04278-2

[331] Yong Wang and Nicholas E. Navin. 2015. Advances and applications of single-cell sequencing technologies. Molecular Cell 58, 4 (2015), 598–609.

[332] Yixin Wang, Yang Zhang, Yang Liu, Zhihao Lin, Jiang Tian, Cheng Zhong, Zhongchao Shi, Jianping Fan, and Zhiqiang He. 2021. ACN: Adversarial co-training network for brain tumor segmentation with missing modalities. In International Conference on Medical Image Computing and Computer-assisted Intervention. Springer, 410–420.

[333] Yixin Wang, Yao Zhang, Yang Liu, Zihao Lin, Jiang Tian, Cheng Zhong, Zhongchao Shi, Yang Zhang, and Zhiqiang He. 2020. Double-uncertainty weighted method for semi-supervised learning. In International Conference on Medical Image Computing and Computer-assisted Intervention. Springer, 542–551.

[334] Jiaxuan Wangwu, Zexuan Sun, and Zhixiang Lin. 2021. scAMACE: Model-based approach to the joint analysis of single-cell data on chromatin accessibility, gene expression and methylation. Bioinformatics 37, 21 (2021), 3874–3880.

[335] Lisa N. Waylen, Hieu T. Nam, Luciano G. Martelotto, and Mirana Ramialison. 2020. From whole-mount to single-cell spatial assessment of gene expression in 3D. Communications Biology 3, 1 (2020), 1–11.

[336] Caleb Weinreb, Samuel Wolock, and Allon M. Klein. 2018. SPRING: A kinetic interface for visualizing high dimensional single-cell expression data. Bioinformatics 34, 7 (2018), 1246–1248. Code Link: https://github.com/AllonKleinLab/SPRING

[337] Joshua D. Welch, Alexander J. Hartemink, and Jan F. Prins. 2017. MATCHER: Manifold alignment reveals correspondence between single cell transcriptome and epigenome dynamics. Genome Biology 18, 1 (2017), 1–19.
[338] Hongzhi Wen, Jiayuan Ding, Wei Jin, Yiqi Wang, Yuying Xie, and Jiliang Tang. 2022. Graph neural networks for multimodal single-cell data integration. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 4153–4163. Code Link: https://github.com/OmicsML/dance

[339] Lu Wen and Fuchou Tang. 2022. Recent advances in single-cell sequencing technologies. Precision Clinical Medicine 5, 1 (2022), pbac002.

[340] Wei-Hung Weng, Mingwu Gao, Zhe He, Susu Yan, and Peter Szolovits. 2017. Representation and reinforcement learning for personalized glycemic control in septic patients. arXiv preprint arXiv:1712.00654 (2017).

[341] George M. Whitesides. 2006. The origins and the future of microfluidics. Nature 442, 7101 (2006), 368–373.

[342] Kevin E. Wu, Kathryn E. Yost, Howard Y. Chang, and James Zou. 2021. BABEL enables cross-modality translation between multomic profiles at single-cell resolution. Proceedings of the National Academy of Sciences 118, 15 (2021), e202307118. Code Link: https://github.com/wukevin/babel

[343] Junyu Yang, Anastasiya Belyaeva, Saradha Venkatachalapathy, Karthik Damodaran, Abigail Katcoff, Adityanarayanan Radhakrishnan, G. V. Shivashankar, and Caroline Uhler. 2021. Multi-domain translation between single-cell imaging and sequencing data using autoencoders. Nature Communications 12, 1 (2021), 1–10. Code Link: https://github.com/OmicsML/dance

[344] Peng Xie, Mingxuan Gao, Chunming Wang, Jianfei Zhang, Pawan Noel, Chaoyong Yang, Haiyong Han, Michael Q. Zhang, and Wei Lin. 2019. SuperCT: A supervised-learning framework for enhanced characterization of single-cell transcriptomic profiles. Nucleic Acids Research 47, 8 (2019), e48–e48. https://doi.org/10.1093/nar/gkz116

[345] Haiqing Xiong, Yingjie Luo, Qianhao Wang, Xianhong Yu, and Aibin He. 2021. Single-cell joint detection of chromatin occupancy and transcriptome enables higher-dimensional epigenomic reconstructions. Nature Methods 18, 6 (May 2021), 652–660. https://doi.org/10.1038/s41592-021-01129-z

[346] Chen Xu and Zhengchang Su. 2015. Identification of cell types from single-cell transcriptomes using a novel clustering method. Bioinformatics 31, 12 (2015), 1974–1980.

[347] Junlin Xu, Lijun Cai, Bo Liao, Wen Zhu, and Jialiang Yang. 2020. CMF-Impute: An accurate imputation tool for single-cell RNA-seq data. Bioinformatics 36, 10 (2020), 3139–3147.

[348] Yang Xu, Priyojit Das, and Rachel Patton McCord. 2022. SMILE: Mutual information learning for integration of single-cell omics data. Bioinformatics 38, 2 (2022), 476–486. Code Link: https://github.com/rpmccordlab/SMILE.

[349] Yungang Xu, Zhigang Zhang, Lei You, Jiajia Liu, Zhiwei Fan, and Xiaobo Zhou. 2020. sciGANs: Single-cell RNA-seq imputation using generative adversarial networks. Nucleic Acids Research 48, 15 (2020), e85–e85.

[350] Zhe Xu, Donghuan Lu, Yixin Wang, Jie Luo, Jagadeesan Jayender, Kai Ma, Yefeng Zheng, and Xiu Li. 2021. Noisy labels are treasure: Mean-teacher-assisted confident learning for hepatic vessel segmentation. In International Conference on Medical Image Computing and Computer-assisted Intervention. Springer, 3–13.

[351] Samir S. Yadav and Shivajirao M. Jadhav. 2019. Deep convolutional neural network based medical image classification for disease diagnosis. Journal of Big Data 6, 1 (2019), 1–18.

[352] Pingkun Yan, Xiaobo Zhou, Mubarak Shah, and Stephen T. C. Wong. 2008. Automatic segmentation of high-throughput RNAi fluorescent cellular images. IEEE Transactions on Information Technology in Biomedicine 12, 1 (2008), 109–117. https://doi.org/10.1109/TITB.2007.898006

[353] Ka Yee Yeung and Walter L. Ruzzo. 2001. Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. Bioinformatics 17, 9 (2001), 763–774.

[354] Qingyang Yin, Yang Wang, Jenching Guan, and Guoli Ji. 2022. scIAE: An integrative autoencoder-based ensemble classification framework for single-cell RNA-seq data. Briefings in Bioinformatics 23, 1 (2022), bbab508. https://doi.org/10.1093/bib/bbab508

[355] Ka Yee Yeung and Walter L. Ruzzo. 2001. Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. Bioinformatics 17, 9 (2001), 763–774.

[356] Pengcheng Zeng and Zhixiang Lin. 2021. CoupleCoC+: An information-theoretic co-clustering-based transfer learning method. Bioinformatics 37, 1 (2021), 4671–4679. Code Link: https://github.com/Philyzh8/scTAG

[357] Zhuohan Yu, Yifu Lu, Yunhe Wang, Fan Tang, Ka-Chun Wong, and Xiangtang Li. 2022. ZINB-based graph embedding autoencoder for single-cell RNA-seq interpretations. Proceedings of the AAAI Conference on Artificial Intelligence 36, 4 (2022), 4671–4679. Code Link: https://github.com/Philyzh8/scTAG

[358] Pengcheng Zeng and Zhixiang Lin. 2021. CoupleCoC+: An information-theoretic co-clustering-based transfer learning method. Bioinformatics 37, 1 (2021), 4671–4679. Code Link: https://github.com/Philyzh8/scTAG

[359] Wanwen Zeng, Xi Chen, Zhana Duren, Yong Wang, Rui Jiang, and Wing Hung Wong. 2019. DC3 is a method for deconvolution and coupled clustering from bulk and single-cell genomics data. Nature Communications 10, 1 (2019), 1–11.
[360] Yuansong Zeng, Xiang Zhou, Jiahua Rao, Yutong Lu, and Yuedong Yang. 2020. Accurately clustering single-cell RNA-seq data by capturing structural relations between cells through graph convolutional network. In 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM ’20). IEEE, 519–522. Code Link: https://github.com/biomed-AI/GraphSCC

[361] Leonie L. Zeune, Yoeri E. Boink, Guus van Dalum, Afroditi Nanou, Sanne de Wit, Kiki C. Andree, Joost F. Swennenhuis, Stephan A. van Gils, Leon W. M. M. Terstappen, and Christophe Brune. 2020. Deep learning of circulating tumour cells. Nature Machine Intelligence, 2 (2020), 124–133.

[362] Allen W. Zhang, Ciara O’Flanagan, Elizabeth A. Chavez, Jamie L. P. Lim, Nicholas Ceglia, Andrew McPherson, Matt Wiens, Pascale Walters, Tim Chan, Briana Hewitson, Daniel Lai, Anja Mottok, Clementine Sarkozy, Lauren Chong, Tomohiro Aoki, Xuehai Wang, Andrew P. Weng, Jessica N. McAlpine, Samuel Aparicio, Christian Steidl, Kieran R. Campbell, and Sohrab P. Shah. 2019. Probabilistic cell-type assignment of single-cell RNA-seq for tumor microenvironment profiling. Nature Methods 16, 10 (2019), 1007–1015.

[363] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2017. Mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017). https://doi.org/10.48550/ARXIV.1710.09412

[364] Yuanchao Zhang, Man S. Kim, Erin R. Reichenberger, Ben Stear, and Deanne M. Taylor. 2020. Scedar: A scalable Python package for single-cell RNA-seq exploratory data analysis. PLoS Computational Biology 16, 4 (2020), e1007794. Code Link: https://pypi.org/project/scedar

[365] Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2020. Deep learning on graphs: A survey. IEEE Transactions on Knowledge and Data Engineering (2020).

[366] Ze Zhang, Danni Luo, Xue Zhong, Jin Huk Choi, Yuanqing Ma, Stacy Wang, Elena Mahrt, Wei Guo, Eric W. Stawiski, Zora Modrusan, et al. 2019. SCINA: A semi-supervised subtyping algorithm of single cells and bulk samples. Genes 10, 7 (2019), 531.

[367] Ziqi Zhang, Chengkai Yang, and Xiuwei Zhang. 2022. scDART: Integrating unmatched scRNA-seq and scATAC-seq data and learning cross-modality relationship simultaneously. Genome Biology 23, 1 (2022), 1–28. Code Link: https://github.com/PeterZZQ/scDART_test

[368] Edward Zhao, Matthew R. Stone, Xing Ren, Jamie Guenthoer, Kimberly S. Smythe, Thomas Pulliam, Stephen R. Williams, Cedric R. Uytingco, Sarah E. B. Taylor, Paul Nghiem, et al. 2017. Massively parallel digital transcriptional profiling of single cells. Nature Communications 8, 1 (2017), 1–12. Data Link: https://support.10xgenomics.com/single-cell-gene-expression/datasets/2.1.0/pbmc4k

[369] Yanning Zhou, Omer Fahrri Onder, Qi Dou, Efstathios Tsougenis, Hao Chen, and Pheng-Ann Heng. 2019. Cia-net: Robust nuclei instance segmentation with contour-aware information aggregation. In International Conference on Information Processing in Medical Imaging. Springer, 682–693.

[370] Chenxu Zhu, Miao Yu, Hui Huang, Ivan Juric, Armen Abnousi, Rong Hu, Jacinta Lucero, M. Margarita Behrens, Ming Hu, and Bing Ren. 2019. An ultra high-throughput method for single-cell joint analysis of open chromatin and transcriptome. Nature Structural & Molecular Biology 26, 11 (2019), 1063–1070.

[371] Chenxu Zhu, Yanxiao Zhang, Yang Eric Li, Jacinta Lucero, M. Margarita Behrens, and Bing Ren. 2021. Joint profiling of histone modifications and transcriptome in single cells from mouse brain. Nature Methods 18, 3 (Feb. 2021), 283–292. https://doi.org/10.1038/s41592-021-01060-3

[372] Lingxue Zhu, Jing Lei, Lambertus Klei, Bernie Devlin, and Kathryn Roeder. 2019. Semisoft clustering of single-cell data. Proceedings of the National Academy of Sciences 116, 2 (2019), 466–471. Code Link: https://github.com/lingxuezu/SOUPR

[373] Qian Zhu, Sheel Shah, Ruben Dries, Long Cai, and Guo-Cheng Yuan. 2018. Identification of spatially associated subpopulations by combining scRNAseq and sequential fluorescence in situ hybridization data. Nature Biotechnology 36, 12 (2018), 1183–1190.

[374] Maya Ziv, Gil Gruber, Moran Sharon, Ekaterina Vinogradov, and Esti Yeger-Lotem. 2022. The tissuenet v. 3 database: Protein-protein interactions in adult and embryonic human tissue contexts. Journal of Adult and Molecular Biology 434, 11 (2022), 167532.

[375] Chunman Zuo and Luonan Chen. 2021. Deep-joint-learning analysis model of single cell transcriptome and open chromatin accessibility data. Briefings in Bioinformatics 22, 4 (2021), bbaa287. Code Link: https://github.com/cmzuo11/scMVAE

[376] Chunman Zuo, Hao Dai, and Luonan Chen. 2021. Deep cross-omics cycle attention model for joint analysis of single-cell multi-omics data. Bioinformatics 37, 22 (2021), 4091–4099. Code Link: https://github.com/cmzuo11/DCCA

Received 20 March 2023; revised 14 November 2023; accepted 29 December 2023