Biophysics Reviews

REVIEW ARTICLE | FEBRUARY 07 2023

Deep learning in spatial transcriptomics: Learning from the next next-generation sequencing

A. Ali Heydari ; Suzanne S. Sindi

Biophysics Rev. 4, 011306 (2023)
https://doi.org/10.1063/5.0091135

AIP Publishing

Biophysics Reviews REVIEW scitation.org/journal/bpr

Deep learning in spatial transcriptomics: Learning from the next next-generation sequencing

Cite as: Biophysics Rev. 4, 011306 (2023); doi: 10.1063/5.0091135
Submitted: 11 March 2022 · Accepted: 19 December 2022 ·
Published Online: 7 February 2023

A. Ali Heydari[1,2] and Suzanne S. Sindi[1,2,a)]

AFFILIATIONS

[1]Department of Applied Mathematics, University of California, Merced, California 95343, USA
[2]Health Sciences Research Institute, University of California, Merced, California 95343, USA

[a)]Author to whom correspondence should be addressed: ssindi@ucmerced.edu

ABSTRACT

Spatial transcriptomics (ST) technologies are rapidly becoming the extension of single-cell RNA sequencing (scRNAseq), holding the potential of profiling gene expression at a single-cell resolution while maintaining cellular compositions within a tissue. Having both expression profiles and tissue organization enables researchers to better understand cellular interactions and heterogeneity, providing insight into complex biological processes that would not be possible with traditional sequencing technologies. Data generated by ST technologies are inherently noisy, high-dimensional, sparse, and multi-modal (including histological images, count matrices, etc.), thus requiring specialized computational tools for accurate and robust analysis. However, many ST studies currently utilize traditional scRNAseq tools, which are inadequate for analyzing complex ST datasets. On the other hand, many of the existing ST-specific methods are built upon traditional statistical or machine learning frameworks, which have shown to be sub-optimal in many applications due to the scale, multi-modality, and limitations of spatially resolved data (such as spatial resolution, sensitivity, and gene coverage). Given these intricacies, researchers have developed deep learning (DL)-based models to alleviate ST-specific challenges. These methods include new state-of-the-art models in alignment, spatial reconstruction, and spatial clustering, among others. However, DL models for ST analysis are nascent and remain largely underexplored. In this review, we provide an overview of existing state-of-the-art tools for analyzing spatially resolved transcriptomics while delving deeper into the DL-based approaches. We discuss the new frontiers and the open questions in this field and highlight domains in which we anticipate transformational DL applications.

TABLE OF CONTENTS

I. INTRODUCTION

Although multicellular organisms contain a common genome within their cells, the morphology and gene expression patterns of cells are largely distinct and dynamic. These differences arise from internal gene regulatory systems and external environmental signals. Cells

proliferate, differentiate, and function in tissues while sending and receiving signals from their surroundings. These environmental factors cause cell fate to be highly dependent on the environment in which it exists. Therefore, monitoring a cell's behavior in the residing tissue is crucial to understanding cell function, as well as its past and future fate.[1]

Advancements in single-cell sequencing have transformed the genomics and bioinformatics fields. The advent of single-cell RNA sequencing (scRNAseq) has enabled researchers to profile gene expression levels of various tissues and organs, allowing them to create comprehensive atlases in different species.[2–6] Moreover, scRNAseq enables the detection of distinct subpopulations present within a tissue, which has been paramount in discovering new biological processes, the inner workings of diseases, and effectiveness of treatments.[7–14] However, high-throughput sequencing of solid tissues requires tissue dissociation, resulting in the loss of spatial information.[15,16] To fully understand cellular interactions, data on tissue morphology and spatial information are needed, which scRNAseq alone cannot provide. The placement of cells within a tissue is crucial from the developmental stages (e.g., asymmetric cell fate of mother and daughter cells[17]) and beyond cell differentiation (such as cellular functions, response to stimuli, and tissue homeostasis[18]). These limitations would be alleviated by technologies that could preserve spatial information while measuring gene expression at the single-cell level.

Spatial transcriptomics (ST) provide an *unbiased* view of tissue organization crucial in understanding cell fate, delineating heterogeneity, and other applications.[19] However, many current ST technologies suffer from lower sensitivities as compared to scRNAseq while lacking the single-cell resolution that scRNAseq provides.[20] Targeted *in situ* technologies have tried to solve the issue of resolution and sensitivity but are limited in gene throughput and often require *a priori* knowledge of target genes.[20] More specifically, *in situ* technologies [such as *in situ* sequencing (ISS),[21] single-molecule fluorescence *in situ* hybridization (smFISH),[22–24] targeted expansion sequencing,[25] cyclic-ouroboros smFISH (osmFISH),[26] multiplexed error-robust fluorescence *in situ* hybridization (MERFISH),[27] sequential FISH (seqFISH+),[28] and spatially resolved transcript amplicon readout mapping (STARmap)[29]] are typically limited to pre-selected genes that are on the order of hundreds, with the accuracy potentially dropping as more probes are added.[29] We will refer to these methods as *image-based* techniques.

On the other hand, next generation sequencing (NGS)-based technologies [such as 10× Genomics' Visium and its predecessor,[30,31] Slide-Seq,[32] and high-definition spatial transcriptomics (HDST[33])] barcode entire transcriptomes but have limited capture rates, and resolutions that are larger than a single cell[34] (50–100 μm for Visium and 10 μm for Slide-Seq). Moreover, unlike image-based technologies, NGS-based methods allow for unbiased profiling of large tissue sections without necessitating a set of target genes.[35,36] However, NGS-based technologies do not have a single-cell resolution, requiring cellular features to be inferred or related to the histological scale using computational approaches. Many current algorithms use traditional statistical or medical image processing frameworks that require human supervision,[34,37,38] which is not ideal for large-scale analyses. Additionally, many algorithms are not generalizable across different sequencing platforms, which limits their utility and restricts multi-omics integration efforts.

Deep learning (DL) methods can use raw data to extract useful representations (or information) needed for performing a task, such as classification or detection.[39] This quality makes this class of Machine Learning (ML) algorithms ideal for applications where the available data are large, higher-dimensional, and noisy, such as single-cell omics. DL models have been extensively used in scRNAseq studies (e.g., preprocessing,[40,41] clustering,[42,43] cell-type identification,[44,45] and data augmentation[46,47]) and have shown to significantly improve upon traditional methods,[10] suggesting the potential of such methods in ST analysis. Moreover, DL models can leverage multiple data sources, such as images and text data, to learn a set of tasks.[48] Given that spatially resolved transcriptomics are inherently multi-modal (i.e., they consist of images and gene expression count data) and that downstream analysis consists of multiple tasks (e.g., clustering and cell-type detection), researchers have sought to develop ST-specific DL algorithms.

Spatially resolved transcriptomics have been utilized to unravel complex biological processes in many diseases (e.g., COVID-19,[49,50] arthritis,[51,52] cancer,[31,33,53–55] Alzheimer's,[56] diabetes,[57,58] etc.). Continuous improvements and commercialization of ST technologies (such as 10×'s Visium) are resulting in wider use across individual labs. Therefore, scalable and platform-agnostic computational approaches are needed for the accurate and robust analysis of ST data. So far, DL methods have shown promising results in handling the scale and multi-modality of spatially resolved transcriptomics; however, DL-based models in this space remain nascent. Similar to scRNAseq analysis, we anticipate a suite of DL models to be developed in the near future to address many of the pressing challenges in the spatial omics field. This review aims to provide an overview of the current state-of-the-art (SOTA) DL models developed for ST analysis. Due to the potential and accessibility of NGS-based ST technologies, we primarily focus on methods and techniques developed for these technologies.

The remainder of this manuscript is organized as follows: We provide an overview of common scRNAseq and ST technologies in Sec. II, followed by a general description of common DL architectures used for ST analysis in Sec. III. Section IV is dedicated to the current DL methods developed for analyzing spatially resolved transcriptomics. In Sec. V, we conclude by discussing our outlook on the current challenges and future research directions in the ST domain. We provide an illustration of the main methods reviewed in this paper in Fig. 1 and provide the reader with a list of current SOTA methods for ST analysis in Table I. Given the pace of advancements in this field, the authors have compiled an online list of current DL methods for ST analysis on a dedicated repository (https://github.com/SindiLab/Deep-Learning-in-Spatial-Transcriptomics-Analysis), which will be maintained and continuously updated.

II. BIOLOGICAL BACKGROUND

A. Single-cell RNA sequencing (scRNAseq)

RNA sequencing (RNA-seq) provides comprehensive insight on cellular processes (such as identifying genes that are upregulated or downregulated, etc.). However, traditional bulk RNA-seq is limited to revealing the average expression from a collection of cells, and not disambiguation single-cell behavior. Thus, it is difficult to delineate cellular heterogeneity with traditional RNA-seq, which is a disadvantage since cellular heterogeneity has been shown to play a crucial role in understanding many diseases.[83] Therefore, researchers have turned to single-cell RNA-seq (scRNAseq) in order to identify cellular

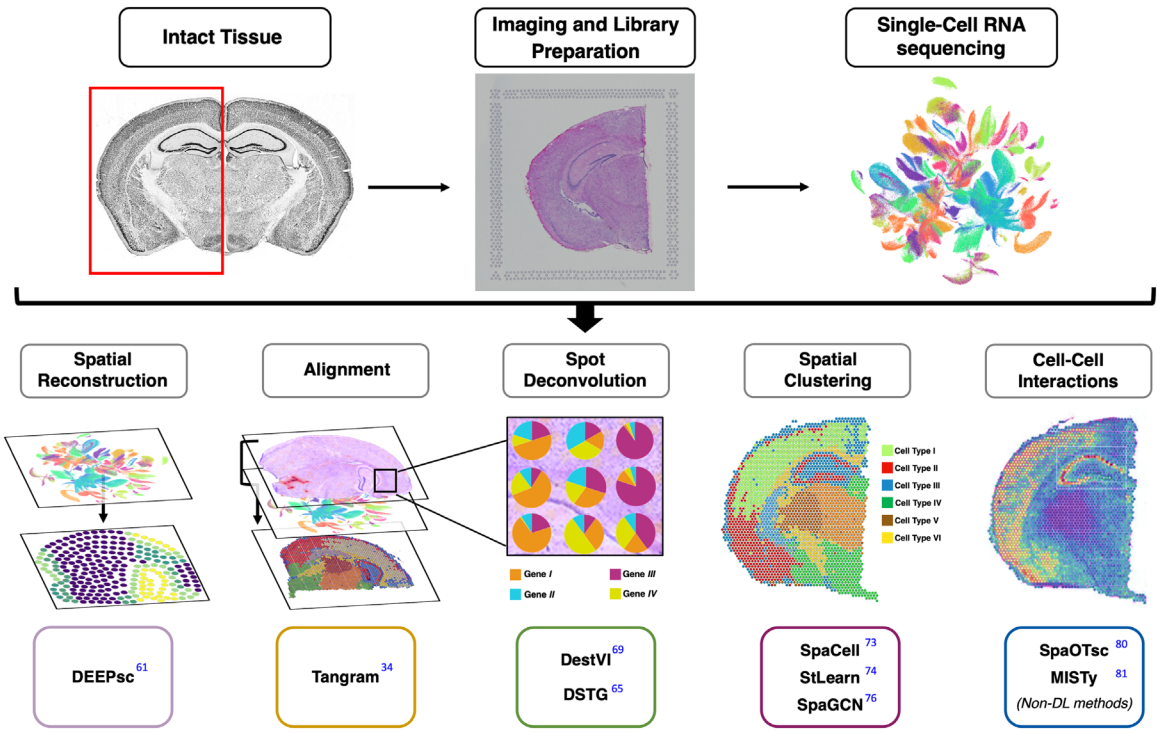

FIG. 1. An overview of deep learning (and machine learning) methods for spatial transcriptomics presented in this review. In this work, we provide a brief background on related biological concepts, such as single-cell RNA sequencing (scRNAseq) and spatial transcriptomic (ST) technologies (Sec. II), followed by an overview of common deep learning architectures in Sec. III. We then dive deeper into specific machine learning techniques for spatial reconstruction (Sec. IV A), scRNAseq and ST alignment (Sec. IV B), ST spot deconvolution (Sec. IV C), spatial clustering (Sec. IV D), and cell–cell interaction (Sec. IV E). A more comprehensive list of the state-of-the-art methods for spatial transcriptomics is provided in Table I.

heterogeneity within tissues. ScRNAseq technologies have been instrumental in the study of key biological processes in many diseases, such as cancer,[84] Alzheimer's,[85] cardiovascular diseases,[86] etc. (see Ref. 83 for more details). RNA sequencing of cells at a single-cell resolution, scRNAseq, generally consists of four stages:

(i) **Isolation of single-cells and lysing**: Cells are selected through laser microdissection, fluorescence-activated cell sorting (FACS), microfluidic/microplate technology (MT), or a combination of these methods,[87] with MT being highly complementary to NGS-based technologies.[88] MT encapsulates each single-cell into an independent microdroplet containing unique molecular identifiers (UMI), lysis buffer for cell lysis (to increase the capturing of as many RNA molecules as possible), oligonucleotide primers, and deoxynucleotide triphosphates (dNTPs) in addition to the cells themselves. Due to MT's higher isolation capacity, thousands of cells can be simultaneously tagged and analyzed, which is beneficial for large-scale scRNAseq studies.

(ii) **Reverse transcription**: One challenge in RNA sequencing is that RNA cannot be directly sequenced from cells, and thus, RNA must first be converted to complementary DNA (cDNA).[89] Although various technologies employ different techniques, the reverse transcription phase generally involves capturing mRNA using poly[T] sequence primers that bind to mRNA ploy[A] tail prior to cDNA conversion. Based on the sequencing platform, other nucleotide sequences are added to the reverse transcription; for example, in NGS protocols, UMIs are added to tag unique mRNA molecules so that it could be traced back to their originating cells, enabling the combination of different cells for sequencing.

(iii) **cDNA amplification**: Given that RNA cannot be directly sequenced from cells, single-stranded RNAs must first be reverse transcribed to cDNA. However, due to the small amount of mRNA in cells, limited cDNA is produced, which is not optimal for sequencing. Therefore, the limited quantity of cDNA must be amplified prior to library preparation and sequencing.[90] The amplification is often done by either PCR (an exponential amplification process with its efficiency being sequence dependent) or IVT (a linear amplification method that requires an additional round of reverse transcription of the amplified RNA) before sequencing.[89,91] The final cDNA library consists of an adaptor-ligated sequencing library attached to each end.

(iv) **Sequencing library construction**: Finally, every cell's tagged and amplified cDNA is combined for library preparation and sequencing similar to bulk RNA sequencing methods, followed by computational pipelines for processing and analysis.[92]

TABLE I. A list of relevant methods for the analysis of spatial transcriptomics data. The italicized boldfaced methods are the ones that utilize deep learning (or elements closely aligned). We review these methods in depth in this paper.

Category	Method	Year	Framework	Language	Software availability
Spatial reconstruction	Seurat[59]	2015	Statistical	R	https://github.com/satijalab/seurat
	novoSpaRc[60]	2019	Optimization/statistical	Python	https://github.com/rajewsky-lab/novosparc
	DEEPsc[61]	2021	Machine learning	MATLAB	https://github.com/fmaseda/DEEPsc
Alignment and integration	Spatial backmapping[62]	2015	Scoring scheme	R	https://github.com/jbogp/nbt_spatial_backmapping
	Tangram[34]	2021	Machine learning	Python	https://github.com/broadinstitute/Tangram
	GLUER[63]	2021	Machine learning	Python	https://github.com/software-github/GLUER
Spot deconvolution	Stereoscope[64]	2020	Statistical	Python	https://github.com/almaan/stereoscope
	DSTG[65]	2021	Machine learning	Python, R	https://github.com/Su-informatics-lab/DSTG
	SPOTlight[66]	2021	Machine learning	R	https://github.com/MarcElosua/SPOTlight
	RTCD[67]	2021	Statistical	R	https://github.com/dmcable/RCTD
	SpatialDWLS[68]	2021	Optimization/statistical	R	https://github.com/RubD/Giotto
	DestVI[69]	2021	Machine learning	Python	https://github.com/YosefLab/scvi-tools
	Cell2location[70]	2022	Statistical	Python	https://github.com/BayraktarLab/cell2location
	CARD[71]	2022	Machine learning	R, C++	https://github.com/YingMa0107/CARD
Spatial clustering	HMRF[72]	2018	Statistical	R, Python, C	https://bitbucket.org/qzhudfci/smfishhmrf-py/
	SpaCell[73]	2019	Machine learning	Python	https://github.com/BiomedicalMachineLearning/SpaCell
	StLearn[74]	2020	Machine learning	Python	https://github.com/BiomedicalMachineLearning/stLearn
	BayesSpace[75]	2021	Statistical	R, C++	https://github.com/edward130603/BayesSpace
	SpaGCN[76]	2021	Machine learning	Python	https://github.com/jianhuupenn/SpaGCN
Spatially variable genes identification	Trendsceek[77]	2018	Statistial	R	https://github.com/edsgard/trendsceek
	SpatialDE[78]	2018	Statistical	Python	https://github.com/Teichlab/SpatialDE
	Spark[79]	2020	Statistical	R, C++	https://github.com/xzhoulab/SPARK
Cell–cell communication	SpaOTsc[80]	2020	Machine learning	Python	https://github.com/zcang/SpaOTsc
	MISTy[81]	2020	Machine learning	R	https://github.com/saezlab/mistyR
	Giotto[82]	2021	Statistical	R	https://github.com/RubD/Giotto

Figure 2(A) illustrates an example of the workflow for scRNAseq. For more details of each stage and various scRNAseq workflows, we refer the reader to Refs. 91 and 93–95.

B. Spatial transcriptomics technologies

More recently, technologies that profile gene expression while retaining spatial information have emerged. These technologies are collectively known as *spatial transcriptomics* (ST). The various ST technologies provide different advantages and are chosen based on experimental factors, such as the size of tissue to be assayed, the number of genes to be probed, *a priori* knowledge of target genes, cost, etc. In general, ST technologies can be divided into two broad categories: imaging-based and next-generation sequencing (NSG)-based technologies. In this section, we provide an overview of popular techniques, with more emphasis on NGS-based approaches. For a more comprehensive and technical review of ST technologies, we refer the reader to Asp *et al.*,[96] Rao *et al.*,[20] and Moses and Pachter.[97]

1. Imaging-based technologies

Imaging-based technologies are broadly subdivided into *in situ* hybridization (ISH)-based methods, *in situ* sequencing (ISS)-based methods, or methods that borrow elements from both approaches. Unlike RNAseq methods described above, ISH and most ISS-based techniques require labeled probes. This means that the target genes must be known in advance, and, moreover, the number of genes that can be measured is limited.[20]

a. In situ hybridization (ISH)-based approaches. ISH-based methods aim to detect the absence or presence of target RNA (or DNA) sequences while localizing the information of the desired sequences to specific cells or chromosomal sites.[98,99] ISH-based techniques use the labeled probes (usually made with DNA or RNA) that bind to desired sequences in fixed cells or tissue, therefore detecting the desired sequence through the hybridization of a complementary probe. The hybridized probes are then visualized through isotopic and nonisotopic (fluorescent and nonfluorescent) approaches.[99] ISH-based techniques have been limited by the number of distinguishable transcripts; however, recent innovations have resulted in ample multiplexing capabilities.[20]

b. In situ sequencing (ISS)-based approaches. ISS-based approaches aim to sequence the RNA content of a cell *in situ* using DNA balls that amplify the RNA signals: RNA is first reverse transcribed to cDNA,

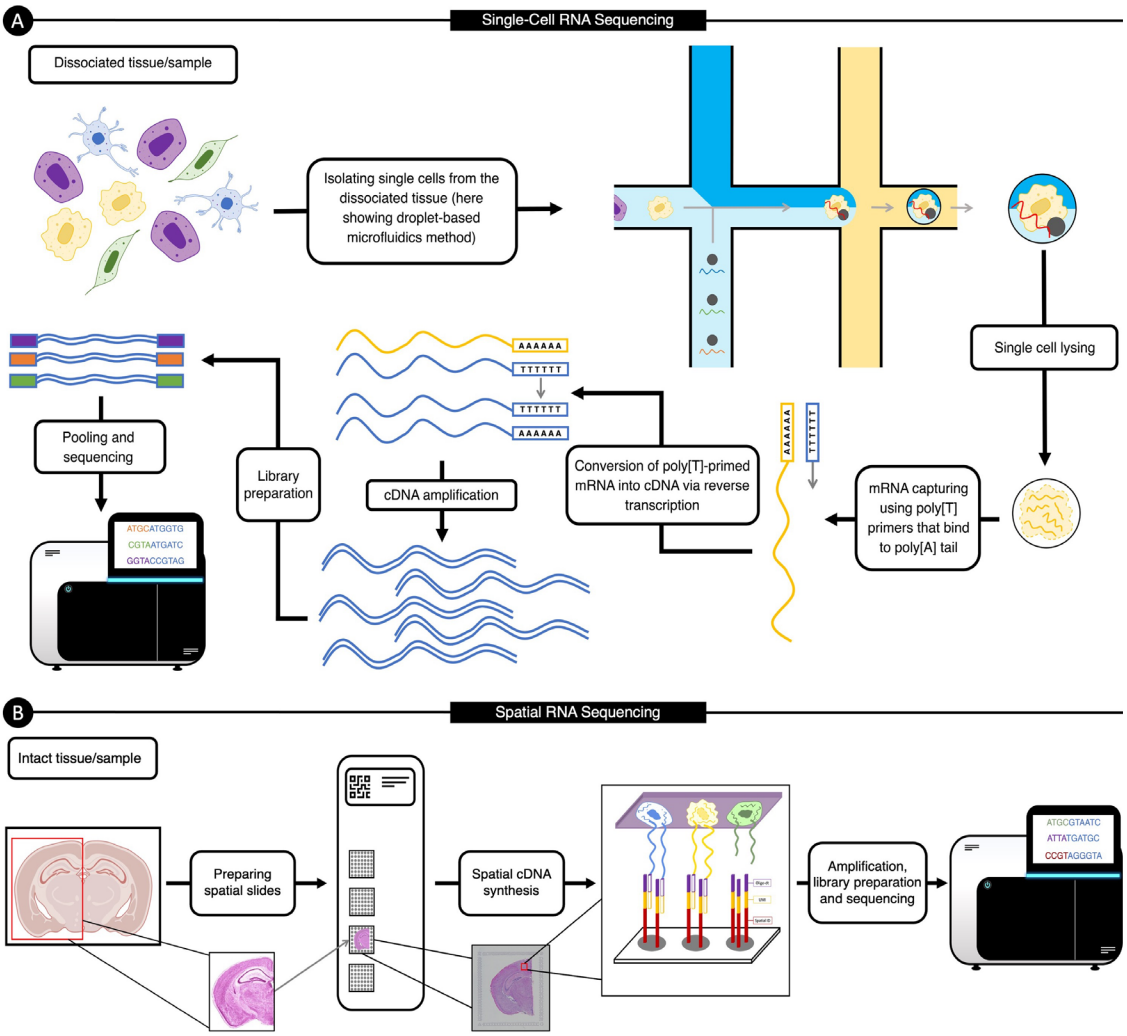

FIG. 2. Example of single-cell RNA sequencing and spatial transcriptomics workflows. (A) An illustration of droplet-based microfluidics single-cell RNA sequencing that consists of (1) dissociating a tissue or biological sample, (2) isolating single cells, unique molecular identifiers, and lysis buffer, (3) cell lysis, (4) mRNA capture and reverse transcription, (5) cDNA amplification, and (6) library construction. We describe this process in more detail in Sec. II A). (B) A visualization of the steps for next-generation sequencing-based spatial transcriptomic (as described in Sec. II B), which include (1) tissue preparation, staining, and imaging, (2) permeabilizing the tissue, (3) cDNA synthesis, amplification, and library construction, followed by (5) sequencing.

followed by circular amplification (to generate a long single-stranded DNA that contains repeats of the target sequence) and sequencing.[100] Although the transcript can be localized at subcellular resolution, micrometer- or nanometer-sized DNA balls are often used to amplify the signals to reach sufficient signal for imaging.[96] Initially, the first ISS-based method[101] used targeted padlock probes (a single-stranded DNA molecule containing regions complementary to the target cDNA) followed by sequence-by-ligation[21] to detect desired genes. This method provided a subcellular resolution and an ability to detect single-nucleotide variants (SNVs). This ISS protocol is targeted and yields a detection efficiency of approximately 30%.[102] Several recent ISS techniques build upon this approach and improve traditional ISS by introducing novel modifications to the experimental aspect of the protocol, as well as ways to mitigate the number of cells that can be discriminated simultaneously.[103–105] For example, *barcode in situ targeted sequencing* (BaristaSeq)[104] uses sequencing-by-synthesis and has led to increased read lengths and enabled higher throughput and cellular barcoding with improved detection efficiency compared to the initial ISS approach.[104] Another ISS-based technique is spatially resolved transcript amplicon readout mapping (STARmap)[29] that reduces noise and avoids the cDNA conversion complications by utilizing the improved padlock-probe and primer design; STARmap adds a second primer to target the site next to the padlock probe in order to circumvent the reverse transcription step. STARmap also uses advanced hydrogel chemistry and takes advantage of an error-robust sequencing-by-ligation method, resulting in detection efficiency that is comparable to scRNAseq methods (around 40%).[29,96] Although most ISS approaches (including the ones mentioned here) are targeted,

ISS-based methods could also be untargeted,[25,106] but this typically leads to much lower sensitivity (around 0.005%) and molecular crowding, affecting the rolling-circle amplification bias.[106,107]

In imaging-based approaches, the generated image is segmented and processed to produce a cell-level gene-expression matrix. The gene-expression matrix is generated through processing the generated image(s), which can be done manually or automatically. However, given the biased and laborious nature of manual segmentation, there has been a shift toward designing general and automated techniques.[108,109] The accurate and general automation of this process still remains a challenge, therefore motivating the application of recent machine learning and computer vision approaches to this field,[108–111] which have shown improvements compared to the traditional methods.[112] Although this manuscript focuses on methods for NGS-based technologies, many of those techniques (including the ones in Secs. IV C and IV D) can be extended to image-based technologies as well.

C. Next generation sequencing (NGS)-based technologies

Due to the unbiased capture of mRNA, NGS-based technologies can shed light on the known and unknown morphological features using only the molecular characterization of tissues.[1] This unbiased and untargeted nature of NGS technologies makes them ideal for studying and exploring new systems,[20] a major advantage compared to most image-based technologies that require target genes *a priori*. While NGS-based approaches differ in the specifics of the protocols, they all build on the idea of adding spatial barcodes before library preparation, which are then used to map transcripts back to the appropriate positions (known as spots or voxels). An example workflow of NGS-based spatial sequencing is depicted in Fig. 2(B). In this section, we provide a general overview of the four most common spatial transcriptomics technologies. For a more complete review of these technologies, we refer the readers to Refs. 20 and 96.

Ståhl *et al.*[31] were the first to successfully demonstrate the feasibility of using NGS for spatial transcriptomics (this initial approach is often referred to as *spatial transcriptomics*). Their innovation was to add spatial barcodes prior to library preparation, enabling the mapping of expressions to appropriate spatial spots. More specifically, Ståhl *et al.* positioned oligo(dT) probes and unique spatial barcodes as microarrays of spots on the surface of slides. Next, fresh frozen tissue slices were placed on the microarray and processed to release mRNA (using enzymatic permeabilization), which was then hybridized with the probes on the surface of the slides. This approach consists of (i) collecting histological imaging [using standard fixation and staining techniques, including hematoxylin and eosin (H and E) staining] for investigating the morphological characteristics and (ii) sequencing the spatially barcoded cDNA to profile gene expressions. In the initial experiments, each slide consisted of approximately 1000 spots, each having a diameter of 100 μm with 200 μm center-to-center distance.[1] This approach provides researchers with an unbiased technique for analyzing large tissue areas without the need for selecting target genes in advance.[20,35,112]

After the initial success of *spatial transcriptomics*, 10× Genomics subsequently improved the resolution (shrinking the spot diameters to 55 μm with 100 μm center-to-center distance) and sensitivity (capturing more than 10^4 transcripts per spot) of the approach and eventually commercializing it as Visium[30,113]) The development and commercialization of the spatial transcriptomics resulted in relatively rapid adoption across fields, such as cancer biology,[55,114] developmental biology,[115,116] and neuroscience.[117,118] The histological imaging and gene expression profiling of Visium are similar to the initial approach: the staining and imaging of the tissues are through traditional staining techniques, including H and E staining for visualizing the tissue sections using a brightfield microscope and immunofluorescence staining to visualize protein detection in the tissue sections through a fluorescent microscope. Visium protocol allows for both fresh frozen (FF) or formalin-fixed paraffin-embedded (FFPE) tissues. For FF tissues, similar to Ståhl *et al.*,[31] the tissue is permeabilized, allowing the release of mRNA, which hybridizes to the spatially barcoded oligonucleotides present on the spots. The captured mRNA then goes through a reverse transcription process that results in cDNA, which is then barcoded and pooled for generating a library.[119] For FFPE tissues, tissue is permeabilized to release ligated probe pairs from the cells that bind to the spatial barcodes on the slide, and the barcoded molecules are pooled for downstream processing to library generation.[119]

Building on the *spatial transcriptomics*, Vickovic *et al.*[33] proposed high-definition spatial transcriptomics (HDST) that improved the resolution to about 2 μm. Similar to the other approaches, HDST also employs specific barcodes ligated to beads that are coupled to a spot (prior to lysis), so that expressions are mapped to the tissue image. However, the innovation of HDST includes the use of 2 μm beads placed in hexagonal wells, enabling accurate compartmentalization and grouping of the biological materials in the experiment.[33] Simultaneously, Rodriques *et al.*[32] introduced SlideSeq that utilizes slides with randomly barcoded beads to capture mRNA, also increasing the resolution (to 10 μm) and sensitivity (500 transcripts per 54 bead) spatial-resolved sequencing compared to Ståhl *et al.*[31] However, SlideSeq placed the barcoded beads in rubber and onto glass slides, as opposed to HDST's hexagonal beads, and determines the position of each random barcode by *in situ*-indexing.[20,32]

Despite the differences, all NGS-based technologies use spatial barcodes to tag released RNAs, which then go through conventional processes for sequencing similar to scRNAseq. After sequencing, the data are processed to construct the spatial location of each read (using the spatial barcode) and to construct a gene-expression matrix (mapping the reads to the genome to identify the transcript of origin). Given that most technologies have resolutions larger than a single cell (commonly having expression for 3–30 cells in each spot), the data processing and analysis procedures are relatively similar.

III. MACHINE LEARNING AND DEEP LEARNING BACKGROUND

With the technologies now defined, we next describe common *Machine Learning* (ML) methods used to analyze the ST data. In this section, we first provide a discussion of the algorithmic development of ML and *deep learning* (DL) models and then discuss common architectures used for spatially resolved transcriptomics (and scRNAseq data).

ML refers to a computer algorithm's ability to acquire knowledge by extracting patterns and features from the raw data.[120] All ML algorithms depend on the data, which must be available before the methods can be used, and a defined mathematical objective. ML models' lifecycle consists of two phases, namely, *training* and *evaluation*.

During training, ML algorithms analyze the data to extract patterns and adjust their internal parameters based on optimizing their objectives (known as *loss function*). In the evaluation (or inference) stage, the trained model makes predictions (or performs the task it was trained to do) on *unseen* data.

There are two main types of ML algorithms: *supervised* and *unsupervised*. An ML algorithm is considered to be *unsupervised* if it utilizes raw inputs without any labels to optimize its objective function (an example would be the K-Means clustering algorithm[121]). Conversely, if an algorithm uses both raw data and the associated labels (or targets) in training, then it is a *supervised* learning algorithm. Supervised learning is the most common form of ML.[39] An example of supervised learning in scRNAseq analysis would be classifying cell subpopulations using prior annotations: this requires a labeled set of cell types for training (the available annotations), an objective function for calculating learning statistics ("teaching" the model), and testing data for measuring how well the model can predict the cell type (label) on data it has not seen before (i.e., generalizability of the model). Another common example of supervised learning is regression, where a model predicts continuous values as opposed to outputting labels or categorical values in classification. For supervised tasks, a model is trained on the majority of the data (known as *training set*) and then evaluated on held-out data (*test set*). Depending on the size of our dataset, there can also be a third data split known as a *validation set*, which is used to measure the performance of the model throughout training to determine *early stopping*.[122] Early stopping is when we decide to stop the training of a model due to overfitting (or over-optimization) on the training set. Overfitting training data worsens the generalizability of the model on unseen data, which early stopping aims to avoid.[122] In addition to supervised and unsupervised algorithms, there are also *semi-supervised* learning, where a model uses a mix of both supervised and unsupervised tasks, and *self-supervised*, where the computer algorithm generates new or additional labels to improve its training or to learn a new task.

Raw experimental data typically contains noise or other unwanted features, which present many challenges for ML algorithms. Therefore, it is often necessary to carefully preprocess data or to rely on domain-specific expertise in order to transform the raw data into some internal representation from which ML models can learn.[39] Deep learning (DL) algorithms, however, aim to use only raw data to automatically extract and construct useful representations required for learning the tasks at hand. In a broad sense, DL models are able to learn from observations through constructing a hierarchy of concepts, where each concept is defined by its relation to simpler concepts. A graphical representation of the hierarchy of concepts (and learning) will consist of many layers, with many nodes and edges connecting the vertices, somewhat resembling humans' neural network. This graph is referred to as an artificial neural network (ANN). ANNs are composed of interconnected nodes ("artificial neurons") that resemble and mimic our brains' neuronal functions. An ANN is considered to be a DL model if it consists of many layers—often more than three, hence being called *deep*.

Many tasks that humans perform can be viewed as mappings between sets of inputs and outputs. For example, humans can take a snapshot image of their surroundings (input) and detect the relevant objects (the outputs). DL, and more generally, Artificial Intelligence, aims to learn such mappings in order to model human-level intelligence. Mathematically, ANNs are universal function approximators, meaning that, theoretically, they can approximate any (continuous) function.[123–125] Cybenko[123] proved this result for a one-layer neural network with an arbitrary number of neurons (nodes) and a sigmoid activation function by showing that such architecture is dense within the space of continuous functions (this result has now been extended to ANNs with multiple layers[124]). While constructing arbitrarily-long single-layer ANNs is not possible, it has been shown that ANNs with many layers (deeper) generally learn faster and more reliably than ANNs with few wide (many neurons) layers.[126] This has allowed researchers to employ deep networks for learning very complex functions through constructing simple non-linear layers that can transform the representation of each module (starting with the raw input) into a representation at a higher, slightly more abstract level.[39]

DL models' ability to approximate highly non-linear functions has revolutionized many domains of science, including computer vision,[127–129] natural language processing,[130–132] and bioinformatics.[133–135] DL is becoming increasingly incorporated in many computational pipelines and studies, especially in genomics and bioinformatics, including scRNAseq and spatial transcriptomics analysis. In Secs. III A–III F, we provide a brief overview of essential deep learning architectures that have been used in spatial transcriptomics and scRNAseq analysis. In Fig. 3, we present illustrations of the architectures discussed in Secs. III A–III F. Note that for simplicity, we have categorized all graph convolution networks (GCNs)[136] as DL models; this is because (i) GCNs can easily be extended to include more layers (deeper networks) and (ii) lack of other existing methods that incorporate some elements of DL. A more comprehensive description of each architecture can be found in the seminal textbook by Goodfellow *et al.*[120]

A. Feed forward neural network (FFNN)

FFNNs, the quintessential example of Artificial Neural Networks (ANNs), aim to approximate a function mapping a set of inputs to their corresponding targets [see Fig. 3(a)]. More specifically, given an input $\mathbf{x} \in \mathbb{R}^n$ and a target $\mathbf{y} \in \mathbb{R}^m$, where $n, m \in \mathbb{R}$, FFNNs aim to *learn the optimal parameters* $\boldsymbol{\theta}$ *such that* $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta})$. FFNNs are the building blocks of many more advanced architectures (e.g., convolutional neural networks) and, therefore, are of paramount importance in the field of ML.[120] As mentioned previously, ANNs are universal function approximators, and they represent a directed acyclic graph of function compositions hierarchy within the network. Each layer of an FFNN, $f^{(i)}(\mathbf{x}; \boldsymbol{\theta})$ ($i \in \mathbb{N}$ being the ith layer), is often a simple linear function: For example, we can have a linear function for outputting $y \in \mathbb{R}$ of the form Eq. (1), with weight parameters $\mathbf{w} \in \mathbb{R}^n$ and a bias $b \in \mathbb{R}$

$$y = f^{(1)}(\mathbf{x}; \boldsymbol{\theta}) = f^{(1)}(\mathbf{x}; \mathbf{w}, b) = \mathbf{x}^T\mathbf{w} + b. \tag{1}$$

However, a model composed of *only* linear functions can *only* approximate linear mappings. As such, we must consider *non-linear activation* functions to increase model capacity, enabling the approximation of complex non-linear functions. In the simplest case, neural networks (NNs) use an affine transform (controlled by learned parameters) followed by a non-linear activation function, which, theoretically, enables them to approximate any non-linear

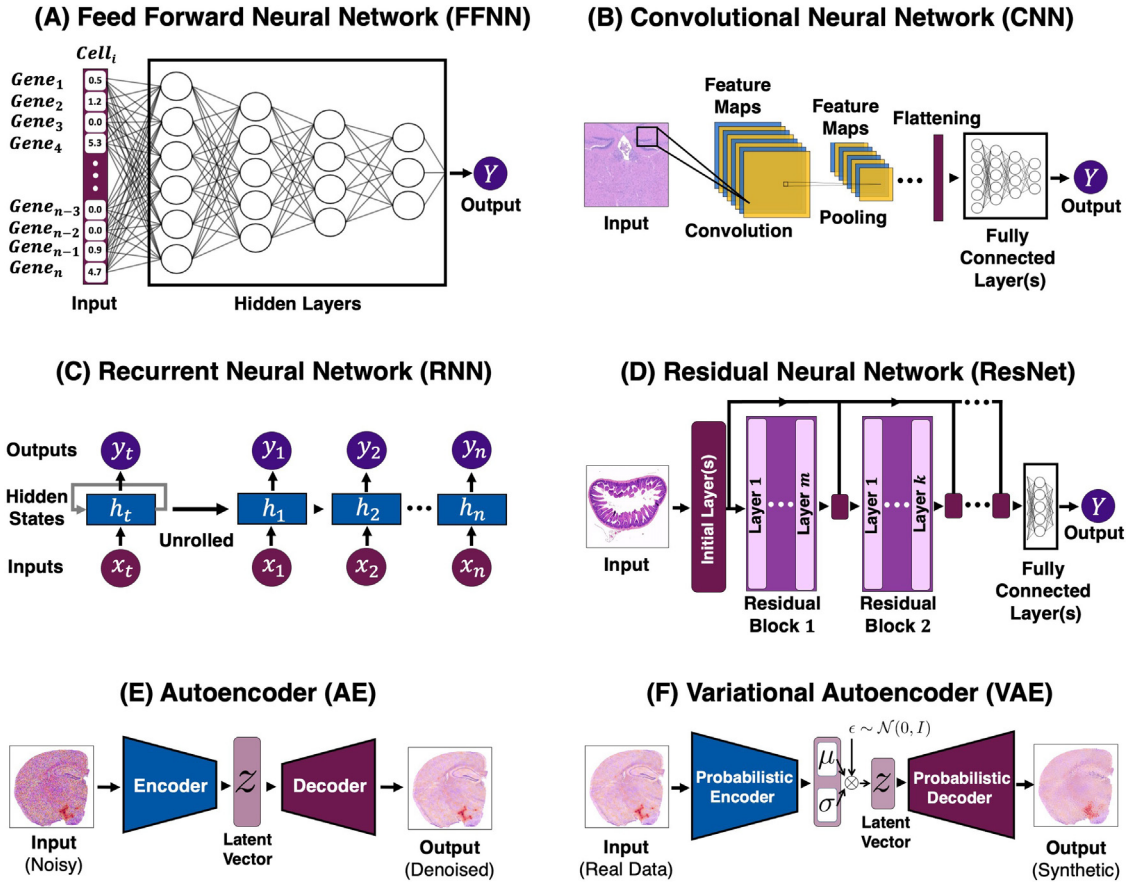

FIG. 3. Examples of deep learning architectures. Models depicted in (a), (b), (c), and (d) are examples of supervised learning, and networks shown in (e) and (f) are unsupervised. (a) An example of an FFNN architecture with gene expression count as its input. (b) An example of CNN architecture, where the model passes the inputs through the three stages of a CNN (with non-linear activation not depicted) to extract features. Then, outputs are flattened and fed into a fully connected layer (or layers). (c) The general training flow of an RNN, with the unrolled version showing the time step-dependent inputs, hidden state, and outputs. The inputs to RNNs need to have a sequential structure (e.g., time-series data). (d) An illustration of an ResNet. In traditional ResNets, there are identity mappings (or skip connections) that pass the input of a residual block to its output (often through addition). (e) Here, we show the general architecture of a *trained* denoising AE in the inference stage, with a noisy histology slide as its input, yielding a denoised version of the input image. (6) A depiction of a traditional VAE in the inference stage. VAE's aim is to generate synthetic data that closely resembles the original input. This is done through regularizing the latent space of an AE with the use of a probabilistic encoder and decoder.

function.[137] Moreover, we could compose many of such non-linear transformations to avoid infinitely wide-neural networks when approximating complex functions. However, in this context, finding a set of optimal functions $f^{(i)}: \mathbb{R}^{q_i} \to \mathbb{R}^{d_i}$ $(q_i, d_i \in \mathbb{R})$ is a practically impossible task. As such, we restrict the class of function that we use for $f^{(i)}$ to the following form in Eq. (2):

$$f^{(i)}(\mathbf{x}^{(i-1)}; \boldsymbol{\theta}^{(i)}) = \sigma^{(i)}(W^{(i)}\mathbf{x}^{(i-1)} + \mathbf{b}^{(i)}), \tag{2}$$

where superscript i enumerates the layers, $\sigma(\cdot)$ is a non-linear activation function (usually a Rectified Linear Unit[138]), $\mathbf{x}^{(i-1)} \in \mathbb{R}^{q_i}$ denotes the output of the layer $(i-1)$ (with $\mathbf{x}^{(0)}$ indicating the input data), weights $W \in \mathbb{R}^{d_i \times q_i}$, and biases $\mathbf{b}^{(i)} \in \mathbb{R}^{d_i}$. Note that because of the dimensionality of the mapping, $W^{(i)}X^{(i-1)} \in \mathbb{R}^{d_i}$, and we must have a vector of biases $\mathbf{b}^{(i)} \in \mathbb{R}^{d_i}$. FFNNs are composed of such functions in chains; to illustrate, consider a three-layer neural network

$$\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta}), \tag{3a}$$

$$= f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x}; \boldsymbol{\theta}^{(1)}); \boldsymbol{\theta}^{(2)}); \boldsymbol{\theta}^{(3)}) \tag{3b}$$

$$= f^{(3)}\left(h^{(2)}\left(h^{(1)}\left(\mathbf{x}; \mathbf{w}^{(1)}, \mathbf{b}^{(1)}\right); \mathbf{w}^{(2)}, \mathbf{b}^{(2)}\right); \mathbf{w}^{(3)}, \mathbf{b}^{(3)}\right) \tag{3c}$$

with h representing the *hidden states or hidden layers.*

FFNNs find the optimal contribution of each parameter (i.e., weights and biases) by minimizing the desired objective. The goal is to generalize the task to data the model has never seen before (testing data). Although the non-linearity increases the capacity of FFNNs, it causes most objective functions to become non-convex. In contrast to convex optimization, non-convex loss functions do not have global convergence guarantees and are sensitive to the initial starting point (parameters of the network).[139] Therefore, such optimization is often done through stochastic gradient descent (SGD) (or some variant of it). Moreover, given the sensitivity to initial values, weights are

typically chosen to be small random values, with biases initialized to zero or small positive values.[120,140,141]

Almost all neural networks use iterative gradient descent (GD)-based optimizers to train. GD has three main variants, which differ in the amount of data utilized to calculate the gradients for updating the parameters. The classic GD variant, referred to as *batch* GD, uses all data points to make the updates to the parameters in *one iteration*. However, this approach is generally not feasible, since the amount of data required for training DL models almost never fit in memory.[142] The second variant of GD is *stochastic gradient descent* (SGD) where the parameters are updated for every training datum. Computationally, it has been shown that the noise in SGD accelerates its convergence compared to batch GD, but SGD also has the possibility of overshooting, especially for highly non-convex optimization functions.[142,143] The third variant, and the most frequently used one for deep learning, is mini-batch GD that updates the parameters for every batch of training data—if the batch size is one, then this variant is just SGD, and if batch size is the entire dataset, then it is equivalent to batch GD. Conventionally, optimization of NNs is done through gradient descent performed backward in the network, which itself consists of two components: a numerical algorithm for efficient computation of the chain rule for derivatives (backpropagation[144]) and a GD-based optimizer (e.g., Adam[145] or AdaGrad[146]). The optimizer is an algorithm that performs gradient descent, while backpropagation is an algorithm for computing the expression for gradients during the backward pass of the model.

B. Convolutional neural network (CNN)

Learning from images, such as detecting edges and identifying objects, has been of interest for some time in computer science.[147] Images contain a lot of information; however, only a small amount of that information is often relevant to the task at hand. For example, an image of a stained tissue contains both important information, namely, the tissue itself, and irrelevant pixels, such as the background. Prior to DL, researchers would hand-design a feature extractor to learn relevant information from the input. Much of the work had focused on the appropriate feature extractors for desired tasks (e.g., see the seminal work by Marr and Hildreth[148]). However, one of the main goals in ML is to extract features from raw inputs without hand-tuned kernels for feature extraction. CNNs[149,150] are a specialized subset of ANNs that use the convolution operation (in at least one of their layers) to learn appropriate kernels for extracting important features beneficial to the task at hand. Mathematically, convolution between two functions f and w is defined as a commutative operation shown in the following equation:

$$(f * w)(x) \triangleq \int_{-\infty}^{\infty} f(s)w(s-x)ds. \tag{4}$$

Using our notation, we intuitively view convolution as the area under $f(s)$ weighted by $w(-s)$ and shifted by x. In most applications, discrete functions are used. As an example, assume we have a 2D kernel K that can detect edges in a 2D image I with dimension $m \times n$. Since I is discrete, we can use the discrete form of Eq. (4) for the convolution of I and K over all pixels

$$E(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n). \tag{5}$$

However, since there is less variation in the valid range of m and n (the dimensions of the image) and the operation is commutative, most algorithms implement Eq. (5) equivalently

$$E(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n). \tag{6}$$

Typical CNNs consist of a sequence of layers (usually three) which include a layer performing convolution, hence called a *convolutional layer* (affine transform), a detector stage (non-linear transformation), and a pooling layer. The learning unit of a convolutional layer is called a *filter or kernel*. Each convolutional filter is a matrix, typically of small dimensions (e.g., 3×3 pixels), composed of a set of weights that acts as an object detector, with the weights being continuously calibrated during the learning process. CNNs' objective is to learn an optimal set of filters (weights) that can detect the needed features for specific tasks (e.g., image classification). The result of convolution between the input data and the filter's weights is often referred to as a *feature map* [as shown in Fig. 3(b)]. Once a feature map is available, each value of this map is passed through a non-linearity (e.g., ReLU). The output of a convolutional layer consists of as many stacked feature maps as the number of filters present within the layer.

There are two key ideas behind the design of CNNs: First, in data with grid-like topology, local neighbors have highly correlated information. Second, equivariance to translation can be obtained if units at different locations *share weights*. In other words, sharing parameters in CNNs enabled the detection of features regardless of the locations where they appear in. An example of this would be detecting a car. In a dataset, a car could appear at any position in a 2D image, but the network should be able to detect it regardless of the specific coordinates.[147] These design choices provide CNNs with three main benefits compared to other ANNs: (i) sparse interactions, (ii) shared weights, and (iii) equivariant representations.[149]

Another way of achieving equivariance to translation is to utilize pooling layers. Pooling decreases the dimension of learned representations and makes the model insensitive to small shifts and distortions.[39] In the pooling layers, we use the outputs of the detector stage (at certain locations) to calculate a summary statistic for a rectangular window of values (e.g., calculating the mean of a 3×3 patch). There are many pooling operations, with common choices being max-pooling (taking the maximum value of a rectangular neighborhood), mean-pooling (taking the average), and L2 norm (taking the norm). In all cases, rectangular patches from one or several feature maps are inputted to the pooling layer, where semantically similar features are merged into one. CNNs typically have an ensemble of stacked convolution layers, non-linearity, and pooling layers, followed by fully connected layers that produce the final output of the network. The backpropagation of gradients through CNNs is analogous to FFNNs, enabling the model to learn an optimal set of filters for the task(s) at hand. CNNs have been effectively used in many applications in computer vision and time-series analysis and are being increasingly utilized for analysis of the ST data, since spatial-omics are multi-modal, with one of the modalities being images (as we discuss in Sec. IV).

C. Recurrent neural network (RNN)

Just as CNNs are specialized to process data with a grid-like topology, RNNs'[144] special characteristics make them ideal for processing sequential data $X = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}$, where $\mathbf{x}^{(i)}$ denotes the

ith element in the ordered sequence X. Examples of such sequence-like structures include time series and natural language. RNNs process sequential inputs one at a time and implicitly maintain a history of previous elements of the input sequence. We present an illustration of the conventional RNN architecture in Fig. 3(c). Similar to FFNNs or CNNs, RNNs can be composed of many layers, with each layer depending on the previous hidden state, $h^{(t-1)}$, and a shared set of parameters, $\boldsymbol{\theta}$. A deep RNN with n hidden states can be expressed as follows:

$$h^{(n)} = f(\mathbf{x}^{(n)}, h^{(n-1)}; \boldsymbol{\theta}); \boldsymbol{\theta}), \tag{7a}$$

$$= f(\mathbf{x}^{(n)}, f(\mathbf{x}^{(n-1)}, h^{(n-2)}; \boldsymbol{\theta}); \boldsymbol{\theta}) \tag{7b}$$

$$= f(f(\cdots f(\mathbf{x}^{(2)}, h^{(1)}(\mathbf{x}^{(1)}; \theta); \boldsymbol{\theta}) \cdots; \boldsymbol{\theta}); \boldsymbol{\theta}). \tag{7c}$$

The idea behind sharing $\boldsymbol{\theta}$ in RNN states is similar to CNNs: parameter sharing across different time points allows RNNs to generalize the model to sequences of variable lengths and share statistical strengths at different positions in time.[120,151] Similar to FFNNs, RNNs learn by propagating the gradients of each hidden state's inputs at discrete times. This process becomes more intuitive if we consider the outputs of hidden units at various time iterations as if they were the outputs of different neurons in a deep multi-layer network. However, due to the sequential nature of RNNs, the backpropagation of gradients shrinks or grows at each time step, causing the gradients to potentially vanish or blow up. This fact and the inability to parallelize training at different hidden states (due to the sequential nature of RNNs) makes RNNs notoriously hard to train, especially for longer sequences.[130,152] However, when these issues are averted (via gradient clipping or other techniques), RNNs are powerful models and gain state-of-the-art capabilities in many domains, such as natural language processing. The training challenges combined with the nature of scRNAseq data have resulted in fewer developments of RNNs for single-cell analysis. However, recently, some studies have used RNNs and long short-term memory[153] (a variant of RNNs) for predicting cell types and cell motility (e.g., see Kimmel *et al.*[154]).

D. Residual neural network (RestNet)

As mentioned above, deep RNNs may suffer from vanishing or exploding gradients. Such issues can also arise in other deep neural networks as well, where gradient information could diminish as the depth increases (through approaches, such as batch normalization,[155] aim to help with gradient issues). One way to alleviate vanishing gradients in very deep networks is to allow gradient information from successive layers to pass through, helping with maintaining information propagation even as networks become deeper. ResNets[156] achieve this by *skip* (or *residual*) connections that add the input to a block (a collection of sequential layers) to its output. For an FFNN, consider function f in Eq. (2). Using the same notation as in Eq. (2), ResNet's inner layers take the form shown in the following equation:

$$f^{(i)}(\mathbf{x}^{(i-1)}) = \mathbf{x}^{(i-1)} + \sigma^{(i)}(W^{(i)}\mathbf{x}^{(i-1)} + \mathbf{b}). \tag{8}$$

The addition of $\mathbf{x}^{(i-1)}$, the input of the current layer [or the output of the $(i-1)$ th layer], to the current ith layer output is the *skip or residual* connection helps flow the information from the input deeper in the network, thus stabilizing training and avoiding vanishing gradient in many cases.[156,157] Indeed, this approach can be contextualized within the traditional time integration framework for dynamical systems. For example, consider the following equation:

$$\dot{x}(t) = \frac{dx}{dt} = \mathscr{F}(t, x(t)), \quad x(t_0) = x_0. \tag{9}$$

In the simplest case, this system can be discretized and advanced using $x(t_n)$ and some scaled value of $\mathscr{F}(t_n, x(t_n))$, or a combination of scaled values of $\dot{x}(t_n)$. Forward Euler, perhaps the simplest time integrator, advances the solution as shown through the scheme in the following equation:

$$x_{n+1} = x_n + h\mathscr{F}(t_n, y_n), \tag{10}$$

where h is a *sufficiently small* real positive value. ResNets use this idea to propose a different way of calculating the transformations in each layer, as shown in Eq. (8).

ResNets consist of residual blocks (also called modules), each of which contains a series of layers. For visual tasks, these blocks often consist of convolutional layers, followed by activation functions, with the skip connection adding the input information to the output of the residual blocks (as opposed to the individual layers inside). ResNets have different depths and architectures, with a number usually describing the depth of the model [e.g., ResNet50 means there are 50 layers (there are 48 convolution layers, one MaxPool, and one AveragePool layer)].

ResNets have transformed DL by enabling the training of *very* deep neural networks, setting the state-of-the-art performance in many areas, particularly in computer vision.[156] The pre-trained ResNets on ImageNet dataset[158] are widely used for transfer learning, where the network is either used as is or further fine-tuned on the specific dataset. Pre-trained ResNet models have also been used in spatial transcriptomics analysis, as we discuss later in this manuscript. (ImageNet is the standard dataset for benchmarking performance of machine learning algorithms in classification and object recognition. ImageNet contains more than 14 million hand-annotated images).

E. Autoencoder (AE)

AEs[159,160] are neural networks that aim to reconstruct (or copy) the original input via a *non-trivial mapping*. Conventional AEs have an "hour-glass" architecture [see Fig. 3(e)] consisting of two networks: (i) an encoder network, $Enc(\cdot)$, which maps an input $\mathbf{x} \in \mathbb{R}^n$ to a latent vector $\mathbf{z} \in \mathbb{R}^d$ where, ideally, $\mathbf{z}$ contains the most important information from $\mathbf{x}$ in a reduced space (i.e., $d \ll n$), (ii) the decoder network, $Dec(\cdot)$, which takes $\mathbf{z}$ as the input and maps it back to $\mathbb{R}^n$, ideally, reconstructing $\mathbf{x}$ exactly; i.e., $\mathbf{x} = AE(\mathbf{x}) = Dec(Enc(\mathbf{x}))$. AEs were traditionally used for dimensionality reduction and denoising, trained by minimizing a mean squared error (MSE) objective between the input data and the reconstructed samples (outputs of the decoder).

Over time, the AE framework has been generalized to stochastic mappings, i.e., probabilistic encoder–decoder mappings, $p_{Enc}(\mathbf{z}|\mathbf{x})$ and $p_{Dec}(\mathbf{x}|\mathbf{z})$. A well-known example of such generalization is variational autoencoders (VAEs),[161] where by using the same hour-glass architecture, one can use probabilistic encoders and decoders to generate new samples drawn from an approximated posterior. Both traditional AEs and VAEs have practical applications in many biological fields, have been used extensively in scRNAseq (see Ref. 10 for an overview of these models), and are

becoming more frequently employed in spatial transcriptomics analysis, which we overview later in this work.

F. Variational autoencoder (VAE)

One can describe VAEs[161] as AEs that regularize the encoding distribution, enabling the model to generate new synthetic data. The general idea behind VAEs is to encode the inputs as a *distribution over the latent space*, as opposed to a single point (which is done by AEs). More specifically, VAEs draw samples $\mathbf{z}$ from an encoding distribution, $p_{model}(\mathbf{z})$, and subsequently feed the sample through a differentiable generator network, obtaining $Gen(\mathbf{z})$. Then, $\mathbf{x}$ is sampled from a distribution $p_{model}(\mathbf{x}; Gen(\mathbf{z})) = p_{model}(\mathbf{x}|\mathbf{z})$. Moreover, VAEs utilize an approximate inference network $q(\mathbf{z}|\mathbf{x})$ (i.e., the encoder) to obtain $\mathbf{z}$. With this approach, $p_{model}(\mathbf{x}|\mathbf{z})$ now is considered a decoder network, decoding $\mathbf{z}$ that comes from $q(\mathbf{z}|\mathbf{x})$. VAEs can take advantage of gradient-based optimization for training through *maximizing the variational lower bound*, $\mathscr{L}$, associated with $\mathbf{x}$. Figure 3(f) depicts the architecture of traditional VAEs.

Mathematically, we can express the objective function as in Eq. (11)

$$\mathscr{L}(q) = \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})} \log p_{model}(\mathbf{z}, \mathbf{x}) + \mathrm{H}(q(\mathbf{z}|\mathbf{x})), \tag{11a}$$

$$= \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})} \log p_{model}(\mathbf{x}|\mathbf{z}) - \mathbb{KL}(q(\mathbf{z}|\mathbf{x})||p_{model}(\mathbf{z})) \tag{11b}$$

$$\leq \log p_{model}(\mathbf{x}), \tag{11c}$$

where $\mathscr{H}(\cdot)$ denotes the entropy, and $\mathbb{KL}$ is the Kullback–Leibler divergence. The first term in Eq. (11a) is the joint log-likelihood of the hidden and visible variables under the approximate posteriors over the latent variables. The second term of Eq. (11a) is the entropy of the approximate posterior. This entropy term encourages the variational posterior to increase the probability mass on a range of $\mathbf{z}$ which could have produced $\mathbf{x}$, as opposed to mapping to a one-point estimate of the most likely value.[120]

Compared to other generative models [e.g., generative adversarial networks (GANs)[162]], VAEs have desirable mathematical properties and training stability.[120] However, they suffer from two major weaknesses: (i) classic VAEs create "blurry" samples (those that adhere to an average of the data points), rather than the sharp samples that GANs generate due to GANs' adversarial training. (ii) The other major issue with VAEs is posterior collapse: when the variational posterior and actual posterior are nearly identical to the prior (or collapse to the prior), which results in poor data generation quality.[163] To alleviate these issues, different algorithms have been developed, which have been shown to significantly improve the quality of data generation.[163–169] VAEs are used extensively for the analysis of single-cell RNA sequencing (see Erfanian *et al.*[10]), and we anticipate them to be applied to a wide range of spatial transcriptomics analyses as well.

IV. DEEP LEARNING MODELS FOR SPATIALLY RESOLVED TRANSCRIPTOMICS ANALYSIS

In Secs. IV A–IV E, we describe the use of ML and DL to problems emerging from spatial transcriptomics.

A. Spatial reconstruction

Prior to the advancement of spatial transcriptomics, several studies aimed to reconstruct spatial information using gene expression data, with most of these works using a statistical framework. As perhaps one of the most influential models in this space, Satija *et al.*[59] introduced Seurat: a tool that utilized spatial reference maps constructed from a few landmark *in situ* patterns to infer the spatial location of cells from corresponding gene expression profiles (i.e., scRNAseq data). This approach showed promising results: Satija *et al.* tested Seurat's capabilities and performance on developing zebrafish embryo dataset (containing 851 cells) and a reference atlas constructed from colorigenic *in situ* data for 47 genes,[59] confirming Seurat's accuracy with several experimental assays. Additionally, they showed that Seurat can accurately identify and localize rare cell populations. Satija *et al.* also demonstrated that Seurat was a feasible computational solution for handling stochastic noise in omics data and finding a correspondence between ST and scRNAseq data.

Although Seurat proved to be successful in some applications, it had the limitation of requiring spatial patterns of marker genes expression.[60] To alleviate Seurat's limitations, newer methods that did not require spatial reference atlases were developed. A more recent and an influential model in this space is *novoSpaRc*,[60] with the ability to infer spatial mappings of single cells *de novo*. For novoSpaRc, Nitzan *et al.*[60] assume that cells that are closer to one another physically have similar gene expressions as well, therefore searching for spatial arrangement possibilities that place cells with similar expressions closer in space. Nitzan *et al.* formulate this search through a generalized optimal-transport problem for probabilistic embedding.

NovoSpaRc shows very promising results when it is applied to spatially reconstruct mammalian liver and intestinal epithelium, and embryos from fly and zebrafish from gene expression data.[60] However, novoSpaRc (and similar models) use a generic framework and cannot be easily adapted to specific biological systems, which may be required given the vast diversity of biological processes and organisms. For this reason, many have utilized ML algorithms to specifically adapt to the biological system by learning from the data, as opposed to using pre-defined algorithms that remain unchanged. Indeed, we anticipate that DL models will soon play a salient role in spatial reconstruction of scRNAseq, given their ability to extract features from the raw data while remaining flexible across different applications. In this section, we review DEEPsc,[61] a system-adaptive ML model that aims to impute spatial information onto non-spatial scRNAseq data.

DEEPsc is a spatial reconstruction method that requires a reference atlas (see Fig. 4). This reference map can be expressed as a matrix $M_{spatial} \in \mathbb{R}^{n_{positions} \times n_{genes}}$, where $n_{positions} \in \mathbb{N}$ is the number of spatial locations, and $n_{genes} \in \mathbb{N}$ is the number of genes. Maseda *et al.* start by selecting common genes between $M_{spatial}$ and the gene expression matrix, $M_{expression} \in \mathbb{R}^{m_{cells} \times m_{genes}}$ (where $m_{cells} \in \mathbb{N}$ is the number of cells, and $m_{genes} \in \mathbb{N}$ is the number of genes), resulting in a spatial matrix $S \in \mathbb{R}^{n_{positions} \times g}$ and an expression matrix $E \in \mathbb{R}^{m_{cells} \times g}$, where $g \in \mathbb{N}$ is the common genes between the two matrices. Next, S is projected into a lower dimension using principal component analysis (PCA), and the same PCA coefficients are used to project E into these principal components. In the last step of processing, both matrices are normalized by their largest elements, resulting all elements of the matrices E and S to be in [0, 1]. Let us denote the normalized and PCA-reduced spatial and gene expression matrices as $\tilde{S}$ and $\tilde{E}$, respectively.

DEEPsc requires known spatial expression to learn the correct spatial positions, given the gene expression. More specifically, Maseda *et al.* construct training vectors of size $Inp_{ij} = [pos_i; pos_j] \in \mathbb{R}^{2N}$ (with N being the number of features preserved in the reduced matrix $\tilde{S}$).

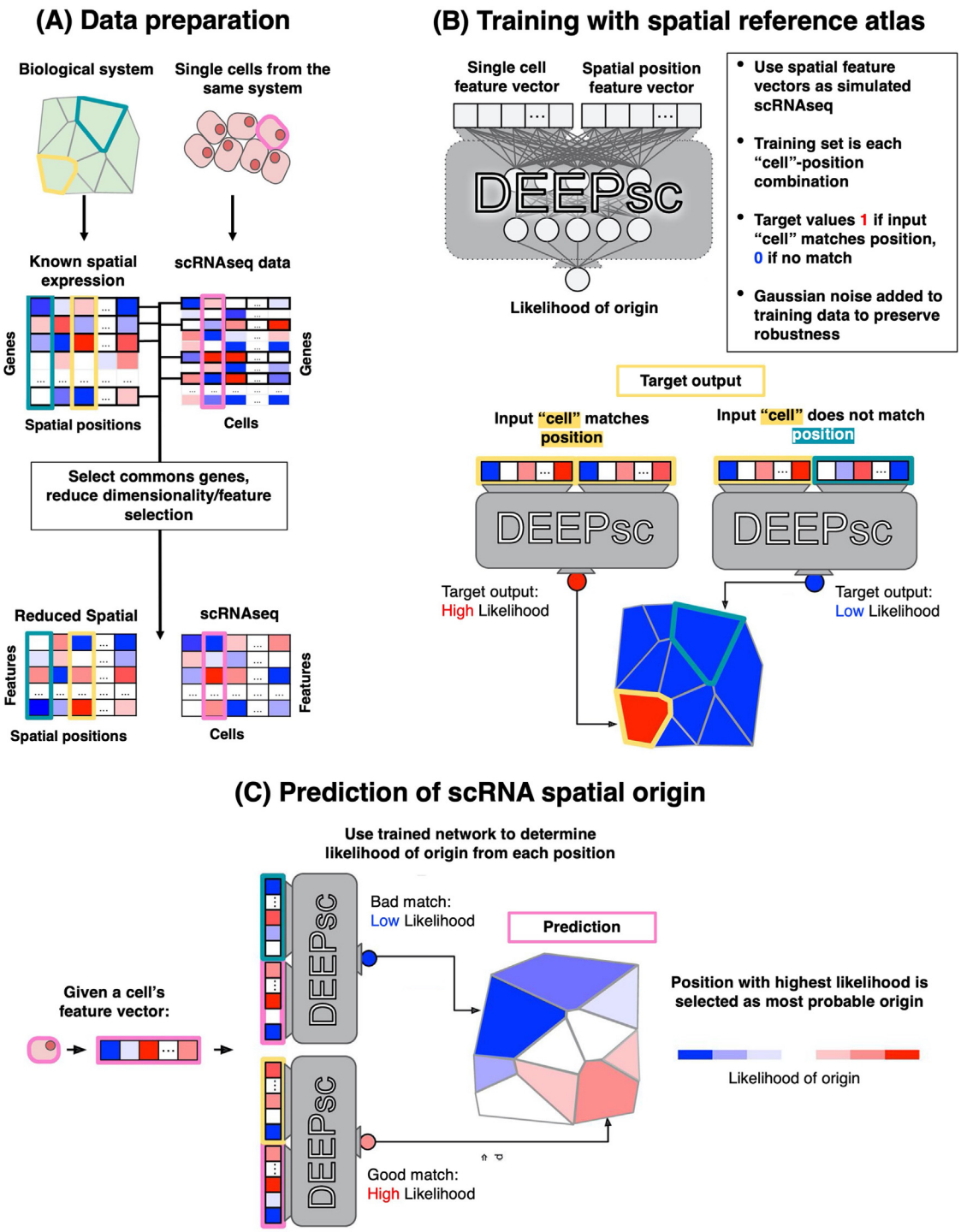

FIG. 4. An overview of DEEPsc training and inference. (a) Maseda *et al.* find the common genes in both spatial and scRNAseq data and perform dimensionality reduction on each data modality (with the final matrices having the same number of features). (b) During the training, DEEPsc uses spatial expression to "simulate" single-cell gene expression vectors. More specifically, every feature vector from the spatial expression is concatenated with all other vectors (labeled as "non-match") and also itself (labeled as "match") to form the input data to the neural network. (c) During inference, the scRNAseq feature vectors are concatenated with all spatial feature vectors, where the model should place a high probability for locations where the gene expression could have originated from. This figure was obtained from Maseda *et al.*[61] [Front. Genet. **12**, 348 (2021). Copyright 2021 Authors; licensed under a Creative Commons Attribution [CC-BY] license).

The first N elements of Inp_{ij} correspond to the spatial expression at the ith position, and the last N elements correspond to some position j in the reference atlas, including the position $j = i$. During training, DEEPsc's goal is produce the highest likelihood when $j = i$ (meaning that $Inp_{ij} = [pos_i; pos_i]$) and assign low likelihood when $j \neq i$. DEEPsc also adds Gaussian noise to pos_i (the first N elements of Inp_{ij}), which aims to preserve robustness and avoid overfitting. The addition of noise can lead to DEEPsc learning a complex non-linear mapping between the spatial positions in the reference atlas rather than a simple step-like function that activates when an exact match is inputted. During inference stage (i.e., after DEEPsc is trained), pos_i is replaced with the gene expression feature vector, which are the elements of $\tilde{E}$, and the goal is to predict the likelihood of the expression vector being originated from all possible positions j.

DEEPsc's network is an FFNN with two hidden layers, with each $h^{(1),(2)} \in \mathbb{R}^N$, mapping to an $y \in [0, 1]$, where y can be viewed as a likelihood that the input cell originated from the input spatial position.[170] Given that for each training data Inp_{ij} there will be $n_{positions} - 1$ non-matches (labels of zero) and only one match for when $j = i$, the training labels will have many more zeros than ones. Therefore, Maseda *et al.* propose a nontraditional objective function that accounts for the imbalance between zeros and ones in the training data. This objective function is shown in the following equation:

$$\mathcal{L}(Y^{true}, Y^{pred}) = \sum_{i=1}^{p} \frac{(y_i^{true} - y_i^{pred})}{1.001 - y_i^{true}}, \tag{12}$$

where y_i^{pred} is the networks predicted outputs, and y_i^{true} is the true target ($y_i^{true} = 1$ if it exactly matches, and $y_i^{true} = 0$ otherwise). This allows DEEPsc to avoid producing trivially zero outputs, which is important given the sparsity of the data. Maseda *et al.* also employ strategies in data splitting which helps to account for the inherent sparsity in the targets.[61] It is important to note that Maseda *et al.* also formulate a novel system-adaptive scoring scheme to evaluate the performance of DEEPsc using the spatial reference. However, the scoring scheme does not fall within the scope of this manuscript.

Maseda *et al.* apply DEEPsc for spatial imputation of four different biological systems (Zebrafish,[59] Drosophila,[171] Cortex,[172] and Follicle[173]) achieving accuracy comparable to several existing models while having higher precision and robustness.[61] DEEPsc also shows better consistency across the different biological systems tested, which can be attributed to its system-adaptive design. In addition, the authors attribute the performance and generalizability of DEEPsc to the use of FFNN (which have been noted before in other biological applications as well[45,47]) and the various strategies for robustness used during the training of DEEPsc. On the other hand, a weakness of DEEPsc is its training time, which depends non-linearly on the number of locations available. However, this issue can be potentially mitigated by considering a small subset of possible locations, or a more optimized design when training the model.

B. Alignment

Alignment in ST analysis refers to the process of mapping scRNAseq data to a physical domain while aiming to match the geometry with the available spatial data. As previously stated, NGS-based technologies suffer from limited capture rates and significant dropout[174] (especially at higher resolutions). Before the use of DL in ST analysis, many computational approaches aimed to spatially reconstruct key marker genes scRNAseq data by assuming continuity in the gene space,[60] or by leveraging local alignment information.[59] Moreover, most techniques for alignment or deconvolution of spatial data either learned a program dictionary[32] or estimated a probabilistic distribution of the data[64] for the cell types at each spot. However, such approaches are not generalizable to all experimental settings, since finding the mapping of sparse or sporadically distributed genes to the spots is difficult and is error-prone due to dropouts.[34]

DL frameworks have the potential of providing robust models that can adapt to specific species or technologies while being generalizable to other datasets and platforms. The potential application of DL in the alignment of spatially resolved transcriptomics came to fruition recently through the work of Biancalani *et al.*, called **Tangram.**[34] Tangram is a framework that, among many of its capabilities, can align scRNAseq or single-nucleus(sn) RNAseq profiles to spatial data; for the sake of simplicity, we refer to both data types as scRNAseq, although there are differences between the two methods (see Ref. 175 for a systematic comparison of scRNAseq and snRNAseq approaches). Tangram aims to: (i) learn the transcriptome-wide spatial gene expression map at a single-cell resolution and (ii) relate the spatial information back to histological and anatomical data obtained from the same samples. Tangram's general workflow is to learn a mapping between the data modalities and then construct specific models for the downstream tasks (such as deconvolution, correcting low-quality data, etc.). We first summarize Tangram's alignment algorithm and then provide the applications in which DL models are utilized.

Tangram's general objective is to learn a spatial matrix $S \in \mathbb{R}^{n_{cells} \times n_{genes}}$ describing the spatial alignments for the cells, with n_{cells} and n_{genes} denoting the number of single-cells and the number of genes, respectively. Let the expression of gene k in cell i be denoted by $S_{ik} \in \mathbb{R}_{[0,\infty)}$, a non-negative value. Next, Tangram partitions ("voxelizes") the spatial volume at the finest possible resolution (depending on the spatial technology) as a one-dimensional array. This allows Tangram to construct (1) a matrix $G \in \mathbb{R}^{n_{voxels} \times n_{genes}}_{[0,\infty)}$, where G_{jk} is a non-negative value denoting the expression of gene k in voxel j, and (2) a cell-density vector $\mathbf{v} = \{v_1, v_2, \ldots, v_{n_{voxels}}\}$, where $0 \leq v_j \leq 1$ is the cell density in voxel j (with the total density for each voxel summing to 1).

The learning of transcriptome-wide spatial gene expression map at a single-resolution happens through learning a mapping operator $M \in \mathbb{R}^{n_{cells} \times n_{voxels}}_{[0,1]}$, where M_{ij} denotes the probability of cell i being in voxel j. Moreover, given any matrix $\tilde{M} \in \mathbb{R}^{n_{cells} \times n_{voxels}}$, each element of the operator M is assigned according the following equation:

$$M_{ij} = \frac{e^{\tilde{M}_{i,j}}}{\sum_{q=0}^{n_{voxels}} e^{\tilde{M}_{q,j}}} \tag{13}$$

ensuring that $\sum_{j=1}^{n_{voxels}} M_{ij} = 1$, i.e., assigning a probability distribution along the voxels using the well-known $softmax(\cdot)$ function. Biancalani *et al.* define an additional quantity, $M^T S$, which denotes the spatial gene expression as predicted by the operator M, and a vector $\mathbf{m} = \{m_1, \ldots, m_{n_{cells}}\}$ where $m_j = \sum_i^{n_{voxels}} \frac{M_{ij}}{n_{cells}}$ is the *predicted* cell density for each voxel j.

Given the preliminary quantities, we can now write Tangram's generic objective function as shown in the following equation:

$$\mathscr{L}(S,M) = \sum_{k=1}^{n_{genes}} \cos_{sim}\left((M^T S)_{[:],k}, G_{[:],k}\right), \tag{14}$$

where "[:]" denotes the matrix slicing, and $\cos_{sim}$ is the cosine similarity, defined as follows:

$$\cos_{sim}(\mathbf{a},\mathbf{b}) \triangleq \frac{a \cdot b}{||a||||b||}. \tag{15}$$

The objective function, Eq. (14), learns a proportional mapping of the genes to the voxels. Additionally, this loss function can be further modified to incorporate prior knowledge. Indeed, Biancalani *et al.* modify this to regularize for the learned density distributions and the cells contained within each voxel, as shown in the following equation:

$$\mathscr{L}(S,M) = \mathbb{KL}(\mathbf{m},\mathbf{v}) - \sum_{k=1}^{n_{genes}} \cos_{sim}\left((M^T S)_{[:],k}, G_{[:],k}\right) - \sum_{j=1}^{n_{voxels}} \cos_{sim}\left((M^T S)_{j,[:]}, G_{j,[:]}\right), \tag{16}$$

where minimizing the divergence (first term) enforces that the learned density distribution and the expected distribution are similar, and the additional loss over the voxels (third term) penalizes the model if the predicted gene expression is not proportional to the expected gene expression. Biancalani *et al.* minimize the objective shown in Eq. (16) through gradient-based optimizers implemented in PyTorch.[176,177] After optimizing Eq. (16), Tangram is able to map all scRNAseq profiles onto the physical space, thus performing the alignment. It is important to note that although Tangram learns a linear operator M, this mapping could be replaced with a deep neural network as well.

Tangram utilizes DL to integrate anatomical and molecular features, specifically for mouse brain images. To do so, the authors use an image segmentation network (U-Net[178]) in combination with a "Twin" network[179] to produce segmentation masks of anatomical images, with both networks being CNNs. We present a general overview of these architectures in Fig. 5. The twin network uses DenseNet:[180] a deep NNs that concatenates the outputs at each layer to propagate salient information to deeper layers in the network (refer to Sec. III D for the motivation behind such approaches). More specifically, Tangram uses a pre-trained DenseNet encoder (trained on ImageNet) to encode images and remove technical noise and artifacts. Biancalani *et al.* also add two additional layers to the pre-trained encoder, which map the outputs to a smaller latent space. The encoder of the twin network is fine-tuned on learning the prediction of spatial depth difference between two images: Two random images are inputted to the twin network with their spatial difference depth being the desired target output, d^{true}. The network then tries to predict the depth, d^{pred}, for all N inputs, ultimately comparing them against the corresponding true depth differences, d^{true} (as shown in the following equation):

$$MSE(d^{pred}, d^{true}) = \frac{1}{N}\sum_{i=1}^{N}\left(d_i^{pred} - d_i^{true}\right)^2. \tag{17}$$

The segmentation model of Tangram generates five custom segmentation masks (background, cortex, cerebellum, white matter, and other gray matter) that are compatible with the existing Allen ontology atlas. The segmentation model is a U-Net, which uses a pre-trained ResNet50 (Ref. 156) as its core. Finally, for each pixel in input images, the model's last layer (a softmax function) assigns a probability of belonging to one of the five segmentation classes. Tangram's segmentation model aims to optimize the superposition of the cross entropy and Jaccard index, as presented in the following equation:

$$\mathscr{L}(g,p) = -g \cdot \log(p) - \frac{p \cap g}{p \cup g} \tag{18}$$

with p denoting the model prediction and g referring to the ground truth image.

Biancalani *et al.* demonstrate that Tangram learns an accurate mapping between the spatial data and scRNAseq gene expression when applied to fine- or coarse-grained spatial atlases. The authors show that their approach works well across different technologies (namely, ISH, smFISH, Visium, STARmap, and MERFISH) at different resolutions and gene coverage and is able to learn a robust and accurate alignment mapping for the isocortex of the adult healthy mouse brain.[34] While Tangram can offer different advantages based on spatial technology, it can produce consistent spatial mappings and overcome limitations in resolution or throughput, which is beneficial in many ST experiments and studies.

C. Spot deconvolution

One downside of using NGS-based technologies remains to be their resolution: Despite the recent technological advancements, most ST platforms (e.g., *spatial transcriptomics*, Visium, DBiT-seq,[181] Nanostring GeoMx,[182] and SlideSeq) do not have a single-cell resolution. The number of cells captured in each spot still varies based on the tissues [about 1–10 (Ref. 183)] and the technology used. On the other hand, we cannot assume that all cells within a spot are the same, due to the heterogeneity of the cells. Therefore, it is necessary to use computational approaches for inferring the cell types in each spot or voxel. Such estimations would be possible if there were a complementary scRNAseq dataset. The process of inferring the cellular composition of each spot is known as *cell-type deconvolution*. Deconvolution has been at the forefront of computational efforts, and it is important in building organ atlases.[20,118,184] In fact, cell-type deconvolution is an existing procedure for inferring cell-type composition in RNAseq data using scRNAseq. However, methods developed for bulk RNAseq do not account for the spatial components of ST datasets and are therefore generally inadequate. Given that deconvolution is an existing practice in RNAseq studies, we will refer to the spatial deconvolution problem as *spot deconvolution* to distinguish between the traditional methods and the ones developed for ST analysis.

We divide spot deconvolution methods into three categories: (i) statistical methods, (ii) machine learning, and (iii) deep learning, with many of the current models falling into the first two categories. We now dive deeper into the two existing models that use DL for performing spot deconvolution.

1. DestVI

DestVI (deconvolution of spatial transcriptomics profiles using variation inference) is a Bayesian model for spot deconvolution. DestVI employs a conditional deep generative model (similar to scVI,[185] a popular model for scRNAseq analysis) to learn cell-type

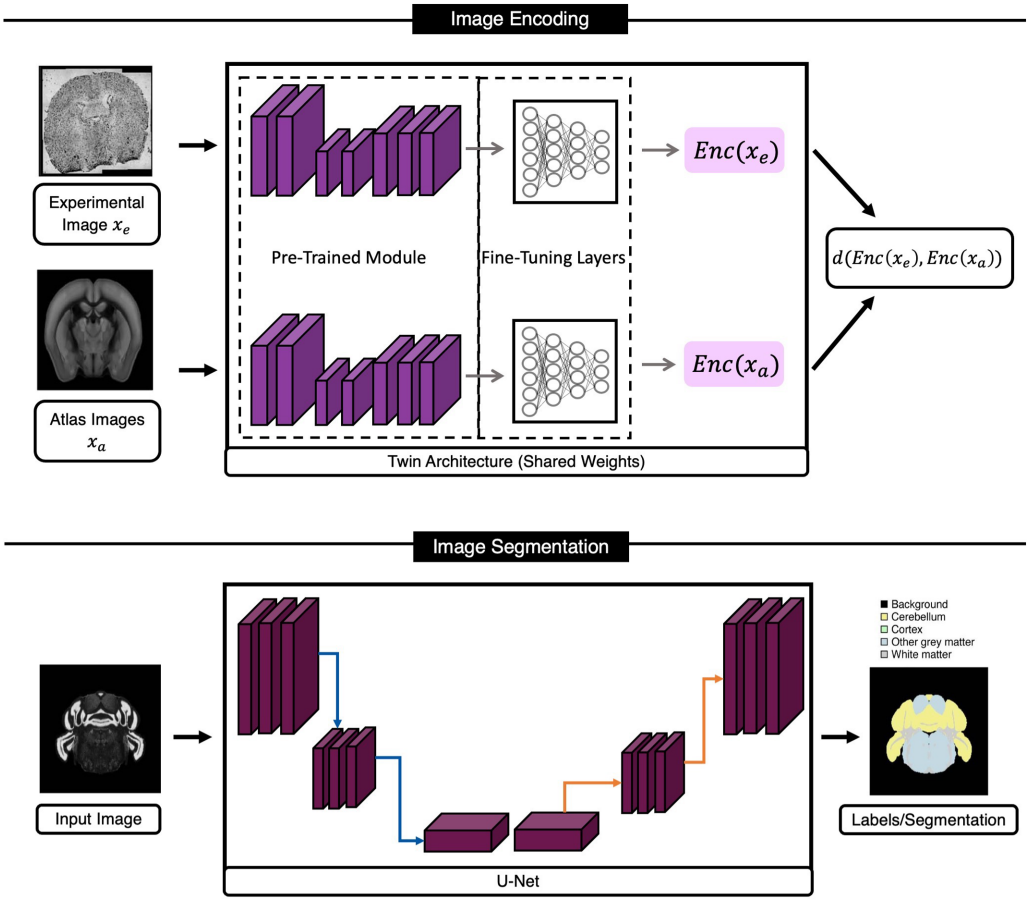

FIG. 5. Tangram's DL framework for aligning/integrating histology and anatomical data with molecular data. Tangram's model for this task is a combination of encoding (using a twin network) and segmentation modules. The twin network learns a similarity metric for brain sections based on anatomical features in images, while the U-Net model is trained to segment five different classes on mouse brain images. This figure was recreated for this manuscript based on illustrations from Biancalani *et al.*[34]

profiles and continuous sub-cell-type variations, aiming to recover the cell-type frequency and the average transcriptions state of cell types at each spot. To do so, DestVI takes a pair of transcriptomics datasets as inputs: (i) a reference scRNA-seq data and (ii) a query spatial transcriptomics data (from the same samples). DestVI then outputs the expected proportion of cell types for every spot and continuous estimation of cell state for the cell types present in each spot, which can be viewed as the average state of the cell types in each spot, which Lopez *et al.* suggest as useful for downstream analysis and formulation of biological hypotheses.[69]

DestVI uses two different latent variable models (LVMs) for distinguishing cell-type proportions and delineating cell-type-specific sub-states (shown in Fig. 6). The first LVM is for single-cell data (therefore named *scLVM*) which assumes the counts follow a negative binomial (NB) distribution, which has shown to model RNAseq count data well.[185–187] Specifically, Lopez *et al.* assume that for each gene g and cell n, the count of observed transcripts, x_{ng}, follows an NB distribution parameterized with (r_{ng}, p_g): $r_{ng} = l_n \cdot f(\gamma_n, c_n; \boldsymbol{\theta})$ is a parameter that depends on the type assigned to the cell c_n, the total number of detected molecules l_n, and a low-dimensional latent vector γ_n (which Lopez *et al.* set $\gamma_n = 5$) that describes the variability of cell-type assignment to cell c_n, and a neural network f parameters $\boldsymbol{\theta}$ (in this case, a two-layer NN). The second parameter of the NB, p_g, is optimized using variational Bayesian inference. We can summarize the assumptions for scLVM as shown in the following equation:

$$x_{ng} \sim NB(l_n f(c_n, \gamma_n), p_g) \tag{19}$$

with the latent variable $\gamma_n \sim \mathcal{N}(0, I)$. Each c_n (the annotations) are represented by a one-hot encoded vector, which is concatenated with γ to serve as the input of the NN f. Lopez *et al.* use a VAE to optimize for the marginal conditional likelihood $\log p_\theta(x_n | l_n, c_n)$.

Finally, for scLVM, a mean-field Gaussian distribution $q_\phi(\gamma | c_n, x_n)$, parameterized by another two-layer NN g, is inferred for each cell which quantifies the cell state and the associated uncertainty. The NN g takes a concatenation of (i) the gene expression vector x_n and (ii) the one-hot encoded cell annotations as its inputs. The network g outputs the mean and variance of the variational distribution for γ_n, obtained through optimizing Eq. (20)

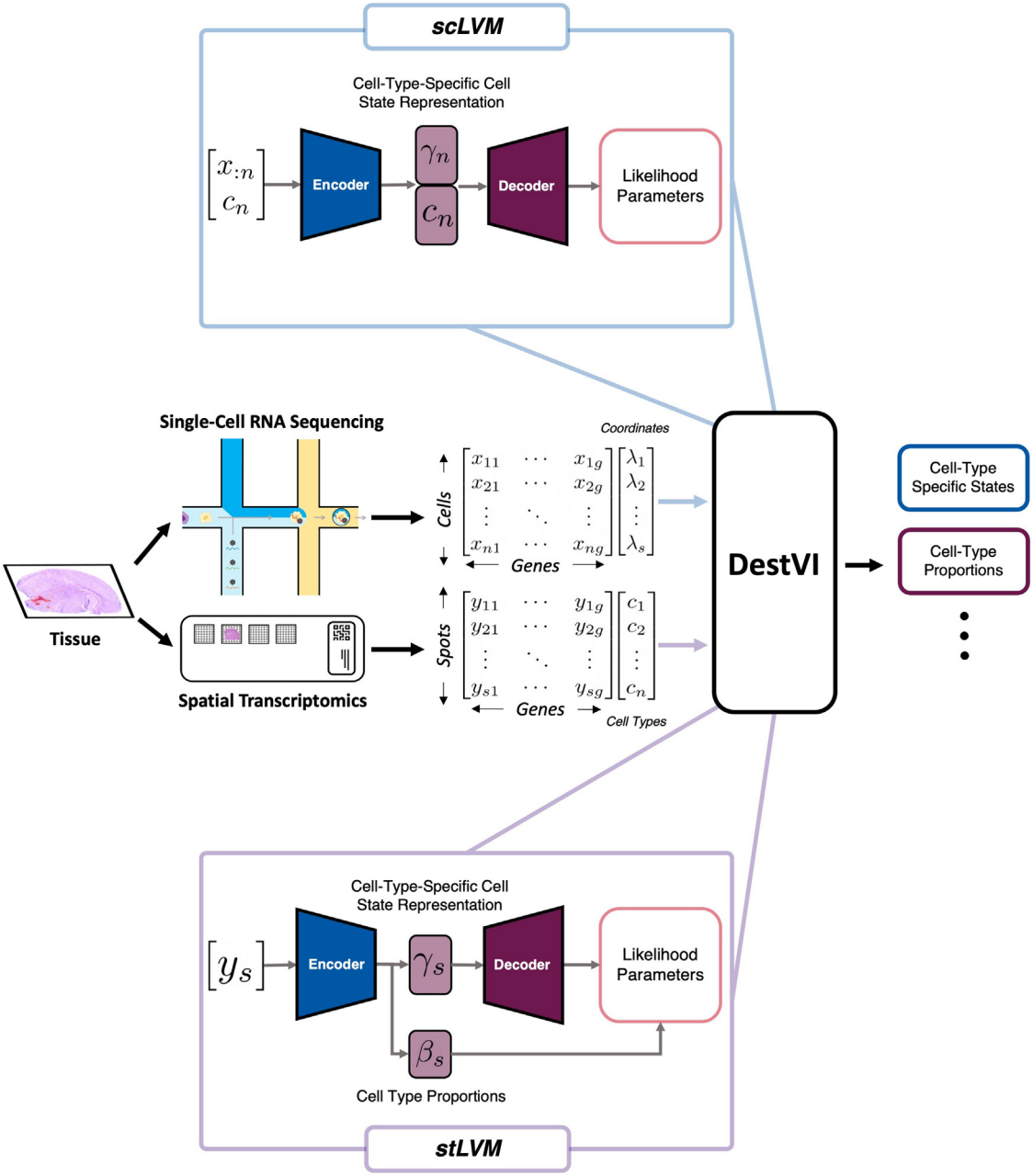

FIG. 6. Visualization of DestVI's computation workflow for spot deconvolution. DestVI uses information from both data modalities of the ST data (coordinates and scRNAseq). DestVI defines two latent variable models (LVMs) for each data modality: an LVM for modeling scRNAseq data (scLVM, shown at the top) and one that aims to model the ST data (stLVM, shown at the bottom). We describe each one in Sec. IV C. This figure was recreated for this manuscript based on illustrations from Lopez *et al.*[69]

$$\mathbb{E}_{q_\phi(\gamma_n|x_n,c_n)} \log p_{\boldsymbol{\theta}}(x_n, \gamma_n|l_n, c_n) - \mathbb{KL}(q_\phi(\gamma_n|x_n, c_n)||p_{\boldsymbol{\theta}}(\gamma_n)) \leq \log p_{\boldsymbol{\theta}}(x_n|l_n, c_n), \tag{20}$$

where $p_{\boldsymbol{\theta}}(\gamma_n)$ is the prior likelihood for γ_n. Similar to training any other VAE, the observations are split into mini-batches, and sampling from the variational distribution is done using the reparameterization trick described in Kingma and Welling.[161] The computational workflow for scLVM is visualized in Fig. 6(b).

The second LVM aims to model the spatial transcriptomics data (hence called *stLVM*) with the assumption that the number of observed transcripts x_{sg} at each spot s for each gene g follows an NB distribution. Additionally, Lopez *et al.* also assume that each spot has

$C(s)$ cells, with each cell n in a spot s being generated from the latent variables (c_{ns}, γ_{ns}). For stLVM's NB distribution, the rate parameter $r_{sg} = \alpha_g l_s f_g(c_{ns}, \gamma_{ns}; \boldsymbol{\theta}_g)$, where α_g is a correction factor for the gene-specific bias between spatial and scRNAseq data, l_g is the overall number of molecules observed in each spot, and f_g is an NN network with parameters $\boldsymbol{\theta}_g$. These assumptions and quantities allow Lopez *et al.* to formulate the total gene expression x_{sg} as shown in the following equation:

$$x_{st} \sim NB(l_s \alpha_g f_g(c_{ns}, \gamma_{ns}), p_g). \tag{21}$$

Moreover, using a parameter to designate the abundance of every cell type in every spot, β_{sc}, and NB's rate-shape parameterization property (see Aragón *et al.*[188]), Eq. (21) can be rewritten as follows:

$$x_{st} \sim NB\left(l_s \alpha_g \sum_{n=1}^{C(s)} \beta_{sc} f_g(c_{ns}, \gamma_{ns}), p_g\right) \tag{22}$$

supposing that cells from a given cell type c in a spot s must come from the same covariate γ_s^c.

The covariate γ_s^c in DestVI allows for the model to account for ST technology discrepancies by assuming various empirical priors (refer to Fig. 6). Lopez *et al.* simplify the problem of identifying every cell type in each spot to determine the density cell types, i.e., assuming that there cannot be significantly different cell states of the same cell types within a spot. Lopez *et al.* use a penalized likelihood method to infer point estimates for γ^c, α, and β. With the additional two strategies to stabilize the training of DestVI, the final objective function for stLVM consists of (i) the negative binomial likelihood, (ii) the likelihood of the empirical prior, and (iii) the variance penalization for α.

Lopez *et al.* use simulations to present DestVI's ability to provide higher resolution compared to the existing methods and estimate gene expression by every cell type in all spots. Furthermore, they show that DestVI is able to accurately deconvolute spatial organization when applied mouse tumor model. In the cases tested, Lopez *et al.* demonstrate that DestVI is capable of identifying important cell-type-specific changes in gene expression between different tissue regions or between conditions and that it can provide a high resolution and accurate spatial characterization of the cellular organization of tissues.

2. DSTG

Deconvoluting spatial transcriptomics data through the graph-based convolutional network (DSTG)[65] is a recent semi-supervised model that employs graph convolutional networks (GCN)[136] for spot deconvolution. DSTG utilizes scRNAseq to construct pseudo-ST data and then build a link graph that represents the similarity between all spots in both real and pseudo-ST data. The pseudo-ST is generated by combining scRNAseq transcriptomics of multiple cells to mimic the expression profiles at each spot; while the real ST data are unlabeled, the pseudo-ST has labels. To construct the link graph, DSTG first reduces the dimensionality of both real and pseudo data using canonical correlation analysis[189] and then identifies mutual nearest neighbors.[190] Next, a GCN is used on the link graph to propagate the real and pseudo-ST data into a latent space that is turned into a probability distribution of the cell compositions for each spot.

Song *et al.* form the link graph by taking the number of spots as the number of vertices, resulting in a graph $G = (V, E)$, where $|V|$ denotes the number of spots, and E represents the edges between them. DSTG takes two inputs: (i) the adjacency matrix of graph G, represented by A, and (ii) a combination of both real and pseudo datasets $X = [x_{pseudo}; x_{real}] \in \mathbb{R}^{m \times N}$, where m is the number of variable genes, and $N = S_{pseudo} + S_{real}$ (the total number of spots in both datasets) with S_{pseudo} and S_{real} indicating the number of spots in the pseudo and real ST datasets, respectively. Next, Song *et al.* normalize the adjacency matrix (for efficient training of DSTG) using the diagonal degree of A, denoted by D, as shown in the following equation:

$$\hat{A} = D^{-\frac{1}{2}} A_I D^{-\frac{1}{2}}, \tag{23}$$

where $A_I = A + I$ (with I denoting the identity matrix). Given the two inputs, DSTG's graph convolution layers take the following form:

$$h^{(i+1)} = \begin{cases} \sigma(\hat{A} X^T W^{(i)}), & i = 0, \\ \sigma(\hat{A} h^{(i)} W^{(i)}), & i > 0 \end{cases} \tag{24}$$

with $h^{(i)}$ denoting the ith hidden layer of DSTG, and $\sigma(\cdot)$ being a non-linear activation function [in this case, $\sigma(\cdot) = ReLU(\cdot)$]. The output of DSTG is denoted by $y_{s,t}$, the proportion of cell type $t = \{0, \ldots, T\}$ at each spot $s = \{0, \ldots, N\}$. Song *et al.* design DSTG's architecture as shown in the following equation:

$$Y_{pred} = softmax(\hat{A}\sigma(\cdots(\hat{A}\sigma(\hat{A} X^T W^{(0)}) W^{(1)}) \cdots W^{(k)}), \tag{25}$$

where k is the last layer, and $Y^{pred} = [Y^{pred}_{pseudo}; Y^{pred}_{real}] \in \mathbb{R}^{N \times T}$ is the predicted proportions at each spot in the pseudo and real data, denoted by Y_{pseudo} and Y_{real}, respectively. It is important to note that Song *et al.* chose a GCN with three layers after performing an ablation study on the number of layers. Finally, DSTG is trained by optimizing the cross entropy loss

$$\mathscr{L}(Y^{pred}_{pseudo}, Y^{true}_{pseudo}) = -\sum_{s=0}^{S_{pseudo}} \sum_{t=1}^{T} y^{true}_{s,t} \log(y^{pred}_{s,t}) \tag{26}$$

with $y^{\{true,pred\}}_{s,t}$ denoting the label for true/predicted cell type t at spot s. Note that this constitutes a semi-supervised training for DSTG since only labels for the pseudo-ST are used in training, but the model will also learn to predict labels for the real dataset as well [refer to Eq. (25)].

Song *et al.* note that, compared to traditional approaches, DSTG provides three key advantages: (i) given that DSTG uses variable genes and a non-linear GCN, it allows for learning complex deconvolution mappings from the ST data. (ii) The weights assigned to the different cell types in the pseudo-ST and the semi-supervised scheme allow DSTG to identify key features that allow the model to learn the cellular composition in real data. (iii) DSTG's scalability and adaptability will be beneficial in ST analysis, given that the sequencing depth of the ST data is expected to increase. Song *et al.* show that DSTG consistently outperforms the benchmarked state-of-the-art model (SPOTlight) on both synthetic data and real data. More specifically, DSTG is evaluated on simulated data generated from PBMC where it shows high accuracy between the predicted cell compositions and the true proportions. Song *et al.* also find that DSTG's deconvolution of the ST data from complex tissues, including mouse cortex, hippocampus, and human pancreatic tumor slices, is consistent with the underlying cellular mixtures.[65]

D. Spatial clustering

Clustering allows the aggregation of data into subpopulations based on some shared metric of distance or "closeness." In RNA sequencing studies, clustering is the first step of identifying cell clusters, often followed by laborious manual annotation (e.g., through identifying differentially expressed genes) or some automated workflows.[191] Clustering has been a crucial step in many scRNAseq studies, often performed using graph-based community detection algorithms (such as Louvain[192] or Leiden[193]) or more traditional methods (such as K-Means[121]). Although the scRNAseq techniques can be used in some ST studies (e.g., for multiplexed FISH data where a single-cell resolution is available), the result may be discontinuous or erroneous since the spatial coordinates have not been taken into account.[76] Therefore, there is a need for ST-specific methods that can utilize both gene expression and histology data to produce clusters that are coherent, both in gene expression and physical space.

Recently, new frameworks for spatial clustering of the ST data have emerged which utilize both spatial and expression information available. Zhu *et al.*[72] introduced a hidden-Markov random field (HMRF)-based method to model the spatial dependency of gene expression using both the sequencing and imaging-based single-cell transcriptomic profiling technologies. HMRF is a graph-based model used to model the spatial distribution of signals. Using the ST data, Zhu *et al.* create a grid where neighboring nodes are connected to each other. However, the spatial pattern cannot be observed directly (since it is "hidden"), and it must be inferred through observations that depend on the hidden states probabilistically. Similar to Zhu *et al.*, BayesSpace[75] employs a Bayesian formulation of HMRF and uses the Markov chain Monte Carlo (MCMC) algorithm to estimate the model parameters. Despite the ability of these methods to cluster voxels (or cells) into distinct subpopulations, these approaches suffer from the lack of versatility required to handle different modalities present in the ST data.[76]

With the emergence of newer technologies, the scale and variability within datasets are increasing, requiring more general and flexible models for accurate and robust analysis of these studies. A few ML-based approaches have been proposed to combat some of these mentioned challenges. Below, we review the ML-based methods that offer scalability and are generally more applicable to various experimental settings.

1. SpaCell

SpaCell[73] is a double-stream DL framework that utilizes both histology images and the associated spot gene counts. For the histology data, Tan *et al.* first preprocess the images (removing low-quality images, stain normalization, normalizing the pixels using a z-transform, and removing background noise). Next, they split each histology image into tiles that contain one spot each (sub-images of 299×299 pixels). For the preprocessing of the count matrix (which contains the reads at each spot), Tan *et al.* follow traditional scRNAseq preprocessing workflows, including count normalization, removal of outlier genes, and cells with too few genes. After the preprocessing stage, each tile (containing the image of a spot) corresponds to a column in the count matrix (reads from the same spot). At this point, each image $X_i \in \mathbb{R}^{299\times299}$, and the count matrix M will be in a $\mathbb{R}^{n_{spots}\times n_{genes}}$ space. However, Tan *et al.* reduce the count matrix to only contain 2048 most variable genes at each spot, therefore resulting in a new count matrix $\hat{M} \in \mathbb{R}^{n_{spots}\times 2048}$. Let us denote the ith spot of $\hat{M}$ as $\hat{m}_i \in \mathbb{R}^{2048}$, which has a corresponding image x_i.

In order to spatially cluster cells of the same type, both image and count data must be used. The first step in SpaCell is to pass on the spot images, x_i, to a pre-trained ResNet50 (trained on ImageNet data) in order to output feature vectors, $\hat{x}_i \in \mathbb{R}^{2048}$ (each having the same dimensionality as columns of $\hat{M}$). Next, to extract features from both modalities, SpaCell uses two separate AEs for the image feature vectors, $\hat{X} \in \mathbb{R}^{n_{spots}\times 2048}$, and the most-variable-genes counts, $\hat{M}$, with both AEs having the same latent dimension (we discuss the reason behind this later). Let us denote the AE for images as $AE_I(\cdot) = Dec_I(Enc_I(\cdot))$, and the gene counts AE as $AE_G(\cdot) = Dec_G(Enc_G(\cdot))$, with $Enc_{\{I,G\}}(\cdot)$ and $Dec_{\{I,G\}}(\cdot)$ indicating the encoder and decoders, respectively.

Given N spots, each AE in SpaCell aims to minimize three objective functions for their respective inputs [i.e., $\hat{x}_i$ for $AE_I(\cdot)$ and $\hat{m}_i$ for $AE_G(\cdot)$]: (i) the mean squared error (MSE) between the input and the output [shown in Eq. (27)], (ii) the KL divergence between the probability distributions for the input and constructed output of all N spots [denoted by p and q in Eq. (28), respectively], and (iii) binary cross entropy (BCE), shown in Eq. (29)

$$MSE_{\{I,G\}}(v_i, \tilde{v}_i) = \frac{1}{N}\sum_{i=1}^{N}(v_i - AE_{\{I,G\}}(v_i))^2, \tag{27}$$

$$\mathbb{KL}(p||q) = \sum_{i=1}^{N} p(v_i)\frac{\log p(v_i)}{\log q(v_i)}, \tag{28}$$

$$BCE(q) = -\frac{1}{N}\sum_{i=1}^{N}[v_i \log(p) + (1 - v_i)\log(1 - p)]. \tag{29}$$

Once training has concluded, SpaCell encodes both images and gene counts, i.e., $Enc_I(\hat{x}_i)$ and $Enc_G(\hat{m}_i)$, to be used for clustering. More specifically, clustering is performed on a matrix that is the concatenation of the latent vectors produced by each AE, $C = [Enc_I(\hat{x}_i); Enc_G(\hat{m}_i)]$. This is why the latent spaces of $AE_{I,G}(\cdot)$ have the same dimension. After obtaining the concatenated matrix, the downstream clustering is performed using K-Means (which can be substituted for other algorithms as well). Through this procedure, SpaCell uses both data modalities and can produce clusters that are highly accurate when compared to the true clusters (annotated by pathologists).

2. StLearn

StLearn[74] is a DL model that, among many of its capabilities, can perform segment (cluster) spatial regions based on their function and expression. StLearn's DL component lies within its spatial morphological gene expression (SME) normalization. The SME normalization aims to combine critical information from Hematoxylin and Eosin stained (H and E) tissue images and transcriptome-wide gene expression profiles to then take advantage of in downstream analysis, such as clustering, spatial trajectory inference, and cell–cell interaction.

The SME normalization procedure includes *(i) spatial location:* In order to use the spatial positions for selecting neighboring spot pairs, Pham *et al.* consider two spots s_i and s_j as neighbors if the center-to-center Euclidean distance between the two spots,

$|C(s_i) - C(s_j)|$, is less than a specified distance r, i.e., $|C(s_i) - C(s_j)| < r$. Pham *et al.* include all paired spots s_i and s_j as the input to adjust for the gene expression of the center spot s_i. The next step in SME normalization is *(ii) Morphological similarity:* stLearn calculates the morphological similarity between spots using feature vectors produced by an ImageNet-pre-trained ResNet50. More specifically, all H&E images corresponding to each spot s_i are inputted to the ResNet50 model, producing a feature vector $x_i \in \mathbb{R}^{2048}$. Subsequently, stLearn performs PCA on each feature vector x_i, resulting in reduced-dimension feature vectors $\hat{x}_i \in \mathbb{R}^{50}$. To calculate the morphological distance (MD) between two neighboring spots s_i and s_j [according to the criterion defined in *(i)*], Pham *et al.* measure the cosine similarity between two reduced feature vectors [refer to Eq. (15) for the definition of cosine similarity]; this MD is shown in the following equation:

$$MD(s_i, s_j) \triangleq \cos_{sim}(\hat{x}_i, \hat{x}_j). \tag{30}$$

As the last step in SME normalization, the gene expression at each spot s_i is normalized using the MD distance, as shown in the following equation:

$$\hat{GE}_i = GE_i + \frac{\sum_{j=1}^{n} \left(GE_j \cdot MD(s_i, s_j)\right)}{n}, \tag{31}$$

where GE_i denotes raw gene expression counts at spot s_i, and n is the total number of neighbors identified for spot s_i. StLearn's SME normalization enables a smoothing of the expression data based on morphological similarity with the goal of improving clustering through grouping morphologically similar sections of the tissue together.

After the SME normalization step, stLearn's clustering algorithm consists of two steps: First, finding general *global* clusters (utilizing traditional clustering techniques) using SME-normalized data as its input, followed by *local* clustering that aims to identify spatially separated sub-clusters and merge spots that are collocated, resulting in spatially contiguous clusters. Pham *et al.* note that stLearn constitutes the first-of-its-kind method to use tissue morphology in normalizing gene expression, outperforming two competing approaches on two tested data (Seurat on Allen Mouse Brain Atlas[194] and SpatialLIBD[195] on human brain tissue sections[196]).[74]

3. SpaGCN

SpaGCN[76] is a graph convolution network (GCN) that integrates both spatial information and histology images to perform spatial clustering. Using each spot as vertices, Hu *et al.* create a weighted undirected graph, $G = (V, E)$, where $|V|$ is the total number of spots, and E is the set of edges with prescribed weights representing the similarity between the nodes. The weight of each of these edges is determined by (i) the distance between the two spots (nodes) that the edge connects and (ii) the associated histology information (in this case, pixel intensity). This means that two spots are deemed similar if they are physically close to one another, *and* they seem similar in the histology image.

In order to attribute the pixel information to each spot, Hu *et al.* use the mean RGB pixel intensity of each spot within a window of size 50×50 pixels. That is, given a spot s with physical coordinates (x_s, y_s) and pixel coordinates (x_{ps}, y_{ps}), SpaGCN calculates the mean and variance of all the pixels present within a 50×50 pixels centered at (x_{ps}, y_{ps}). Let ps_r, ps_g and ps_b denote the means, and $var_r(ps)$, $var_g(ps)$, $var_b(ps)$ refer to the variance for the red, green, and blue channels, respectively. SpaGAN then summarizes the pixel mean and variance information as a unified value, as shown in the following equation:

$$z_s = \frac{(ps_r \cdot var_r(ps)) + (ps_g \cdot var_g(ps)) + (ps_b \cdot var_b(ps))}{var_r(ps) + var_g(ps) + var_b(ps)}. \tag{32}$$

Furthermore, z_s is rescaled using the mean and standard deviation of each coordinate (including the newly created z axis), with an additional scaling factor that can put more emphasis on histology data when needed. Let μ_z denote the mean of z_s, and let $\sigma_{x,y,z}$ be the standard deviation of x_s, y_s, z_s with $s \in V$; then, we can formulate the rescaling as the following:

$$\tilde{z}_s = \alpha \frac{(z_s - \mu_z)(\max\{\sigma_x, \sigma_y\})}{\sigma_z}, \tag{33}$$

where α denotes the scaling factor described previously ($\alpha = 1$ by default).

Using the rescaled value in Eq. (33), the weight of each edge between two vertices s and k is calculated as shown in the following equation:

$$w(s, k) = e^{-d(s,k)^2/(2l^2)}, \tag{34}$$

where l denotes the characteristic length scale, and $d(s, k)$ is the traditional Euclidean distance, as shown in the following equation:

$$d(s, k) = \left((x_s - x_k)^2 + (y_s - y_k)^2 + (\tilde{z}_s - \tilde{z}_k)^2\right)^{\frac{1}{2}}. \tag{35}$$

SpaGCN's network construction (and backpropagation) is similar to other GCNs, inspired by Kipf and Welling[136] (for an overview of GCNs, refer to Sec. IV C 2). The network intakes the adjacency matrix A to represent the graph G, and a reduced-dimension representation of the gene expression matrix, which Hu *et al.* achieve using PCA with 50 principal components. The output of the GCN network is a matrix that includes combined information on histology, gene expression, and spatial position. SpaGCN then uses the output of the GCN to perform unsupervised clustering of the spatial data.

SpaGCN uses the Louvain algorithm (an iterative unsupervised clustering algorithm) on the output of GCN to initialize cluster centroids, with the number of clusters (controlled by Louvain's *resolution* parameter) being optimized on maximizing the Silhouette score[197]). The iterative updates are based on optimizing a metric that defines the distance between each spot and all cluster centroids using the t-distribution as a kernel. For a centroid c_j, a total of N clusters, and the embedded point h_i for spot i, this metric can be defined as the probability of assigning cell i to cluster j, as shown in the following equation:

$$q_{ij} = \frac{(1 + h_i - \mu_j^2)^{-1}}{\sum_{c=1}^{N} (1 + h_i - \mu_c^2)^{-1}}. \tag{36}$$

Hu *et al.* further refine the clusters using an auxiliary target distribution [shown in Eq. (37)] that prefers spots assignments with the highest confidence and normalizes the centroid contribution to the overall loss function as the following:

$$p_{ij} = \frac{q_{ij}^2}{\sum_i^S q_{ij}} \cdot \left(\sum_{c=1}^N \left(\frac{q_{ic}^2}{\sum_i^S q_{ic}} \right) \right)^{-1}. \quad (37)$$

Finally, spaGCN is trained by optimizing the $\mathbb{KL}$ divergence between the p and q distributions, as shown in the following equation:

$$\mathscr{L} = \mathbb{KL}(P||Q) = \sum_{i=1}^S \sum_{j=0}^N p_{ij} \log \frac{p_{ij}}{q_{ij}}. \quad (38)$$

Hu *et al.* demonstrate that SpaGCN can accurately identify spatial clusters that are consistent with manual annotations since SpaGCN utilizes information from both gene expression and histology. The authors perform spatial clustering with SpaGCN on the human dorsolateral prefrontal cortex, human primary pancreatic cancer, and multiple mouse tissue data, showing that SpaGCN performs consistently well, outperforming other state-of-the-art models (stLearn, BayesSpace, and Louvain). These results show the feasibility and potential of SpaGCN for clustering spatial-resolved transcriptomics.

E. Cell–cell interactions

Multicellular organisms depend on intricate cell–cell interactions (CCIs) which dictate cellular development, homeostasis, and single-cell functions.[198] Unraveling such interaction within tissues can present unique insight on complex biological processes and disease pathogenesis.[198–201] CCI has been investigated using both scRNAseq and RNAseq, wherein most approaches test for enrichment in ligand–receptor profiles in the expression data.[202–204] However, the ST data can offer a more comprehensive view of CCI, since the distance traveled by ligand signal is crucial in determining the type of cell–cell signaling.[183] Given the importance of CCI and the advantages that the ST data provide, several computational approaches for inferring cellular interactions using the ST data have been developed, such as SpaOTsc,[80] Giotto,[82] MISTy.[81]

SpaOTsc is a model that can be used in integrating scRNAseq data with spatial measurements, and in inferring cellular interactions in spatial-resolved transcriptomics data. SpaOTsc aims to estimate cellular interactions by analyzing the relationships between ligand–receptor pairs and their downstream genes. SpaOTsc formulates a spatial metric using the optimal transport algorithm, returning a mapping that contains the probability distribution of each scRNA-seq cell over a spatial region. SpaOTsc also utilizes a random forest in order to infer the spatial range of ligand–receptor signaling and subsequently remove the long-distance connections. Another approach is Giotto:[82] Giotto is an extended and comprehensive toolbox designed for ST analysis and visualization, which includes a CCI model that calculates an enrichment score (the weighted mean expression of a ligand and the corresponding receptor in the two neighboring cells). Giotto then constructs an empirical null distribution by moving the locations for each cell type, subsequently calculating the corresponding statistical significance (P-value), and ordering the ligand–receptors pair-wise for all neighboring cells. Although the mentioned models have shown to discover simple cellular interactions, such approaches often fail to detect complex gene–gene interactions, which is essential in understanding many diseases. Though DL models for CCI are nascent, we anticipate such models to better learn complex interactions from the raw data.

V. CONCLUSIONS AND OUTLOOK

The ST field is rapidly growing, with new datasets and analysis pipelines released weekly. The innovations in biological methods will continue to spur creativity in algorithm development, with an emphasis on ML-based frameworks. Although the space of DL models for ST analysis is currently small, we anticipate the field to experience a paradigm shift toward deep-learned models. In this review, our goal was to provide readers with the necessary biological, mathematical, and computational background for understanding the existing approaches and expanding upon the current models to address the challenges posed by the ST domain.

In this manuscript, we provided an overview of current DL-based techniques for alignment and integration of the ST data, spatial clustering, spot deconvolution, inferring cell–cell communication, and approaches for reconstructing spatial coordinates using scRNAseq data (with limited or no spatial reference atlas). The DL methods we presented, in comparison to their conventional counterparts, offer accuracy and scalability advantages. However, DL methods are not always the preferred choice as they are computationally expensive and may lack biological interpretability. As more methods for ST analysis are developed, we believe that standard datasets for benchmarking new models as well as comprehensive accuracy and efficiency analysis of existing techniques will be of significant value to the field. Though the existing methods set the new state-of-the-art in their respective categories, room for improvements remains large. Among the ST downstream analyses, applications of DL algorithms for studying cell–cell communication and identification of spatially variable genes remain mostly underexplored. Given DL models' ability to extract sophisticated patterns from the raw data, we anticipate that DL approaches will prove useful in unraveling complex biological processes, aiding the efforts in identifying cellular interactions and highly variable genes in a spatial context.

Recent technological advancements have enabled researchers to utilize various single-cell omics sources to construct multi-omics datasets, providing comprehensive view of many diseases [e.g., COVID19 (Refs. 205 and 206) and cancer[207]] and developmental processes.[208,209] As single-cell analysis enters the multi-omics age, the need for integrating the ST data with other single-cell sources will increase. Therefore, we expect an increase in ML-based frameworks for data integration and alignment, spearheaded by DL-based approaches. Additionally, due to the noise and multi-modality of the ST data, there exists an unmet need for methods that account for batch effects in spatial and gene expression data. Given the success of DL techniques for batch effect removal in scRNAseq, we foresee DL models being widely used for batch effect correction of spatially resolved transcriptomics data.

Despite the recentness of ST technologies, researchers have successfully used these technologies to generate spatially resolved cell atlases, providing new insight on a wide range of biological processes and organs.[118,210–213] Such studies not only show the tremendous potential that ST technologies hold but also highlight the need for scalable and efficient analysis tools. The application of DL to ST analysis remains a rapidly evolving nascent domain, demonstrating promising

great prospects in advancing the field of ST, and the integration of ST datasets with other omics data.

ACKNOWLEDGMENTS

We wish to acknowledge O. Davalos for offering fruitful discussions and providing useful insight on earlier versions of this work. We also would like to thank M. Hilscher and T. Buvoli for providing valuable feedback on earlier drafts of this manuscript. A.A.H. is supported by the National Human Genome Research Institute of the National Institutes of Health under Award No. F31HG012718 (National Research Service Award). Additionally, the authors received support from the National Institutes of Health (Nos. R15-HL146779 and R01-GM126548) and the National Science Foundation (No. DMS-1840265).

The Visium slide and its visualizations in Figs. 1 and 3 were accessed from 10× Genomics.[214] The mouse brain histology image in Fig. 1 was taken from Ref. 215. The illustration of mouse brain in Fig. 2 was obtained from BioRender.[216]

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

A.A.H. and S.S.S. wrote the manuscript, edited the drafts, and conceptualized the figures, which A.A.H then created.

A. Ali Heydari: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Suzanne Sindi:** Conceptualization (equal); Data curation (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Resources (equal); Supervision (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available within the article.

REFERENCES

[1]L. Larsson, J. Frisén, and J. Lundeberg, "Spatially resolved transcriptomics adds a new dimension to genomics," Nat. Methods **18**, 15–18 (2021).
[2]J. S. Packer, Q. Zhu, C. Huynh, P. Sivaramakrishnan, E. Preston, H. Dueck, D. Stefanik, K. Tan, C. Trapnell, J. Kim, R. H. Waterston, and J. I. Murray, "A lineage-resolved molecular atlas of c. elegans embryogenesis at single-cell resolution," Science **365**, eaax1971 (2019).
[3]X. Han, R. Wang, Y. Zhou, L. Fei, H. Sun, S. Lai, A. Saadatpour, Z. Zhou, H. Chen, F. Ye, D. Huang, Y. Xu, W. Huang, M. Jiang, X. Jiang, J. Mao, Y. Chen, C. Lu, J. Xie, Q. Fang, Y. Wang, R. Yue, T. Li, H. Huang, S. H. Orkin, G.-C. Yuan, M. Chen, and G. Guo, "Mapping the mouse cell atlas by microwell-seq," Cell **172**, 1091–1107 (2018).
[4]N. Schaum, J. Karkanias, N. F. Neff, A. P. May, S. R. Quake, T. Wyss-Coray, S. Darmanis, J. Batson, O. Botvinnik, M. B. Chen, S. Chen, F. Green, R. C. Jones, A. Maynard, L. Penland, A. O. Pisco, R. V. Sit, G. M. Stanley, J. T. Webber, F. Zanini, A. S. Baghel, I. Bakerman, I. Bansal, D. Berdnik, B. Bilen, D. Brownfield, C. Cain, M. B. Chen, M. Cho, G. Cirolia, S. D. Conley, A. Demers, K. Demir, A. de Morree, T. Divita, H. du Bois, L. B. T. Dulgeroff, H. Ebadi, F. H. Espinoza, M. Fish, Q. Gan, B. M. George, A. Gillich, G. Genetiano, X. Gu, G. S. Gulati, Y. Hang, S. Hosseinzadeh, A. Huang, T. Iram, T. Isobe, F. Ives, R. C. Jones, K. S. Kao, G. Karnam, A. M. Kershner, B. M. Kiss, W. Kong, M. E. Kumar, J. Y. Lam, D. P. Lee, S. E. Lee, G. Li, Q. Li, L. Liu, A. Lo, W.-J. Lu, A. Manjunath, A. P. May, K. L. May, O. L. May, M. McKay, R. J. Metzger, M. Mignardi, D. Min, A. N. Nabhan, N. F. Neff, K. M. Ng, J. Noh, R. Patkar, W. C. Peng, R. Puccinelli, E. J. Rulifson, S. S. Sikandar, R. Sinha, R. V. Sit, K. Szade, W. Tan, C. Tato, K. Tellez, K. J. Travaglini, C. Tropini, L. Waldburger, L. J. van Weele, M. N. Wosczyna, J. Xiang, S. Xue, J. Youngyunpipatkul, M. E. Zardeneta, F. Zhang, L. Zhou, A. P. May, N. F. Neff, R. V. Sit, P. Castro, D. Croote, J. L. DeRisi, G. M. Stanley, J. T. Webber, A. S. Baghel, M. B. Chen, F. H. Espinoza, B. M. George, G. S. Gulati, A. M. Kershner, B. M. Kiss, C. S. Kuo, J. Y. Lam, B. Lehallier, A. N. Nabhan, K. M. Ng, P. K. Nguyen, E. J. Rulifson, S. S. Sikandar, S. Y. Tan, K. J. Travaglini, L. J. van Weele, B. M. Wang, M. N. Wosczyna, H. Yousef, A. P. May, S. R. Quake, G. M. Stanley, J. T. Webber, P. A. Beachy, C. K. F. Chan, B. M. George, G. S. Gulati, K. C. Huang, A. M. Kershner, B. M. Kiss, A. N. Nabhan, K. M. Ng, P. K. Nguyen, E. J. Rulifson, S. S. Sikandar, K. J. Travaglini, B. M. Wang, K. Weinberg, M. N. Wosczyna, S. M. Wu, B. A. Barres, P. A. Beachy, C. K. F. Chan, M. F. Clarke, S. K. Kim, M. A. Krasnow, M. E. Kumar, C. S. Kuo, A. P. May, R. J. Metzger, N. F. Neff, R. Nusse, P. K. Nguyen, T. A. Rando, J. Sonnenburg, B. M. Wang, I. L. Weissman, S. M. Wu, S. R. Quake, T. T. M. Consortium, O. coordination, L. coordination, O. collection, processing, L. preparation, sequencing, C. data analysis, C. type annotation, W. Group, S. text writing Group, and P. investigators, "Single-cell transcriptomics of 20 mouse organs creates a tabula muris," Nature **562**, 367–372 (2018).
[5]A. Regev, S. A. Teichmann, E. S. Lander, I. Amit, C. Benoist, E. Birney, B. Bodenmiller, P. Campbell, P. Carninci, M. Clatworthy, H. Clevers, B. Deplancke, I. Dunham, J. Eberwine, R. Eils, W. Enard, A. Farmer, L. Fugger, B. Gottgens, N. Hacohen, M. Haniffa, M. Hemberg, S. Kim, P. Klenerman, A. Kriegstein, E. Lein, S. Linnarsson, E. Lundberg, J. Lundeberg, P. Majumder, J. C. Marioni, M. Merad, M. Mhlanga, M. Nawijn, M. Netea, G. Nolan, D. Pe'er, A. Phillipakis, C. P. Ponting, S. Quake, W. Reik, O. Rozenblatt-Rosen, J. Sanes, R. Satija, T. N. Schumacher, A. Shalek, E. Shapiro, P. Sharma, J. W. Shin, O. Stegle, M. Stratton, M. J. T. Stubbington, F. J. Theis, M. Uhlen, A. van Oudenaarden, A. Wagner, F. Watt, J. Weissman, B. Wold, R. Xavier, and N. Yosef, "The human cell atlas," Elife **6**, e27041 (2017).
[6]K. Davie, J. Janssens, D. Koldere, M. De Waegeneer, U. Pech, Ł. Kreft, S. Aibar, S. Makhzami, V. Christiaens, C. Bravo González-Blas, S. Poovathingal, G. Hulselmans, K. I. Spanier, T. Moerman, B. Vanspauwen, S. Geurs, T. Voet, J. Lammertyn, B. Thienpont, S. Liu, N. Konstantinides, M. Fiers, P. Verstreken, and S. Aerts, "A single-cell transcriptome atlas of the aging drosophila brain," Cell **174**, 982–998 (2018).
[7]Y. Zhang, D. Wang, M. Peng, L. Tang, J. Ouyang, F. Xiong, C. Guo, Y. Tang, Y. Zhou, Q. Liao, X. Wu, H. Wang, J. Yu, Y. Li, X. Li, G. Li, Z. Zeng, Y. Tan, and W. Xiong, "Single-cell RNA sequencing in cancer research," J. Exp. Clin. Cancer Res. **40**, 81 (2021).
[8]A. Derakhshani, Z. Asadzadeh, H. Safarpour, P. Leone, M. A. Shadbad, A. Heydari, B. Baradaran, and V. Racanelli, "Regulation of CTLA-4 and PD-l1 expression in relapsing-remitting multiple sclerosis patients after treatment with fingolimod, IFNβ-1α, glatiramer acetate, and dimethyl fumarate drugs," J. Pers. Med. **11**, 721 (2021).
[9]A. Kusnadi, C. Ramírez-Suástegui, V. Fajardo, S. J. Chee, B. J. Meckiff, H. Simon, E. Pelosi, G. Seumois, F. Ay, P. Vijayanand, and C. H. Ottensmeier, "Severely ill patients with COVID-19 display impaired exhaustion features in SARS-CoV-2–reactive cd8$^+$ T cells," Sci. Immunol. **6**, eabe4782 (2021).
[10]N. Erfanian, A. A. Heydari, P. Iañez, A. Derakhshani, M. Ghasemigol, M. Farahpour, S. Nasseri, H. Safarpour, and A. Sahebkar, "Deep learning applications in single-cell omics data analysis," bioRxiv (2021).
[11]S. R. Park, S. Namkoong, L. Friesen, C.-S. Cho, Z. Z. Zhang, Y.-C. Chen, E. Yoon, C. H. Kim, H. Kwak, H. M. Kang, and J. H. Lee, "Single-cell transcriptome analysis of colon cancer cell response to 5-fluorouracil-induced DNA damage," Cell Rep. **32**, 108077 (2020).

[12]Y. Su, D. Chen, D. Yuan, C. Lausted, J. Choi, C. L. Dai, V. Voillet, V. R. Duvvuri, K. Scherler, P. Troisch, P. Baloni, G. Qin, B. Smith, S. A. Kornilov, C. Rostomily, A. Xu, J. Li, S. Dong, A. Rothchild, J. Zhou, K. Murray, R. Edmark, S. Hong, J. E. Heath, J. Earls, R. Zhang, J. Xie, S. Li, R. Roper, L. Jones, Y. Zhou, L. Rowen, R. Liu, S. Mackay, D. S. O'Mahony, C. R. Dale, J. A. Wallick, H. A. Algren, M. A. Zager, W. Wei, N. D. Price, S. Huang, N. Subramanian, K. Wang, A. T. Magis, J. J. Hadlock, L. Hood, A. Aderem, J. A. Bluestone, L. L. Lanier, P. D. Greenberg, R. Gottardo, M. M. Davis, J. D. Goldman, and J. R. Heath, "Multi-omics resolves a sharp disease-state shift between mild and moderate COVID-19," Cell **183**, 1479–1495 (2020).
[13]F. Iqbal, A. Lupieri, M. Aikawa, and E. Aikawa, "Harnessing single-cell RNA sequencing to better understand how diseased cells behave the way they do in cardiovascular disease," Arteriosclerosis Thrombosis Vasc. Biol. **41**, 585–600 (2021).
[14]M. Heming, X. Li, S. Rauber, A. K. Mausberg, A.-L. Borsch, M. Hartlehnert, A. Singhal, I.-N. Lu, M. Fleischer, F. Szepanowski, O. Witzke, T. Brenner, U. Dittmer, N. Yosef, C. Kleinschnitz, H. Wiendl, M. Stettner, and G. M. Z. Horste, "Neurological manifestations of COVID-19 feature t cell exhaustion and dedifferentiated monocytes in cerebrospinal fluid," Immunity **54**, 164–175 (2021).
[15]A. A. Pollen, T. J. Nowakowski, J. Shuga, X. Wang, A. A. Leyrat, J. H. Lui, N. Li, L. Szpankowski, B. Fowler, P. Chen, N. Ramalingam, G. Sun, M. Thu, M. Norris, R. Lebofsky, D. Toppani, D. W. Kemp, M. Wong, B. Clerkson, B. N. Jones, S. Wu, L. Knutsson, B. Alvarado, J. Wang, L. S. Weaver, A. P. May, R. C. Jones, M. A. Unger, A. R. Kriegstein, and J. A. A. West, "Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex," Nat. Biotechnol. **32**, 1053–1058 (2014).
[16]B. Treutlein, D. G. Brownfield, A. R. Wu, N. F. Neff, G. L. Mantalas, F. H. Espinoza, T. J. Desai, M. A. Krasnow, and S. R. Quake, "Reconstructing lineage hierarchies of the distal lung epithelium using single-cell RNA-seq," Nature **509**, 371–375 (2014).
[17]M. J. Barresi and S. F. Gilbert, *Developmental Biology* (Oxford University Press, 2019).
[18]R. Dries, J. Chen, N. Del Rossi, M. M. Khan, A. Sistig, and G.-C. Yuan, "Advances in spatial transcriptomic data analysis," Genome Res. **31**, 1706–1718 (2021).
[19]T. Noel, Q. S. Wang, A. Greka, and J. L. Marshall, "Principles of spatial transcriptomics analysis: A practical walk-through in kidney tissue," Front. Physiol. **12**, 809346 (2022).
[20]A. Rao, D. Barkley, G. S. França, and I. Yanai, "Exploring tissue architecture using spatial transcriptomics," Nature **596**, 211–220 (2021).
[21]R. Ke, M. Mignardi, A. Pacureanu, J. Svedlund, J. Botling, C. Wahlby, and M. Nilsson, "In situ sequencing for RNA analysis in preserved tissue and cells," Nat. Methods **10**, 857–860 (2013).
[22]S. Codeluppi, L. E. Borm, A. Zeisel, G. L. Manno, J. A. van Lunteren, C. I. Svensson, and S. Linnarsson, "Spatial organization of the somatosensory cortex revealed by cyclic smFISH," bioRxiv (2018).
[23]A. Raj, P. van den Bogaard, S. A. Rifkin, A. van Oudenaarden, and S. Tyagi, "Imaging individual mRNA molecules using multiple singly labeled probes," Nat. Methods **5**, 877–879 (2008).
[24]A. M. Femino, F. S. Fay, K. Fogarty, and R. H. Singer, "Visualization of single RNA transcripts in situ," Science **280**, 585–590 (1998).
[25]S. Alon, D. R. Goodwin, A. Sinha, A. T. Wassie, F. Chen, E. R. Daugharthy, Y. Bando, A. Kajita, A. G. Xue, K. Marrett, R. Prior, Y. Cui, A. C. Payne, C.-C. Yao, H.-J. Suk, R. Wang, C.-C. J. Yu, P. Tillberg, P. Reginato, N. Pak, S. Liu, S. Punthambaker, E. P. R. Iyer, R. E. Kohman, J. A. Miller, E. S. Lein, A. Lako, N. Cullen, S. Rodig, K. Helvie, D. L. Abravanel, N. Wagle, B. E. Johnson, J. Klughammer, M. Slyper, J. Waldman, J. Jané-Valbuena, O. Rozenblatt-Rosen, A. Regev, I. Consortium, G. M. Church, A. H. Marblestone, E. S. Boyden, H. R. Ali, M. A. Sa'd, S. Alon, S. Aparicio, G. Battistoni, S. Balasubramanian, R. Becker, B. Bodenmiller, E. S. Boyden, D. Bressan, A. Bruna, M. Burger, C. Caldas, M. Callari, I. G. Cannell, H. Casbolt, N. Chornay, Y. Cui, A. Dariush, K. Dinh, A. Emenari, Y. Eyal-Lubling, J. Fan, A. Fatemi, E. Fisher, E. A. González-Solares, C. González-Fernández, D. Goodwin, W. Greenwood, F. Grimaldi, G. J. Hannon, O. Harris, S. Harris, C. Jauset, J. A. Joyce, E. D. Karagiannis, T. Kovačević, L. Kuett, R. Kunes, A. K. Yoldaş, D. Lai, E. Laks, H. Lee, M. Lee, G. Lerda, Y. Li, A. McPherson, N. Millar, C. M. Mulvey, F. Nugent, C. H. O'Flanagan, M. Paez-Ribes, I. Pearsall, F. Qosaj, A. J. Roth, O. M. Rueda, T. Ruiz, K. Sawicka, L. A. Sepúlveda, S. P. Shah, A. Shea, A. Sinha, A. Smith, S. Tavaré, S. Tietscher, I. Vázquez-García, S. L. Vogl, N. A. Walton, A. T. Wassie, S. S. Watson, J. Weselak, S. A. Wild, E. Williams, J. Windhager, T. Whitmarsh, C. Xia, P. Zheng, and X. Zhuang, "Expansion sequencing: Spatially precise in situ transcriptomics in intact biological systems," Science **371**, eaax2656 (2021).
[26]S. Codeluppi, L. E. Borm, A. Zeisel, G. La Manno, J. A. van Lunteren, C. I. Svensson, and S. Linnarsson, "Spatial organization of the somatosensory cortex revealed by osmFISH," Nat. Methods **15**, 932–935 (2018).
[27]K. H. Chen, A. N. Boettiger, J. R. Moffitt, S. Wang, and X. Zhuang, "Spatially resolved, highly multiplexed RNA profiling in single cells," Science **348**, aaa6090 (2015).
[28]C.-H. L. Eng, M. Lawson, Q. Zhu, R. Dries, N. Koulena, Y. Takei, J. Yun, C. Cronin, C. Karp, G.-C. Yuan, and L. Cai, "Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH+," Nature **568**, 235–239 (2019).
[29]X. Wang, W. E. Allen, M. A. Wright, E. L. Sylwestrak, N. Samusik, S. Vesuna, K. Evans, C. Liu, C. Ramakrishnan, J. Liu, G. P. Nolan, F.-A. Bava, and K. Deisseroth, "Three-dimensional intact-tissue sequencing of single-cell transcriptional states," Science **361**, eaat5691 (2018).
[30]10x Genomics, see https://www.10xgenomics.com/spatial-transcriptomics for "Spatial transcriptomics," 2021.
[31]P. L. Ståhl, F. Salmén, S. Vickovic, A. Lundmark, J. F. Navarro, J. Magnusson, S. Giacomello, M. Asp, J. O. Westholm, M. Huss, A. Mollbrink, S. Linnarsson, S. Codeluppi, Å. Borg, F. Pontén, P. I. Costea, P. Sahlén, J. Mulder, O. Bergmann, J. Lundeberg, and J. Frisén, "Visualization and analysis of gene expression in tissue sections by spatial transcriptomics," Science **353**, 78–82 (2016).
[32]S. G. Rodriques, R. R. Stickels, A. Goeva, C. A. Martin, E. Murray, C. R. Vanderburg, J. Welch, L. M. Chen, F. Chen, and E. Z. Macosko, "Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution," Science **363**, 1463–1467 (2019).
[33]S. Vickovic, G. Eraslan, F. Salmén, J. Klughammer, L. Stenbeck, D. Schapiro, T. Aijo, R. Bonneau, L. Bergenstråhle, J. Navarro, J. Gould, G. K. Griffin, Å. Borg, M. Ronaghi, J. Frisén, J. Lundeberg, A. Regev, and P. L. Ståhl, "High-definition spatial transcriptomics for in situ tissue profiling," Nat. Methods **16**, 987–990 (2019).
[34]T. Biancalani, G. Scalia, L. Buffoni, R. Avasthi, Z. Lu, A. Sanger, N. Tokcan, C. R. Vanderburg, Å. Segerstolpe, M. Zhang, I. Avraham-Davidi, S. Vickovic, M. Nitzan, S. Ma, A. Subramanian, M. Lipinski, J. Buenrostro, N. B. Brown, D. Fanelli, X. Zhuang, E. Z. Macosko, and A. Regev, "Deep learning and alignment of spatially resolved single-cell transcriptomes with tangram," Nat. Methods **18**, 1352–1362 (2021).
[35]F. Salmén, P. L. Ståhl, A. Mollbrink, J. Navarro, S. Vickovic, J. Frisén, and J. Lundeberg, "Barcoded solid-phase RNA capture for spatial transcriptomics profiling in mammalian tissue sections," Nat. Protoc. **13**, 2501–2534 (2018).
[36]A. Jemt, F. Salmén, A. Lundmark, A. Mollbrink, J. Fernández Navarro, P. L. Ståhl, T. Yucel-Lindberg, and J. Lundeberg, "An automated approach to prepare tissue-derived spatially barcoded RNA-sequencing libraries," Sci. Rep. **6**, 37137 (2016).
[37]N. J. Tustison, P. A. Cook, A. Klein, G. Song, S. R. Das, J. T. Duda, B. M. Kandel, N. van Strien, J. R. Stone, J. C. Gee, and B. B. Avants, "Large-scale evaluation of ANTs and FreeSurfer cortical thickness measurements," NeuroImage **99**, 166–179 (2014).
[38]G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "VoxelMorph: A learning framework for deformable medical image registration," IEEE Trans. Med. Imaging **38**, 1788–1800 (2019).
[39]Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**, 436–444 (2015).
[40]G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis, "Single-cell RNA-seq denoising using a deep count autoencoder," Nat. Commun. **10**, 390 (2019).
[41]M. B. Badsha, R. Li, B. Liu, Y. I. Li, M. Xian, N. E. Banovich, and A. Q. Fu, "Imputation of single-cell gene expression with an autoencoder neural network," Quant. Biol. **8**, 78–94 (2020).

[42]X. Li, K. Wang, Y. Lyu, H. Pan, J. Zhang, D. Stambolian, K. Susztak, M. P. Reilly, G. Hu, and M. Li, "Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis," Nat. Commun. **11**, 2338 (2020).
[43]J. Hu, X. Li, G. Hu, Y. Lyu, K. Susztak, and M. Li, "Iterative transfer learning with neural network for clustering and cell type classification in single-cell RNA-seq analysis," Nat. Mach. Intell. **2**, 607–618 (2020).
[44]X. Shao, H. Yang, X. Zhuang, J. Liao, P. Yang, J. Cheng, X. Lu, H. Chen, and X. Fan, "scDEEPSort: A pre-trained cell-type annotation method for single-cell transcriptomics using deep learning with a weighted graph neural network," Nucl. Acids Res. **49**, e122 (2021).
[45]F. Ma and M. Pellegrini, "ACTINN: Automated identification of cell types in single cell RNA sequencing," Bioinformatics **36**, 533–538 (2020).
[46]A. A. Heydari, O. A. Davalos, L. Zhao, K. K. Hoyer, and S. S. Sindi, "ACTIVA: Realistic single-cell RNA-seq generation with automatic cell-type identification using introspective variational autoencoders," Bioinformatics **38**, 2194–2201 (2022).
[47]M. Marouf, P. Machart, V. Bansal, C. Kilian, D. S. Magruder, C. F. Krebs, and S. Bonn, "Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks," Nat. Commun. **11**, 166 (2020).
[48]K. Bayoudh, R. Knani, F. Hamdaoui, and A. Mtibaa, "A survey on deep multimodal learning for computer vision: Advances, trends, applications, and datasets," Visual Comput. **38**, 2939–2970 (2021).
[49]D. Butler, C. Mozsary, C. Meydan, J. Foox, J. Rosiene, A. Shaiber, D. Danko, E. Afshinnekoo, M. MacKay, F. J. Sedlazeck, N. A. Ivanov, M. Sierra, D. Pohle, M. Zietz, U. Gisladottir, V. Ramlall, E. T. Sholle, E. J. Schenck, C. D. Westover, C. Hassan, K. Ryon, B. Young, C. Bhattacharya, D. L. Ng, A. C. Granados, Y. A. Santos, V. Servellita, S. Federman, P. Ruggiero, A. Fungtammasan, C.-S. Chin, N. M. Pearson, B. W. Langhorst, N. A. Tanner, Y. Kim, J. W. Reeves, T. D. Hether, S. E. Warren, M. Bailey, J. Gawrys, D. Meleshko, D. Xu, M. Couto-Rodriguez, D. Nagy-Szakal, J. Barrows, H. Wells, N. B. O'Hara, J. A. Rosenfeld, Y. Chen, P. A. D. Steel, A. J. Shemesh, J. Xiang, J. Thierry-Mieg, D. Thierry-Mieg, A. Iftner, D. Bezdan, E. Sanchez, T. R. Campion, J. Sipley, L. Cong, A. Craney, P. Velu, A. M. Melnick, S. Shapira, I. Hajirasouliha, A. Borczuk, T. Iftner, M. Salvatore, M. Loda, L. F. Westblade, M. Cushing, S. Wu, S. Levy, C. Chiu, R. E. Schwartz, N. Tatonetti, H. Rennert, M. Imielinski, and C. E. Mason, "Shotgun transcriptome, spatial omics, and isothermal profiling of SARS-CoV-2 infection reveals unique host responses, viral diversification, and drug interactions," Nat. Commun. **12**, 1660 (2021).
[50]T. M. Delorey, C. G. K. Ziegler, G. Heimberg, R. Normand, Y. Yang, Å. Segerstolpe, D. Abbondanza, S. J. Fleming, A. Subramanian, D. T. Montoro, K. A. Jagadeesh, K. K. Dey, P. Sen, M. Slyper, Y. H. Pita-Juárez, D. Phillips, J. Biermann, Z. Bloom-Ackermann, N. Barkas, A. Ganna, J. Gomez, J. C. Melms, I. Katsyv, E. Normandin, P. Naderi, Y. V. Popov, S. S. Raju, S. Niezen, L. T. Y. Tsai, K. J. Siddle, M. Sud, V. M. Tran, S. K. Vellarikkal, Y. Wang, L. Amir-Zilberstein, D. S. Atri, J. Beechem, O. R. Brook, J. Chen, P. Divakar, P. Dorceus, J. M. Engreitz, A. Essene, D. M. Fitzgerald, R. Fropf, S. Gazal, J. Gould, J. Grzyb, T. Harvey, J. Hecht, T. Hether, J. Jané-Valbuena, M. Leney-Greene, H. Ma, C. McCabe, D. E. McLoughlin, E. M. Miller, C. Muus, M. Niemi, R. Padera, L. Pan, D. Pant, C. Pe'er, J. Pfiffner-Borges, C. J. Pinto, J. Plaisted, J. Reeves, M. Ross, M. Rudy, E. H. Rueckert, M. Siciliano, A. Sturm, E. Todres, A. Waghray, S. Warren, S. Zhang, D. R. Zollinger, L. Cosimi, R. M. Gupta, N. Hacohen, H. Hibshoosh, W. Hide, A. L. Price, J. Rajagopal, P. R. Tata, S. Riedel, G. Szabo, T. L. Tickle, P. T. Ellinor, D. Hung, P. C. Sabeti, R. Novak, R. Rogers, D. E. Ingber, Z. G. Jiang, D. Juric, M. Babadi, S. L. Farhi, B. Izar, J. R. Stone, I. S. Vlachos, I. H. Solomon, O. Ashenberg, C. B. M. Porter, B. Li, A. K. Shalek, A.-C. Villani, O. Rozenblatt-Rosen, and A. Regev, "COVID-19 tissue atlases reveal SARS-CoV-2 pathology and cellular targets," Nature **595**, 107–113 (2021).
[51]S. Vickovic, D. Schapiro, K. Carlberg, B. Lotstedt, L. Larsson, "M. Korotkova, A. H. Hensvold, A. I. Catrina, P. K. Sorger, V. Malmstrom, A. Regev, and P. L. Ståhl, Three-dimensional spatial transcriptomics uncovers cell type dynamics in the rheumatoid arthritis synovium," bioRxiv (2020).
[52]K. Carlberg, M. Korotkova, L. Larsson, A. I. Catrina, P. L. Ståhl, and V. Malmstrom, "Exploring inflammatory signatures in arthritic joint biopsies with spatial transcriptomics," Sci. Rep. **9**, 18975 (2019).
[53]E. Berglund, J. Maaskola, N. Schultz, S. Friedrich, M. Marklund, J. Bergenstråhle, F. Tarish, A. Tanoglidi, S. Vickovic, L. Larsson, F. Salmén, C. Ogris, K. Wallenborg, J. Lagergren, P. Ståhl, E. Sonnhammer, T. Helleday, and J. Lundeberg, "Spatial maps of prostate cancer transcriptomes reveal an unexplored landscape of heterogeneity," Nat. Commun. **9**, 2419 (2018).
[54]R. Moncada, D. Barkley, F. Wagner, M. Chiodin, J. C. Devlin, M. Baron, C. H. Hajdu, D. M. Simeone, and I. Yanai, "Integrating microarray-based spatial transcriptomics and single-cell RNA-seq reveals tissue architecture in pancreatic ductal adenocarcinomas," Nat. Biotechnol. **38**, 333–342 (2020).
[55]A. L. Ji, A. J. Rubin, K. Thrane, S. Jiang, D. L. Reynolds, R. M. Meyers, M. G. Guo, B. M. George, A. Mollbrink, J. Bergenstråhle, L. Larsson, Y. Bai, B. Zhu, A. Bhaduri, J. M. Meyers, X. Rovira-Clavé, S. T. Hollmig, S. Z. Aasi, G. P. Nolan, J. Lundeberg, and P. A. Khavari, "Multimodal analysis of composition and spatial architecture in human squamous cell carcinoma," Cell **182**, 497–514 (2020).
[56]W.-T. Chen, A. Lu, K. Craessaerts, B. Pavie, C. Sala Frigerio, N. Corthout, X. Qian, J. Laláková, M. Kuhnemund, I. Voytyuk, L. Wolfs, R. Mancuso, E. Salta, S. Balusu, A. Snellinx, S. Munck, A. Jurek, J. Fernandez Navarro, T. C. Saido, I. Huitinga, J. Lundeberg, M. Fiers, and B. De Strooper, "Spatial transcriptomics and in situ sequencing to study alzheimer's disease," Cell **182**, 976–991 (2020).
[57]J. Backdahl, L. Franzén, L. Massier, Q. Li, J. Jalkanen, H. Gao, A. Andersson, N. Bhalla, A. Thorell, M. Rydén, P. L. Ståhl, and N. Mejhert, "Spatial mapping reveals human adipocyte subpopulations with distinct sensitivities to insulin," Cell Metab. **33**, 1869–1882 (2021).
[58]G. Theocharidis, B. E. Thomas, D. Sarkar, H. L. Mumme, W. J. R. Pilcher, B. Dwivedi, T. Sandoval-Schaefer, R. F. Sîrbulescu, A. Kafanas, I. Mezghani, P. Wang, A. Lobao, I. S. Vlachos, B. Dash, H. C. Hsia, V. Horsley, S. S. Bhasin, A. Veves, and M. Bhasin, "Single cell transcriptomic landscape of diabetic foot ulcers," Nat. Commun. **13**, 181 (2022).
[59]R. Satija, J. A. Farrell, D. Gennert, A. F. Schier, and A. Regev, "Spatial reconstruction of single-cell gene expression data," Nat. Biotechnol. **33**, 495–502 (2015).
[60]M. Nitzan, N. Karaiskos, N. Friedman, and N. Rajewsky, "Gene expression cartography," Nature **576**, 132–137 (2019).
[61]F. Maseda, Z. Cang, and Q. Nie, "DEEPsc: A deep learning-based map connecting single-cell transcriptomics and spatial imaging data," Front. Genet. **12**, 348 (2021).
[62]K. Achim, J.-B. Pettit, L. R. Saraiva, D. Gavriouchkina, T. Larsson, D. Arendt, and J. C. Marioni, "High-throughput spatial mapping of single-cell RNA-seq data to tissue of origin," Nat. Biotechnol. **33**, 503–509 (2015).
[63]T. Peng, G. M. Chen, and K. Tan, "Gluer: Integrative analysis of single-cell omics and imaging data by deep neural network," bioRxiv (2021).
[64]A. Andersson, J. Bergenstråhle, M. Asp, L. Bergenstråhle, A. Jurek, J. F. Navarro, and J. Lundeberg, "Single-cell and spatial transcriptomics enables probabilistic inference of cell type topography," Commun. Biol. **3**, 565 (2020).
[65]Q. Song and J. Su, "DSTG: Deconvoluting spatial transcriptomics data through graph-based artificial intelligence," Briefings Bioinf. **22**, bbaa414 (2021).
[66]M. Elosua-Bayes, P. Nieto, E. Mereu, I. Gut, and H. Heyn, "SPOTlight: Seeded NMF regression to deconvolute spatial transcriptomics spots with single-cell transcriptomes," Nucl. Acids Res. **49**, e50 (2021).
[67]D. M. Cable, E. Murray, L. S. Zou, A. Goeva, E. Z. Macosko, F. Chen, and R. A. Irizarry, "Robust decomposition of cell type mixtures in spatial transcriptomics," Nat. Biotechnol. **40**, 517–526 (2021).
[68]R. Dong and G.-C. Yuan, "SpatialDWLS: Accurate deconvolution of spatial transcriptomic data," Genome Biol. **22**, 145 (2021).
[69]R. Lopez, B. Li, H. Keren-Shaul, P. Boyeau, M. Kedmi, D. Pilzer, A. Jelinski, I. Yofe, E. David, A. Wagner, C. Ergen, Y. Addadi, O. Golani, F. Ronchese, M. I. Jordan, I. Amit, and N. Yosef, "Destvi identifies continuums of cell types in spatial transcriptomics data," Nat. Biotechnol. **40**, 1360–1369 (2022).
[70]V. Kleshchevnikov, A. Shmatko, E. Dann, A. Aivazidis, H. W. King, T. Li, R. Elmentaite, A. Lomakin, V. Kedlian, A. Gayoso, M. S. Jain, J. S. Park, L. Ramona, E. Tuck, A. Arutyunyan, R. Vento-Tormo, M. Gerstung, L. James, O. Stegle, and O. A. Bayraktar, "Cell2location maps fine-grained cell types in spatial transcriptomics," Nat. Biotechnol. **40**, 661–671 (2022).

[71]Y. Ma and X. Zhou, "Spatially informed cell-type deconvolution for spatial transcriptomics," Nat. Biotechnol. **40**, 1349–1359 (2022).
[72]Q. Zhu, S. Shah, R. Dries, L. Cai, and G.-C. Yuan, "Identification of spatially associated subpopulations by combining scRNAseq and sequential fluorescence in situ hybridization data," Nat. Biotechnol. **36**, 1183–1190 (2018).
[73]X. Tan, A. Su, M. Tran, and Q. Nguyen, "SpaCell: Integrating tissue morphology and spatial gene expression to predict disease cells," Bioinformaics **36**, 2293–2294 (2020).
[74]D. Pham, X. Tan, J. Xu, L. F. Grice, P. Y. Lam, "A. Raghubar, J. Vukovic, M. J. Ruitenberg, and Q. Nguyen, stLearn: Integrating spatial location, tissue morphology and gene expression to find cell types, cell-cell interactions and spatial trajectories within undissociated tissues," bioRxiv (2020).
[75]E. Zhao, M. R. Stone, X. Ren, J. Guenthoer, K. S. Smythe, T. Pulliam, S. R. Williams, C. R. Uytingco, S. E. B. Taylor, P. Nghiem, J. H. Bielas, and R. Gottardo, "Spatial transcriptomics at subspot resolution with BayesSpace," Nat. Biotechnol. **39**, 1375–1384 (2021).
[76]J. Hu, X. Li, K. Coleman, A. Schroeder, N. Ma, D. J. Irwin, E. B. Lee, R. T. Shinohara, and M. Li, "SpaGCN: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network," Nat. Methods **18**, 1342–1351 (2021).
[77]D. Edsgard, P. Johnsson, and R. Sandberg, "Identification of spatial expression trends in single-cell gene expression data," Nat. Methods **15**, 339–342 (2018).
[78]V. Svensson, S. A. Teichmann, and O. Stegle, "Spatialde: Identification of spatially variable genes," Nat. Methods **15**, 343–346 (2018).
[79]S. Sun, J. Zhu, and X. Zhou, "Statistical analysis of spatial expression patterns for spatially resolved transcriptomic studies," Nat. Methods **17**, 193–200 (2020).
[80]Z. Cang and Q. Nie, "Inferring spatial and signaling relationships between cells from single cell transcriptomic data," Nat. Commun. **11**, 2084 (2020).
[81]J. Tanevski, R. O. R. Flores, A. Gabor, D. Schapiro, and J. Saez-Rodriguez, "Explainable multi-view framework for dissecting intercellular signaling from highly multiplexed spatial data," bioRxiv (2021).
[82]R. Dries, Q. Zhu, R. Dong, C.-H. L. Eng, H. Li, K. Liu, Y. Fu, T. Zhao, A. Sarkar, F. Bao, R. E. George, N. Pierson, L. Cai, and G.-C. Yuan, "Giotto: A toolbox for integrative analysis and visualization of spatial expression data," Genome Biol. **22**, 78 (2021).
[83]R. K. Gupta and J. Kuznicki, "Biological and medical importance of cellular heterogeneity deciphered by single-cell RNA sequencing," Cells **9**, 1751 (2020).
[84]P. van Galen, V. Hovestadt, M. H. Wadsworth II, T. K. Hughes, G. K. Griffin, S. Battaglia, J. A. Verga, J. Stephansky, T. J. Pastika, J. Lombardi Story, G. S. Pinkus, O. Pozdnyakova, I. Galinsky, R. M. Stone, T. A. Graubert, A. K. Shalek, J. C. Aster, A. A. Lane, and B. E. Bernstein, "Single-cell RNA-seq reveals AML hierarchies relevant to disease progression and immunity," Cell **176**, 1265–1281 (2019).
[85]T. Masuda, R. Sankowski, O. Staszewski, C. Bottcher, L. Amann, Sagar, C. Scheiwe, S. Nessler, P. Kunz, G. van Loo, V. A. Coenen, P. C. Reinacher, A. Michel, U. Sure, R. Gold, D. Grun, J. Priller, C. Stadelmann, and M. Prinz, "Spatial and temporal heterogeneity of mouse and human microglia at single-cell resolution," Nature **566**, 388–392 (2019).
[86]J. M. Churko, P. Garg, B. Treutlein, M. Venkatasubramanian, H. Wu, J. Lee, Q. N. Wessells, S.-Y. Chen, W.-Y. Chen, K. Chetal, G. Mantalas, N. Neff, E. Jabart, A. Sharma, G. P. Nolan, N. Salomonis, and J. C. Wu, "Defining human cardiac transcription factor hierarchies using integrated single-cell heterogeneity analysis," Nat. Commun. **9**, 4906 (2018).
[87]A. Gross, J. Schoendube, S. Zimmermann, M. Steeb, R. Zengerle, and P. Koltay, "Technologies for single-cell isolation," Int. J. Mol. Sci. **16**, 16897–16919 (2015).
[88]S. Ma, T. W. Murphy, and C. Lu, "Microfluidics for genome-wide studies involving next generation sequencing," Biomicrofluidics **11**, 021501 (2017).
[89]B. Hwang, J. H. Lee, and D. Bang, "Single-cell RNA sequencing technologies and bioinformatics pipelines," Exp. Mol. Med. **50**, 1–14 (2018).
[90]A. Haque, J. Engel, S. A. Teichmann, and T. Lonnberg, "A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications," Genome Med. **9**, 75 (2017).
[91]R. Stark, M. Grzelak, and J. Hadfield, "RNA sequencing: The teenage years," Nat. Rev. Genet. **20**, 631–656 (2019).
[92]E. L. van Dijk, H. Auger, Y. Jaszczyszyn, and C. Thermes, "Ten years of next-generation sequencing technology," Trends Genet. **30**, 418–426 (2014).
[93]X. Zhang, T. Li, F. Liu, Y. Chen, J. Yao, Z. Li, Y. Huang, and J. Wang, "Comparative analysis of droplet-based ultra-high-throughput single-cell RNA-seq systems," Mol. Cell **73**, 130–142 (2019).
[94]A. A. Kolodziejczyk, J. K. Kim, V. Svensson, J. C. Marioni, and S. A. Teichmann, "The technology and biology of single-cell RNA sequencing," Mol. Cell **58**, 610–620 (2015).
[95]X.-T. Huang, X. Li, P.-Z. Qin, Y. Zhu, S.-N. Xu, and J.-P. Chen, "Technical advances in single-cell RNA sequencing and applications in normal and malignant hematopoiesis," Front. Oncol. **8**, 582 (2018).
[96]M. Asp, J. Bergenstråhle, and J. Lundeberg, "Spatially resolved transcriptomes—Next generation tools for tissue exploration," BioEssays **42**, 1900221 (2020).
[97]L. Moses and L. Pachter, "Museum of spatial transcriptomics," Nat. Methods **19**, 534–546 (2022).
[98]B. W. Rautenstrauss and T. Liehr, *FISH Technology* (Springer Verlag, Berlin, 2002).
[99]E. Jensen, "Technical review: In situ hybridization," Anat. Rec. **297**, 1349–1353 (2014).
[100]M. M. Hilscher, D. Gyllborg, C. Yokota, and M. Nilsson, "In situ sequencing: A high-throughput, multi-targeted gene expression profiling technique for cell typing in tissue sections," in *In Situ Hybridization Protocols*, edited by B. S. Nielsen and J. Jones (Springer, New York, NY, 2020), pp. 313–329.
[101]J. A. Weinstein, A. Regev, and F. Zhang, "Dna microscopy: Optics-free spatio-genetic imaging by a stand-alone chemical reaction," Cell **178**, 229–241 (2019).
[102]C. Larsson, I. Grundberg, O. Soderberg, and M. Nilsson, "In situ detection and genotyping of individual mRNA molecules," Nat. Methods **7**, 395–397 (2010).
[103]D. Gyllborg, C. M. Langseth, X. Qian, E. Choi, S. M. Salas, M. M. Hilscher, E. S. Lein, and M. Nilsson, "Hybridization-based in situ sequencing (HYBISS) for spatially resolved transcriptomics in human and mouse brain tissue," Nucl. Acids Res. **48**, e112 (2020).
[104]X. Chen, Y.-C. Sun, G. M. Church, J. H. Lee, and A. M. Zador, "Efficient in situ barcode sequencing using padlock probe-based BaristaSeq," Nucl. Acids Res. **46**, e22 (2017).
[105]H. Lee, S. M. Salas, D. Gyllborg, and M. Nilsson, "Direct RNA targeted transcriptomic profiling in tissue using hybridization-based RNA in situ sequencing (HybRISS)," bioRxiv (2020).
[106]J. H. Lee, E. R. Daugharthy, J. Scheiman, R. Kalhor, T. C. Ferrante, R. Terry, B. M. Turczyk, J. L. Yang, H. S. Lee, J. Aach, K. Zhang, and G. M. Church, "Fluorescent in situ sequencing (FISSEQ) of RNA for gene expression profiling in intact cells and tissues," Nat. Protoc. **10**, 442–458 (2015).
[107]J. H. Lee, "Quantitative approaches for investigating the spatial context of gene expression," WIREs Syst. Biol. Med. **9**, e1369 (2017).
[108]P. R. Gudla, K. Nakayama, G. Pegoraro, and T. Misteli, "Spotlearn: Convolutional neural network for detection of fluorescence in situ hybridization (fish) signals in high-throughput imaging approaches," Cold Spring Harbor Symp. Quant. Biol. **82**, 57–70 (2017).
[109]G. Partel, M. M. Hilscher, G. Milli, L. Solorzano, A. H. Klemm, M. Nilsson, and C. Wahlby, "Automated identification of the mouse brain's spatial compartments from in situ sequencing data," BMC Biol. **18**, 144 (2020).
[110]E. Pardo, J. M. T. Morgado, and N. Malpica, "Semantic segmentation of mFISH images using convolutional networks," Cytometry, Part A **93**, 620–627 (2018).
[111]Z. Frankenstein, N. Uraoka, U. Aypar, R. Aryeequaye, M. Rao, M. Hameed, Y. Zhang, and Y. Yagi, "Automated 3d scoring of fluorescence in situ hybridization (FISH) using a confocal whole slide imaging scanner," Appl. Microsc. **51**, 4 (2021).
[112]R. Littman, Z. Hemminger, R. Foreman, D. Arneson, G. Zhang, F. Gómez-Pinilla, X. Yang, and R. Wollman, "Joint cell segmentation and cell type annotation for spatial transcriptomics," Mol. Syst. Biol. **17**, e10108 (2021).
[113]x.Genomics, Visium Spatial Gene Expression Reagent Kits—User Guide, 2021.
[114]M. V. Hunter, R. Moncada, J. M. Weiss, I. Yanai, and R. M. White, "Spatially resolved transcriptomics reveals the architecture of the tumor/microenvironment interface," bioRxiv (2021).

[115]D. Fawkner-Corbett, A. Antanaviciute, K. Parikh, M. Jagielowicz, A. S. Gerós, T. Gupta, N. Ashley, D. Khamis, D. Fowler, E. Morrissey, C. Cunningham, P. R. Johnson, H. Koohy, and A. Simmons, "Spatiotemporal analysis of human intestinal development at single-cell resolution," Cell **184**, 810–826 (2021).
[116]M. Asp, S. Giacomello, L. Larsson, C. Wu, D. Furth, X. Qian, E. Wardell, J. Custodio, J. Reimegård, F. Salmén, C. Osterholm, P. L. Ståhl, E. Sundstrom, E. Åkesson, O. Bergmann, M. Bienko, A. Månsson-Broberg, M. Nilsson, C. Sylvén, and J. Lundeberg, "A spatiotemporal organ-wide gene expression and cell atlas of the developing human heart," Cell **179**, 1647–1660 (2019).
[117]M. Y. Batiuk, T. Tyler, S. Mei, R. Rydbirk, V. Petukhov, D. Sedmak, E. Frank, V. Feher, N. Habek, Q. Hu, A. Igolkina, L. Roszik, U. Pfisterer, Z. Petanjek, I. Adorjan, P. V. Kharchenko, and K. Khodosevich, "Selective vulnerability of supragranular layer neurons in schizophrenia," bioRxiv (2021).
[118]C. Ortiz, J. F. Navarro, A. Jurek, A. Martin, J. Lundeberg, and K. Meletis, "Molecular atlas of the adult mouse brain," Sci. Adv. **6**, eabb3446 (2020).
[119]10x Genomics, see https://www.10xgenomics.com/products/spatial-gene-expression for "Spatial gene expression," 2022.
[120]I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (The MIT Press, 2016).
[121]S. Lloyd, "Least squares quantization in pcm," IEEE Trans. Inf. Theory **28**, 129–137 (1982).
[122]L. Prechelt, "Early stopping-but when?," in *Proceedings of the Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop* (Springer-Verlag, Berlin, Heidelberg, 1998), pp. 55–69.
[123]G. Cybenko, "Approximation by superpositions of a sigmoidal function," Math. Control Signals Syst. **2**, 303–314 (1989).
[124]F. Scarselli and A. Chung Tsoi, "Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results," Neural Networks **11**, 15–37 (1998).
[125]T. Chen and H. Chen, "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems," IEEE Trans. Neural Networks **6**, 911–917 (1995).
[126]M. Uzair and N. Jamil, "Effects of hidden layers on the efficiency of neural networks," in *Proceedings of the IEEE 23rd International Multitopic Conference (INMIC)* (IEEE, 2020), pp. 1–6.
[127]A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Curran Associates, Inc., 2012), Vol. 25.
[128]X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," arXiv:2108.11539 [cs.CV] (2021).
[129]M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," arXiv:2104.00298 [cs.CV] (2021).
[130]A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017), Vol. 30.
[131]T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," arXiv:2005.14165 [cs.CL] (2020).
[132]J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapolis, MN, 2–7 June 2019, edited by J. Burstein, C. Doran, and T. Solorio (Association for Computational Linguistics, 2019), Vol. 1 (Long and Short Papers), pp. 4171–4186.
[133]R. Poplin, P.-C. Chang, D. Alexander, S. Schwartz, T. Colthurst, A. Ku, D. Newburger, J. Dijamco, N. Nguyen, P. T. Afshar, S. S. Gross, L. Dorfman, C. Y. McLean, and M. A. DePristo, "A universal SNP and small-indel variant caller using deep neural networks," Nat. Biotechnol. **36**, 983–987 (2018).
[134]H. Li, U. Shaham, K. P. Stanton, Y. Yao, R. R. Montgomery, and Y. Kluger, "Gating mass cytometry data by deep learning," Bioinformatics **33**, 3423–3430 (2017).
[135]J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with alphafold," Nature **596**, 583–589 (2021).
[136]T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 24–26 April 2017 (OpenReview.net, 2017).
[137]K. Hornik, "Approximation capabilities of multilayer feedforward networks," Neural Networks **4**, 251–257 (1991).
[138]V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)* (Omnipress, Madison, WI, 2010), pp. 807–814.
[139]J. Nocedal and S. Wright, *Numerical Optimization* (Springer Science and Business Media, 2006).
[140]X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, edited by Y. W. Teh and M. Titterington (PMLR, Chia Laguna Resort, Sardinia, Italy, 2010), Vol. 9, pp. 249–256.
[141]K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imageNet classification," in Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015).
[142]S. Ruder, "An overview of gradient descent optimization algorithms," arXiv:1609.04747 (2016).
[143]S. Smith, E. Elsen, and S. De, "On the generalization benefit of noise in stochastic gradient descent," in *Proceedings of the 37th International Conference on Machine Learning*, edited by H. De III and A. Singh (PMLR, 2020), Vol. 119, pp. 9058–9067.
[144]D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature **323**, 533–536 (1986).
[145]D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, 7–9 May 2015, edited by Y. Bengio and Y. LeCun (Conference Track Proceedings, 2015).
[146]J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," J. Mach. Learn. Res. **12**, 2121–2159 (2011), see https://jmlr.org/papers/v12/duchi11a.html
[147]Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," IEEE Trans. Neural Networks Learn. Syst. **30**, 3212–3232 (2019).
[148]D. Marr and E. Hildreth, "Theory of edge detection," Proc. R. Soc. London, Ser. B **207**, 187–217 (1980).
[149]Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural Comput. **1**, 541–551 (1989).
[150]Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks* (MIT Press, Cambridge, MA, 1998), pp. 255–258.
[151]Note that if the model chose a separate parameter for each $\mathbf{x}^{(i)}$, for $i = 1, \ldots, n$, then the model could not generalize to any inputs where $|X| > n$ (size of X is greater than n).
[152]J. F. Kolen and S. C. Kremer, "Gradient flow in recurrent nets: The difficulty of learning longterm dependencies," in *A Field Guide to Dynamical Recurrent Networks* (IEEE, 2001), pp. 237–243.
[153]S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput. **9**, 1735–1780 (1997).
[154]J. C. Kimmel, A. S. Brack, and W. F. Marshall, "Deep convolutional and recurrent neural networks for cell motility discrimination and prediction," IEEE/ACM Trans. Comput. Biol. Bioinf. **18**, 562–574 (2021).

[155]S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, edited by F. Bach and D. Blei (PMLR, Lille, France, 2015), Vol. 37, pp. 448–456.
[156]K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016), pp. 770–778.
[157]A. Shah, E. Kadam, H. Shah, S. Shinde, and S. Shingade, "Deep residual networks with exponential linear unit," in *Proceedings of the Third International Symposium on Computer Vision and the Internet* (Association for Computing Machinery, New York, NY, 2016), pp. 59–65.
[158]J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in 2009 IEEE Conference on Computer Vision and Pattern Recognition (2009) pp. 248–255.
[159]D. H. Ballard, "Modular learning in neural networks," in *Proceedings of the AAAI* (AAAI, 1987), Vol. 1, pp. 279–284.
[160]G. E. Hinton, "Connectionist learning procedures11 this chapter appeared in volume 40 of artificial intelligence in 1989, reprinted with permission of North-holland publishing. It is a revised version of technical report cmu-cs-87-115, which has the same title and was prepared in june 1987 while the author was at Carnegie Mellon University. the research was supported by contract n00014-86-k-00167 from the office of naval research and by grant ist-8520359 from the national science foundation," in *Machine Learning*, edited by Y. Kodratoff and R. S. Michalski (Morgan Kaufmann, San Francisco, CA, 1990), Chap. 20, pp. 555–610.
[161]D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in Proceedings of the International Conference on Learning Representations (2014).
[162]I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Curran Associates, Inc., 2014), Vol. 27.
[163]J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick, "Lagging inference networks and posterior collapse in variational autoencoders," in Proceedings of the International Conference on Learning Representations (2019).
[164]Z. Yang, Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick, "Improved variational autoencoders for text modeling using dilated convolutions," in *Proceedings of the International Conference on Machine Learning* (PMLR, 2017), pp. 3881–3890.
[165]A. A. Heydari and A. Mehmood, "SRVAE: Super resolution using variational autoencoders," Proc. SPIE **11400**, 114000U (2020).
[166]S. Semeniuta, A. Severyn, and E. Barth, "A hybrid convolutional variational autoencoder for text generation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Copenhagen, Denmark, 2017), pp. 627–637.
[167]A. A. Heydari, C. A. Thompson, and A. Mehmood, "Softadapt: Techniques for adaptive loss weighting of neural networks with multi-part loss functions," arXiv:1912.12355 (2019).
[168]I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein auto-encoders," in Proceedings of the International Conference on Learning Representations (2018).
[169]T. Daniel and A. Tamar, "SoftIntroVAE: Analyzing and improving the introspective variational autoencoder," arXiv:2012.13253 [cs.LG] (2021).
[170]The astute reader will note that although Maseda *et al.* refer to DEEPsc as DL model, the methods's two-layer FFNN is not considered deep model in most definitions.
[171]K. Nikos, W. Philipp, A. Jonathan, B. Anastasiya, A. Salah, K. Claudia, K. Christine, R. Nikolaus, and R. P. Zinzen, "The drosophila embryo at single-cell transcriptome resolution," Science **358**, 194–199 (2017).
[172]B. Tasic, V. Menon, T. N. Nguyen, T. K. Kim, T. Jarsky, Z. Yao, B. Levi, L. T. Gray, S. A. Sorensen, T. Dolbeare, D. Bertagnolli, J. Goldy, N. Shapovalova, S. Parry, C. Lee, K. Smith, A. Bernard, L. Madisen, S. M. Sunkin, M. Hawrylycz, C. Koch, and H. Zeng, "Adult mouse cortical cell taxonomy revealed by single cell transcriptomics," Nat. Neurosci. **19**, 335–346 (2016).
[173]S. Joost, A. Zeisel, T. Jacob, X. Sun, G. L. Manno, P. Lonnerberg, S. Linnarsson, and M. Kasper, "Single-cell transcriptomics reveals that differentiation and spatial signatures shape epidermal and hair follicle heterogeneity," Cell Syst. **3**, 221–237 (2016).
[174]Dropout refers to the scenario when a gene is observed at a moderate or high expression level in a subset of cells, but not detected in other cells.
[175]J. Ding, X. Adiconis, S. K. Simmons, M. S. Kowalczyk, C. C. Hession, N. D. Marjanovic, T. K. Hughes, M. H. Wadsworth, T. Burks, L. T. Nguyen, J. Y. H. Kwon, B. Barak, W. Ge, A. J. Kedaigle, S. Carroll, S. Li, N. Hacohen, O. Rozenblatt-Rosen, A. K. Shalek, A.-C. Villani, A. Regev, and J. Z. Levin, "Systematic comparison of single-cell and single-nucleus RNA-sequencing methods," Nat. Biotechnol. **38**, 737–746 (2020).
[176]See https://pytorch.org/ for PyTorch is one of the most popular DL library in Python.
[177]A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, "G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, Pytorch: An imperative style, high-performance deep learning library," arXiv:1912.01703 (2019).
[178]O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of the Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, edited by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi (Springer International Publishing, Cham, 2015), pp. 234–241.
[179]G. R. Koch, "Siamese neural networks for one-shot image recognition," in Proceedings of the 32nd International Conference on Machine Learning (2015).
[180]G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2017), pp. 2261–2269.
[181]Y. Liu, M. Yang, Y. Deng, G. Su, A. Enninful, C. C. Guo, T. Tebaldi, D. Zhang, D. Kim, Z. Bai, E. Norris, A. Pan, J. Li, Y. Xiao, S. Halene, and R. Fan, "High-spatial-resolution multi-omics sequencing via deterministic barcoding in tissue," Cell **183**, 1665–1681 (2020).
[182]See https://www.nanostring.com/ for information on Nanostring's GeoMx technology.
[183]J. Hu, A. Schroeder, K. Coleman, C. Chen, B. J. Auerbach, and M. Li, "Statistical and machine learning methods for spatially resolved transcriptomics with histology," Comput. Struct. Biotechnol. J. **19**, 3829–3841 (2021).
[184]L. Garcia-Alonso, L.-F. Handfield, K. Roberts, K. Nikolakopoulou, R. C. Fernando, "L. Gardner, B. Woodhams, A. Arutyunyan, K. Polanski, R. Hoo, C. Sancho-Serra, T. Li, K. Kwakwa, E. Tuck, V. Kleshchevnikov, A. Tarkowska, T. Porter, C. I. Mazzeo, S. van Dongen, M. Dabrowska, V. Vaskivskyi, K. T. Mahbubani, J.-E. Park, M. Jimenez-Linan, L. Campos, V. Kiselev, C. Lindskog, P. Ayuk, E. Prigmore, M. R. Stratton, K. Saeb-Parsy, A. Moffett, L. Moore, O. A. Bayraktar, S. A. Teichmann, M. Y. Turco, and R. Vento-Tormo, "Mapping the temporal and spatial dynamics of the human endometrium in vivo and in vitro," bioRxiv (2021).
[185]R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef, "Deep generative modeling for single-cell transcriptomics," Nat. Methods **15**, 1053–1058 (2018).
[186]D. Grun, L. Kester, and A. van Oudenaarden, "Validation of noise models for single-cell transcriptomics," Nat. Methods **11**, 637–640 (2014).
[187]C. H. Grønbech, M. F. Vording, P. N. Timshel, C. K. Sønderby, T. H. Pers, and O. Winther, "scVAE: Variational auto-encoders for single-cell gene expression data," Bioinformatics **36**, 4415–4422 (2020).
[188]J. Aragón, D. Eberly, and S. Eberly, "Existence and uniqueness of the maximum likelihood estimator for the two-parameter negative binomial distribution," Stat. Probab. Lett. **15**, 375–379 (1992).
[189]J. R. Kettenring, "Canonical analysis of several sets of variables," Biometrika **58**, 433–451 (1971).
[190]L. Haghverdi, A. T. L. Lun, M. D. Morgan, and J. C. Marioni, "Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors," Nat. Biotechnol. **36**, 421–427 (2018).
[191]G. Pasquini, J. E. Rojo Arias, P. Schafer, and V. Busskamp, "Automated methods for cell type annotation on SCRNA-seq data," Comput. Struct. Biotechnol. J. **19**, 961–969 (2021).

[192]V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," J. Stat. Mech. Theory Exp. **2008**, P10008.
[193]V. A. Traag, L. Waltman, and N. J. van Eck, "From Louvain to Leiden: Guaranteeing well-connected communities," Sci. Rep. **9**, 5233 (2019).
[194]E. S. Lein, M. J. Hawrylycz, N. Ao, M. Ayres, A. Bensinger, A. Bernard, A. F. Boe, M. S. Boguski, K. S. Brockway, E. J. Byrnes, L. Chen, L. Chen, T.-M. Chen, M. Chi Chin, J. Chong, B. E. Crook, A. Czaplinska, C. N. Dang, S. Datta, N. R. Dee, A. L. Desaki, T. Desta, E. Diep, T. A. Dolbeare, M. J. Donelan, H.-W. Dong, J. G. Dougherty, B. J. Duncan, A. J. Ebbert, G. Eichele, L. K. Estin, C. Faber, B. A. Facer, R. Fields, S. R. Fischer, T. P. Fliss, C. Frensley, S. N. Gates, K. J. Glattfelder, K. R. Halverson, M. R. Hart, J. G. Hohmann, M. P. Howell, D. P. Jeung, R. A. Johnson, P. T. Karr, R. Kawal, J. M. Kidney, R. H. Knapik, C. L. Kuan, J. H. Lake, A. R. Laramee, K. D. Larsen, C. Lau, T. A. Lemon, A. J. Liang, Y. Liu, L. T. Luong, J. Michaels, J. J. Morgan, R. J. Morgan, M. T. Mortrud, N. F. Mosqueda, L. L. Ng, R. Ng, G. J. Orta, C. C. Overly, T. H. Pak, S. E. Parry, S. D. Pathak, O. C. Pearson, R. B. Puchalski, Z. L. Riley, H. R. Rockett, S. A. Rowland, J. J. Royall, M. J. Ruiz, N. R. Sarno, K. Schaffnit, N. V. Shapovalova, T. Sivisay, C. R. Slaughterbeck, S. C. Smith, K. A. Smith, B. I. Smith, A. J. Sodt, N. N. Stewart, K.-R. Stumpf, S. M. Sunkin, M. Sutram, A. Tam, C. D. Teemer, C. Thaller, C. L. Thompson, L. R. Varnam, A. Visel, R. M. Whitlock, P. E. Wohnoutka, C. K. Wolkey, V. Y. Wong, M. Wood, M. B. Yaylaoglu, R. C. Young, B. L. Youngstrom, X. F. Yuan, B. Zhang, T. A. Zwingman, and A. R. Jones, "Genome-wide atlas of gene expression in the adult mouse brain," Nature **445**, 168–176 (2007).
[195]B. Pardo, A. Spangler, L. M. Weber, S. C. Page, S. C. Hicks, A. E. Jaffe, K. Martinowich, K. R. Maynard, and L. Collado-Torres, "spatialLIBD: An R/bioconductor package to visualize spatially-resolved transcriptomics data," BMC Genomics **23**, 434 (2022).
[196]K. R. Maynard, L. Collado-Torres, L. M. Weber, C. Uytingco, B. K. Barry, S. R. Williams, J. L. Catallini, M. N. Tran, Z. Besich, M. Tippani, J. Chew, Y. Yin, J. E. Kleinman, T. M. Hyde, N. Rao, S. C. Hicks, K. Martinowich, and A. E. Jaffe, "Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex," Nat. Neurosci. **24**, 425–436 (2021).
[197]P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," J. Comput. Appl. Math. **20**, 53–65 (1987).
[198]E. Armingol, A. Officer, O. Harismendy, and N. E. Lewis, "Deciphering cell–cell interactions and communication from gene expression," Nat. Rev. Genet. **22**, 71–88 (2021).
[199]D. Arneson, G. Zhang, Z. Ying, Y. Zhuang, H. R. Byun, I. S. Ahn, F. Gomez-Pinilla, and X. Yang, "Single cell molecular alterations reveal target cells and pathways of concussive brain injury," Nat. Commun. **9**, 3894 (2018).
[200]D. A. Skelly, G. T. Squiers, M. A. McLellan, M. T. Bolisetty, P. Robson, N. A. Rosenthal, and A. R. Pinto, "Single-cell transcriptional profiling reveals cellular diversity and intercommunication in the mouse heart," Cell Rep. **22**, 600–610 (2018).
[201]M. Cohen, A. Giladi, A.-D. Gorki, D. G. Solodkin, M. Zada, A. Hladik, A. Miklosi, T.-M. Salame, K. B. Halpern, E. David, S. Itzkovitz, T. Harkany, S. Knapp, and I. Amit, "Lung single-cell signaling interaction map reveals basophil role in macrophage imprinting," Cell **175**, 1031–1044 (2018).
[202]B. He, L. Bergenstråhle, L. Stenbeck, A. Abid, A. Andersson, Å. Borg, J. Maaskola, J. Lundeberg, and J. Zou, "Integrating spatial gene expression and breast tumour morphology via deep learning," Nat. Biomed. Eng. **4**, 827–834 (2020).
[203]M. Efremova, M. Vento-Tormo, S. A. Teichmann, and R. Vento-Tormo, "CellPhoneDB: Inferring cell–cell communication from combined expression of multi-subunit ligand–receptor complexes," Nat. Protoc. **15**, 1484–1506 (2020).
[204]S. Cabello-Aguilar, M. Alame, F. Kon-Sun-Tack, C. Fau, M. Lacroix, and J. Colinge, "SingleCellSignalR: Inference of intercellular networks from single-cell transcriptomics," Nucl. Acids Res. **48**, e55 (2020).
[205]E. Stephenson, G. Reynolds, R. A. Botting, F. J. Calero-Nieto, M. D. Morgan, Z. K. Tuong, K. Bach, W. Sungnak, K. B. Worlock, M. Yoshida, N. Kumasaka, K. Kania, J. Engelbert, B. Olabi, J. S. Spegarova, N. K. Wilson, N. Mende, L. Jardine, L. C. S. Gardner, I. Goh, D. Horsfall, J. McGrath, S. Webb, M. W. Mather, R. G. H. Lindeboom, E. Dann, N. Huang, K. Polanski, E. Prigmore, F. Gothe, J. Scott, R. P. Payne, K. F. Baker, A. T. Hanrath, I. C. D. Schim van der Loeff, A. S. Barr, A. Sanchez-Gonzalez, L. Bergamaschi, F. Mescia, J. L. Barnes, E. Kilich, A. de Wilton, A. Saigal, A. Saleh, S. M. Janes, C. M. Smith, N. Gopee, C. Wilson, P. Coupland, J. M. Coxhead, V. Y. Kiselev, S. van Dongen, J. Bacardit, H. W. King, S. Baker, J. R. Bradley, G. Dougan, I. G. Goodfellow, R. K. Gupta, C. Hess, N. Kingston, P. J. Lehner, N. J. Matheson, W. H. Owehand, C. Saunders, K. G. C. Smith, C. Summers, J. E. D. Thaventhiran, M. Toshner, M. P. Weekes, A. Bucke, J. Calder, L. Canna, J. Domingo, A. Elmer, S. Fuller, J. Harris, S. Hewitt, J. Kennet, S. Jose, J. Kourampa, A. Meadows, C. O'Brien, J. Price, C. Publico, R. Rastall, C. Ribeiro, J. Rowlands, V. Ruffolo, H. Tordesillas, B. Bullman, B. J. Dunmore, S. Fawke, S. Graf, J. Hodgson, C. Huang, K. Hunter, E. Jones, E. Legchenko, C. Matara, J. Martin, C. O'Donnell, L. Pointon, N. Pond, J. Shih, R. Sutcliffe, T. Tilly, C. Treacy, Z. Tong, J. Wood, M. Wylot, A. Betancourt, G. Bower, A. De Sa, M. Epping, O. Huhn, S. Jackson, I. Jarvis, J. Marsden, F. Nice, G. Okecha, O. Omarjee, M. Perera, N. Richoz, R. Sharma, L. Turner, E. M. D. D. De Bie, K. Bunclark, M. Josipovic, M. Mackay, A. Michael, S. Rossi, M. Selvan, S. Spencer, C. Yong, A. Ansaripour, L. Mwaura, C. Patterson, G. Polwarth, P. Polgarova, G. d Stefano, J. Allison, H. Butcher, D. Caputo, D. Clapham-Riley, E. Dewhurst, A. Furlong, B. Graves, J. Gray, T. Ivers, M. Kasanicki, E. L. Gresley, R. Linger, S. Meloy, F. Muldoon, N. Ovington, S. Papadia, I. Phelan, H. Stark, K. E. Stirrups, P. Townsend, N. Walker, J. Webster, A. J. Rostron, A. J. Simpson, S. Hambleton, E. Laurenti, P. A. Lyons, K. B. Meyer, M. Z. Nikolić, C. J. A. Duncan, K. G. C. Smith, S. A. Teichmann, M. R. Clatworthy, J. C. Marioni, B. Gottgens, M. Haniffa, C. I. of Therapeutic Immunology, and I. D.-N. I. of Health Research (CITIID-NIHR) COVID-19 BioResource Collaboration, "Single-cell multi-omics analysis of the immune response in COVID-19," Nat. Med. **27**, 904–916 (2021).
[206]J. P. Bernardes, N. Mishra, F. Tran, T. Bahmer, L. Best, J. I. Blase, D. Bordoni, J. Franzenburg, U. Geisen, J. Josephs-Spaulding, P. Kohler, A. Kunstner, E. Rosati, A. C. Aschenbrenner, P. Bacher, N. Baran, T. Boysen, B. Brandt, N. Bruse, J. Dorr, A. Drager, G. Elke, D. Ellinghaus, J. Fischer, M. Forster, A. Franke, S. Franzenburg, N. Frey, A. Friedrichs, J. Fuß, A. Gluck, J. Hamm, F. Hinrichsen, M. P. Hoeppner, S. Imm, R. Junker, S. Kaiser, Y. H. Kan, R. Knoll, C. Lange, G. Laue, C. Lier, M. Lindner, G. Marinos, R. Markewitz, J. Nattermann, R. Noth, P. Pickkers, K. F. Rabe, A. Renz, C. Rocken, J. Rupp, A. Schaffarzyk, A. Scheffold, J. Schulte-Schrepping, D. Schunk, D. Skowasch, T. Ulas, K.-P. Wandinger, M. Wittig, J. Zimmermann, H. Busch, B. F. Hoyer, C. Kaleta, J. Heyckendorf, M. Kox, J. Rybniker, S. Schreiber, J. L. Schultze, and P. Rosenstiel, "Longitudinal multi-omics analyses identify responses of megakaryocytes, erythroid cells, and plasmablasts as hallmarks of severe COVID-19," Immunity **53**, 1296–1314 (2020).
[207]Z. Zhang, K. Huang, C. Gu, L. Zhao, N. Wang, X. Wang, D. Zhao, C. Zhang, Y. Lu, and Y. Meng, "Molecular subtyping of serous ovarian cancer based on multi-omics data," Sci. Rep. **6**, 26001 (2016).
[208]J. Lee, D. Y. Hyeon, and D. Hwang, "Single-cell multiomics: Technologies and data analysis methods," Exp. Mol. Med. **52**, 1428–1442 (2020).
[209]J. D. Welch, V. Kozareva, A. Ferreira, C. Vanderburg, C. Martin, and E. Z. Macosko, "Single-cell multi-omic integration compares and contrasts features of brain cell identity," Cell **177**, 1873–1887 (2019).
[210]S. Maniatis, T. Aijo, S. Vickovic, C. Braine, K. Kang, A. Mollbrink, D. Fagegaltier, Ž. Andrusivová, S. Saarenpaa, G. Saiz-Castro, M. Cuevas, A. Watters, J. Lundeberg, R. Bonneau, and H. Phatnani, "Spatiotemporal dynamics of molecular pathology in amyotrophic lateral sclerosis," Science **364**, 89–93 (2019).
[211]S. Z. Wu, G. Al-Eryani, D. L. Roden, S. Junankar, K. Harvey, A. Andersson, A. Thennavan, C. Wang, J. R. Torpy, N. Bartonicek, T. Wang, L. Larsson, D. Kaczorowski, N. I. Weisenfeld, C. R. Uytingco, J. G. Chew, Z. W. Bent, C.-L. Chan, V. Gnanasambandapillai, C.-A. Dutertre, L. Gluch, M. N. Hui, J. Beith, A. Parker, E. Robbins, D. Segara, C. Cooper, C. Mak, B. Chan, S. Warrier, F. Ginhoux, E. Millar, J. E. Powell, S. R. Williams, X. S. Liu, S. O'Toole, E. Lim, J. Lundeberg, C. M. Perou, and A. Swarbrick, "A single-cell and spatially resolved atlas of human breast cancers," Nat. Genet. **53**, 1334–1347 (2021).
[212]R. C. V. Tyser, E. Mahammadov, S. Nakanoh, L. Vallier, A. Scialdone, and S. Srinivas, "Single-cell transcriptomic characterization of a gastrulating human embryo," Nature **600**, 285–289 (2021).
[213]M. Zhang, S. W. Eichhorn, B. Zingg, Z. Yao, K. Cotter, H. Zeng, H. Dong, and X. Zhuang, "Spatially resolved cell atlas of the mouse primary motor cortex by merfish," Nature **598**, 137–143 (2021).

[214]See https://www.10xgenomics.com/products/spatial-gene-expression for the visuals on spatial transcriptomics.
[215]Z. Mousavi, M. Kourosh-Arami, M. Mohsenzadegan, and A. Komaki, "An immunohistochemical study of the effects of orexin receptor blockade on phospholipase c-b3 level in rat hippocampal dentate gyrus neurons," Biotech. Histochem. **96**, 191–196 (2021).
[216]See https://biorender.com/ for accessing the mouse brain visual used in this manuscript.