Similarity Metrics for Biological Data

Algorithmic developments for high-dimensional datasets

Ashwin Narayan

B.A. Mathematics and Physics

Williams College, 2016

Submitted to the Department of Mathematics in Partial Fulfillment for the Degree of

Doctor of Philosophy in Applied Mathematics

at the

Massachusetts Institute of Technology

May 2022

Signature of Author: _____

Department of Mathematics
May 2022

Certified by: _____

Professor Bonnie Berger
Simons Professor of Mathematics
Thesis Supervisor

Accepted by: _____

Professor Jonathan Kelner
Professor of Applied Mathematics
Chair, Graduate Committee for Applied Mathematics

Colophon

This document was typeset with the help of KOMA-Script and LaTeX using the kaobook class.
The source code of this book is available at:
`https://github.com/fmarotta/kaobook`

Similarity Metrics for Biological Data

by
Ashwin Narayan

Submitted to the Department of Mathematics
on May 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Applied Mathematics

Advances in experimental methods in biology have allowed researchers to gain an unprecedentedly high-resolution view of the molecular processes within cells, using so-called single-cell technologies. Every cell in the sample can be individually profiled — the amount of each type of protein or metabolite or other molecule of interest can be counted. Understanding the molecular basis that determines the differentiation of cell fates is thus the holy grail promised by these data.

However, the high-dimensional nature of the data, replete with correlations between features, noise, and heterogeneity means the computational work required to draw insights is significant. In particular, understanding the differences between cells requires a quantitative measure of similarity between the single-cell feature vectors of those cells. A vast array of existing methods, from those that cluster a given dataset to those that attempt to integrate multiple datasets or learn causal effects of perturbation, are built on this foundational notion of similarity.

In this dissertation, we delve into the question of similarity metrics for high-dimensional biological data generally, and single-cell RNA-seq data specifically. We work from a *global* perspective — where we find a distance function that applies across the entire dataset — to a local perspective — where each cell can learn its own similarity function. In particular, we first present Schema, a method for combining similarity information encoded by several types of data, which has proven useful in analyzing the burgeoning number of datasets which contain multiple modalities of information. We also present DensVis, a package of algorithms for *visualizing* single-cell data, which improve upon existing dimensionality-reduction methods that focus on local structure by accounting for density in high-dimensional space. Lastly, we zoom in on each datapoint, and show a new method for learning k-nearest neighbors graphs based on local decompositions.

Altogether, the works demonstrate the importance — through extensive validation on existing datasets — of understanding high-dimensional similarity.

Thesis Supervisor: Bonnie Berger
Title: Simons Professor of Mathematics

Acknowledgements

Writing a dissertation has been a uniquely rewarding endeavor, and like all such endeavors it has truly taken a village — in fact, virtually all of the fondest memories from the past several years involve the people below. I saved writing this section until the very end because the joy of reminiscing made the best coda to the adventure.

First and foremost, I owe my deepest thanks to my advisor, Bonnie Berger. When I barged into her office at the start of graduate school, knowing nothing about biology — or statistics or machine learning for that matter — she agreed to advise me, and I will be forever grateful. Her ability to connect disparate threads of ideas and shepherd them into a concrete project, to find the intersection between beautiful mathematics and useful biology, and, most importantly, to instill a sense of confidence, belief and self-worth to impostor-syndrome laden graduate students will continue to inspire me.

I also thank my other committee members, Jon Kelner and Bryan Bryson, whose research interests and styles have informed my own. Most of what I know about algorithm design I learned in Jon's course, and his influence is easily seen in the work presented here. Bryan's work, meanwhile, has been a wonderful template for how to find interesting biological questions to apply those algorithms to.

I would never have considered myself a potential math major, much less a candidate for a PhD in mathematics, without the support, guidance, and infinite encouragement of my undergraduate advisors. Julie Blackwood, Steve Miller, David Love, Mihai Stoiciu, to name only a few of the many, many professors who I am indebted to — and of course, Bill Wootters, my first thesis advisor; it was such an honor to work under so accomplished and so humble a scientist.

My peers and colleagues deserve far more thanks than I can write here. To the other members of the Berger group and my fellow graduate students in the math department, it was a pleasure to be among such intellectually stimulating company; to count your colleages as your closest friends is a rare privilege. And to those friends — from college days or childhood, from adventures in the mountains and at the crag — I thank you all for providing me the escape from work that was so crucial to my mental and emotional health.

And lastly, I thank my family. My parents, Narayan and Priya, for their uncoditional love and support, for instilling in me a love for learning, for being there to cheer me on at every juncture, from interminable cross country meets to a thesis defense that must have felt the same way. And my sister Anusha, a life-long friend, an inexhaustible source of joy and perspective.

Contents

List of Figures

List of Tables

1 Introduction

The field of biology in recent decades has been completely revolutionized by a profusion of data. In a virtuous cycle, the availability of computational resources has spurred methods development: experimental methods generating ever more data, and computational methods drawing deeper and deeper insights — thereby encouraging experimentalists to pursue more ambitious methods.

Almost any type of biological data, from the number of genomes sequenced, to the number of cells that can have their cellular products *individually* processed, to the number of known connections between neurons, show exponential growth (see Figure 1.1).

This dissertation aims to contend with the explosion of complexity by approaching what is, at heart, a very straightforward question: **How can we tell when two pieces of biological data are similar to each other?** In fact, it is also one of the fundamental questions of the field. The domain that a computational biologist works in is composed of numerical representations of physical data — translating questions

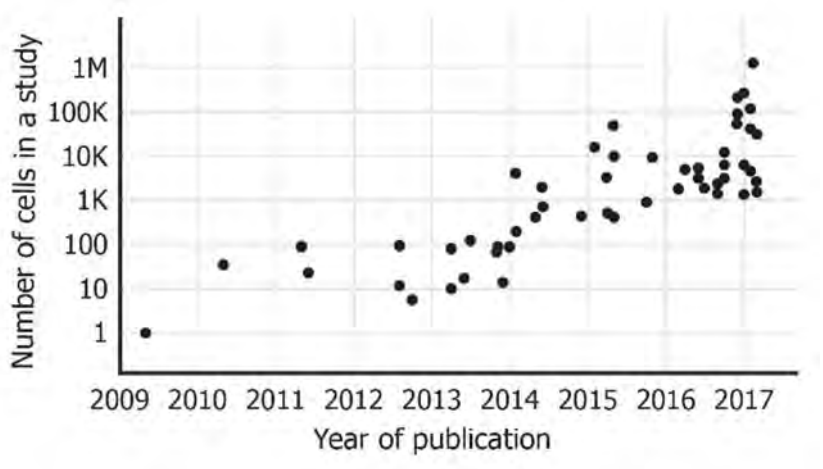

Figure 1.1: Moore's law in biology The largest sizes of single-cell RNA-sequencing datasets have been drastically increasing, super-exponentially over the past decade; this is drawn from Cho, Berger, and Peng [1]

in the physical world to their analogues in numerical world and then translating the numerical insights back is one potential job description for a computational biologist. Many of these questions, as we fully discuss in Chapter 2, either implicitly or explicitly depend on understanding biological similarity. We hope to show, through the work we describe here, that answering this main question actually requires a deep understanding of the biological processes that generate the data *and* the mathematical models that act upon it.

From a historical perspective, parsing an explosion of data is not a fundamentally new challenge for the field of biology, which has long relied on computational insight for categorizing and making sense of the data it collected. Markowetz [2] makes the point that if Carl Linnaeus, considered the father of modern taxonomy, was around today, "he would be a computational biologist". Gregor Mendel's genetic analyses — when couched in terms a machine learning practitioner would understand — involve fitting an empirical distribution to expected counts from a generative model[1].

1: The model is decidedly a straightforward one, but nevertheless a generative model

As far as historical shadows go though, one must consider the turn of the twentieth century, which saw population geneticists develop much of the machinery of classical statistics — but in the service of eugenics and justifying colonialism[2] [3]. I bring this up as a shadow not to disparage the potential benefits of bioinformatics but rather as a stark reminder that any project that aims to understand human differences can be fraught and must be undertaken carefully. As we will find in the work presented in this dissertation, the prospect of personalizing biology to the individual is indeed one of the great promises of the data we work with.

2: This shows that even the field of ethical AI is not fundamentally new. The project of justifying racism with the sterile language of statistics was foundational for population genetics and its spectre continues to be apparent in modern genomics.

Turning to more recognizably modern bioinformatics, we find in the mid- and late-twentieth century the advent of computer science and the paradigm of the biological organism as an information-processing system, with its information stored in *sequences*[3]. For one, Jacob and Monod [4] in 1961 began to describe simple regulatory circuits in *E. coli*, revealing feedback mechanisms that could theoretically be programmed. Because of the central dogma of biology (see Section 2.1) it was known[4] that these regulatory mechanisms must be encoded in strings — which was exactly how theoretical computer scientists were thinking about computers [5] [5].

3: We discuss these sequences in detail in Chapter 2

4: perhaps "posited" is more accurate than "known", as we continue to learn that, in a theme throughout this work, biology is complicated.

5: The computational construct of a Turing machine certainly recalls the biological process of translation, which we will discuss below.

Human Genome Project: a massive project running from 1990 to 2003 with the goal of sequencing the entire human genome

It was developing algorithms for processing strings — reconstructing, matching, imputing, finding an correcting errors— where biologists confronted their next era of Big Data: the **Human Genome Project** (HGP) forced computer scientists and biologists to figure out how to piece together and match millions of short sequences of DNA.

The wealth of sequencing data that became available through the HGP (and through similar sequencing efforts for other organisms), made it routine to find the analogues of particular sequences across organisms.

The HGP provides yet another historical silhouette — in its infancy, the sequencing of the human genome was considered the Holy Grail for not just biology, but personal medicine. While the **nature versus nurture** dilemma was certainly not settled, it was thought that the genome would unveil *all* the secrets of, for example, human disease: knowing your genetic sequence would allow you to map out a great deal of your health[6]. In hindsight, it is perhaps unsurprising that such a panacea did not pan out. As we discuss in Subsection 2.1.4, we keep finding more and more complications and noise.

nature versus nurture: the question of how much of an organism's appearance or behavior are determined by its genes or its environment

6: In a sign of the zeitgeist, the film *Gattaca*, which considers just such a dystopian future, was a hit back in 1997

Today's data are often attached to similar promises. Now, rather than having a genome for every individual, biologists are able to peer even deeper: into *each cell* within that individual, into each copy of each protein found in that cell. Again, the point here is not to disparage the new sources of data[7], it is to once again highlight that the layers of complexity are only just being pared.

7: This dissertation is the best evidence for my faith in the value of these high-resolution data

In fact, analyzing the troves of data that modern bioinformaticians must contend with has led to great insights but also continues to reveal that biology is intensely complicated. Each new type of data or inference method seems to lead to new questions — a bonanza for new computational biologists. For example, we have realized that the regions of so-called "junk" DNA actually serve regulatory purposes and that proteins have regions of continuous variation and RNA can itself have secondary structure[8].

8: Don't worry if these words don't mean anything, all shall be explained!

The work presented here took place entirely upon the shoulders of the giant described above. I certainly do not claim to have "seen farther" but everything done here would have been impossible without the groundwork, in statistics, computer science, and biology built up over the twentieth century. The contribution here attempts to join some threads and understand biological similarity with the language of mathematics.

The structure of the dissertation is as follows. We begin with an in-depth overview of exactly those biological processes and mathematical models in Chapter 2, and then we move into our research contributions. Thematically, our work is ordered going from *global* scale to *local* scale, and our algorithms build from global linear transformations to very local nonlinear transformations. In Chapter 3, we begin by discussing alignment of multimodal single-cell technologies by finding a linear transformation of the original data

that mediates between the distance metrics induced by the different datasets. In Chapter 4, we move towards nonlinear transformations of scRNA-seq data, delving into the objective functions of UMAP and t-SNE[9] and determining that they do not do a good job of preserving high-dimensional density. Each chapter is paired with a chapter that focuses on the extensive biological validation — Chapter 5 for metric alignment, and Chapter 6 for dimensionality reduction. Finally, in Chapter 7, we move towards the construction of the k-NN graphs themselves that underlie the methods and evaluation in the previous two chapters, and explore new methods for creating better k-NN graphs.

9: These are the two most popular methods for visualization of scRNA-seq data, as discussed in Section 2.6.

2 Background

The education of a computational biologist covers a broad range of disciplines. It is imperative for us to understand the experimental methods that generate the matrices that cover our computers, so generative models can be properly calibrated to underlying biological processes. One must also learn the questions that biologists want to be able to answer with the data that they have: numbers can be crunched in any number of ways, but are most useful in the service of understanding or clarifying actual biological hypotheses. Lastly, of course, is the ability to convert questions from biology into algorithmic or statistical questions, the technical ability to actually *answer* those questions, and then translate the insights back into biology.

We attempt in this chapter to lay the above groundwork — describing biological systems we evaluate, the experimental procedures that make those biological systems interrogable, building the map between biological and algorithmic questions, and describing how we can evaluate effectiveness of algorithms.

1: to simplify to an almost irresponsible degree

2.1 Central Dogma of Biology

Molecular biology — and consequently, computational molecular biology — has[1] been built around understanding the **central dogma of biology**, which describes the overarching mechanisms by which information flows through organisms at the molecular level. Broadly, the central dogma says that information flows from DNA (deoxyribonucleic acid) to RNA (ribonucleic acid) to protein. We briefly review each of these crucial macromolecules in turn.

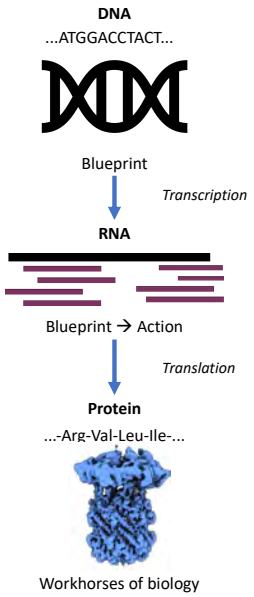

Figure 2.1: The central dogma of biology

2.1.1 DNA

DNA acts as the blueprint for the cell and the organism: each cell in the organism has the same copy of DNA[2]. DNA is a **polymer** made up on monomers called *nucleotides*, of which there are four: adenine, thymine, cytosine, and guanine (see Figure 2.2). Famously, DNA is double-stranded, and the strands are held together by hydrogen bonds between the nucleotides — adenine binds to thymine and guanine to cytosine — so each strand contains the information necessary to replicate the other. When cells replicate, this redundancy is crucial — the DNA strands are separated, and a new strand is synthesized based on the complementarity of the nucleotides.

2: Biologists continue to discover just how much of a simplification the central dogma is. Rather than jamming the narrative with all the caveats, we will use these sidenotes to discuss exceptions. For example, we note here that *germ* cells (sperm and eggs) do not have the same copies of DNA as *somatic* cells.

polymer: a molecule that is made by combining smaller subunits, called monomers

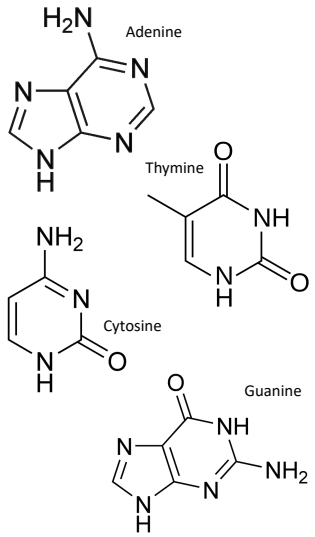

Figure 2.2: The four nucleotides

The sequence of nucleotides provides the "recipe" for the creation of proteins, the workhorses of biology (see Subsection 2.1.2). In regions known as **coding regions** [3], each triplet of nucleotides encodes one of twenty possible molecules called **amino acids**, which are the building blocks of proteins.

Decoding DNA was one of the earliest tasks for computational biologists: from synthesizing the full string based on short sequenced fragments, to figuring out which regions encode proteins, and which regions are conserved across different species and organism. Answering these and related questions is the remit of the field of **genomics**.

3: Of course, this begs the question about what *else* is there — the jury is still out about exactly non-coding regions are for, but it is clear they serve important regulatory roles

2.1.2 Protein

As introduced above, proteins justly are considered the building blocks of biology and mediate a wide, if not exhaustive, array of biological functions, serving as enzymes for catalyzing reactions, physical building blocks for cellular structures, signals for cellular communications, channels for transporting metabolic products, and the list goes on.

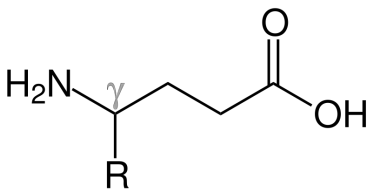

Figure 2.3: Molecular structure of an amino acid

Proteins are also polymers, and their monomers are called amino acids, which have a backbone of carbon and nitrogen along which they are strung together, and a variable group, of which there are twenty found in nature.

The thermodynamic interactions between the amino acids, especially between the variable groups, cause the one dimensional string[4] of monomers to fold into a three-dimensional structure.

4: Some proteins are made up of multiple strings which fold together. Also, there exist proteins that help a protein fold *just so*, when thermodynamics alone isn't enough.

The end structure of a protein is crucial. For enzymes, the **active site** must conform exactly to the molecules whose reaction is being

catalyzed; the microtubulules that give structure to a cell must bear its weight; cell signaling and regulatory proteins must assume their active configuration in the presence of the correct molecular factors; and channel proteins must selectively passage the correct set of cellular products, under the correct circumstances.

active site: the physical site of the enzyme where catalysis happens

Because of the centrality of protein structure, it has long been a goal of computational biologists to figure out the eventual three-dimensional structure of proteins, whether from just the amino acid strings themselves, or through two dimensional images take through electron microscopy. Studying the proteins in a cell is the realm of **proteomics**.

2.1.3 RNA

Ribonucleic acid (RNA) is the molecule that mediates the transformation of information stored in DNA into the amino acid sequences required to build proteins. Notably, while every cell has the same DNA content, different cells in an organism perform vastly different functions by expressing vastly different proteins. Understanding the RNA content of a cell is key to understanding that cell's specific role in an organism. Because of this, keystone role, this molecule is also the main focus of this dissertation.

There are many different types of RNA[5] — in fact the broadly accepted "RNA worlds" hypothesis posits that RNA was once the *only* macromolecule, performing storage and enzymatic functions on its own. Our primary focus here will be messenger RNA (mRNA), which is a **transcribed** (i.e. copied) version of the coding regions of DNA[6]. The mRNA transcripts are then *translated* into amino acid strings[7]. Profiling the mRNA transcripts in a cell thus informs which proteins (and how many copies) will be synthesized.

The diversity of RNA's functions and forms is truly staggering, and it is no wonder that the study of RNA has greatly benefited from computational biology's Big Data era. In addition to mRNA, the central dogma is mediated by **transfer RNA** (tRNA), which actually translates between nucleotide triplets and the particular amino acid they represent by binding specifically to each; and **ribosomal RNA** helps build the **ribosome**, the cellular structure that serves as the site of protein synthesis. More recent work has also discovered regulatory functions for RNA, including **microRNA** and **small-interfering RNA** (siRNA), which help tag mRNAs and other transcripts for degradation.

5: And researchers continue to find more and more types and roles for this molecule. We will discuss only a subset here.

6: Thus **transcriptomics**, the study of the mRNA content of cells, joins the extensive and ever growing list of *-omics* suffixed data types

7: Technically, these are called *polypeptides*, which are the precursor to proteins, but we will use them interchangeably.

2.1.4 Complications to the central dogma

As mentioned above, the central dogma is an extreme oversimplification of the actual process of protein synthesis. Just as Newtonian mechanics in physics or stylized supply-and-demand graphs in economics capture the gist of a labyrinthine reality, the neat **DNA** → **RNA** → **protein** schematic is a useful introduction for non-experts to get a toehold into biology. While understanding the central dogma should be sufficient for the work presented here, we briefly highlight some of its shortcomings for the interested reader.

Alternative splicing

eukaryote: organisms whose cells have individual organelles, e.g. animals and plants; the opposite are **prokaryotes**, like bacteria

The strand of RNA that is transcribed from the coding region of DNA is, in **eukaryotes**, *not* immediately then translated into a protein. The coding region is split into *exons*, which are actually translated, and *introns* which are spliced from the RNA. Crucially, the same transcribed region can have multiple possible splice sites, so the the same region of DNA can lead to many different possible proteins, depending on exactly which pieces are spliced out. This is known as *alternative splicing*, and researchers are actively trying to understand the process that mediates the different splicing possibilities.

Epigenetics and chromatin accessibility

epigenetics: the study of the interaction between environment and expression

chromatin: the combination of DNA and protein that constitute a chromosome

As discussed in Subsection 2.1.3, different regions of the DNA are expressed in different cells, and **epigenetics** tries to understand the mechanisms behind these differences. While the search for these mechanisms continues, one mechanism that is relatively well understood is **chromatin** accessibility: if a region of DNA cannot *physically* be accessed by the enzymes necessary for transcription, then any proteins encoded in that region will not be expressed. Proteins called **histones** are responsible for the packing of chromosomes, and the interplay between these packaging proteins, transcription factors, chromatin binding proteins, and other regulatory elements — all heavily mediated by external and environmental stimuli — creates a dynamic and complex set of conditions for chromatin accessibility [6].

Non-coding regions

Only about one percent of the DNA in the human genome actually encodes proteins [7]. We have discussed one portion of the non-

coding regions above — the introns, which are regions within a primary mRNA transcript that are spliced out before the translation stage. Researchers long thought that the "rest" of the DNA between the transcribed regions (called the **intergenic region**), served no purpose and was thus called junk DNA. However, work by **ENCODE** [8], has found that much of these intergenic regions also serve crucial regulatory roles. Some regions, like **promoters** and **enhancers**, serve as binding sites for important proteins, whereas others transcribe the various types of RNA discussed in Subsection 2.1.3. Still other non-coding sites do not have a known function but heterogeneity in those sites is associated with different phenotypes, like disease.

ENCODE: the **Enc**yclopedia **o**f **D**NA **E**lements, a project with the goal of learning the function of the entire human genome

promoter: DNA region where transcription can start

enhancer: DNA regions that transcription factors can bind to to increase likelihood of transcription of a gene

Polypeptide processing

Even when alternative splicing is taken into account, the mRNA that remains does *not* correspond one-to-one with the protein that emerges in the cell[8] [9, 10]. Once the polypeptide has been generated, it undergoes a number of potential different processing steps before it becomes a "production-ready" protein. Most straightforwardly, many proteins are made up of a composition of several polypeptides [11], and several other proteins are involved in putting these complexes together [12].

8: This is where the earlier distinction between protein and polypeptide becomes meaningful

Other major types of processing involve additions made to the single polypeptide chain itself [14]. *Phosphorylation* involves adding a phosphoryl group to a protein, which often activates that protein for signaling mechanisms (a phosphorylated protein can activate other proteins); *glycosylation* similarly involves the addition of small molecules called *glycans*, which heavily affect the folding and tertiary structure of a protein; proteins destined for membranes[9] often have *lipid modifications* — lipids attached covalently to the protein; the addition of the protein *ubiquinone* often actually marks a polypeptide for degradation. All of the above processes are highly regulated and driven by various (and still poorly understood) temporal and environmental stimuli.

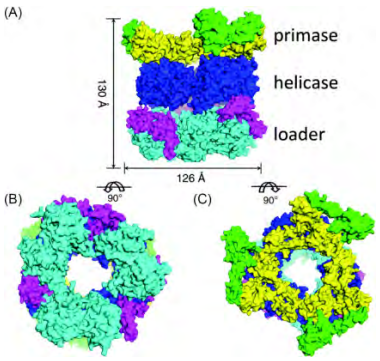

Figure 2.4: An example of a *purinosome*, a multi-protein complex, taken from Roy and Kundu [13]; each color represents a separate peptide

9: for example, acting as channels through membranes

2.2 Single-cell RNA Sequencing

Recent advances in experimental methods have given biologists an unprecedentedly high-resolution view into the transcriptomic profiles of individual cells. Single-cell RNA sequencing (scRNA-seq) is a sequencing technology that, for each cell in a sample, counts the

amount of each particular gene transcript that is expressed in that cell.

The resulting data gives us a so-called *expression profile* for each cell. Under the hypothesis that cellular function is determined by the proteins present in the cell, the differences in expression profiles help us understand the underlying mechanisms for cellular differences.

While new methods for scRNA-seq are actively being developed, they follow a similar framework: RNA fragments that are complementary to the transcripts of interest are first prepared. Then, cells in the sample are individually[10] combined with sets of these complementary transcripts, and each set is identified with a unique "barcode". In this way, the sequenced transcripts can be traced back to the cells that they came from.

10: For example, in *droplet based* sequencing methods, cells fed through a stream of oil droplets, so each oil droplet should contain a single cell.

Because of its incredibly high-resolution, scRNA-seq datasets and corresponding methods for analysis have proliferated. For example, recent experiments include profiles of the various immune cell types known as *peripheral blood mononuclear cells* (PBMCs), which are crucial to our immune response [15]; a look into each and every cell in the roundworm *C. elegans* embryo [16]; and a comparative view into the profiles of cells in tumors versus in healthy blood [17].

2.2.1 Multimodal single-cell technologies

While mRNA transcripts remain the most widely profiled type of single-cell data, various other single-cell technologies are proliferating. *Slide-seq* is a technology that records the *spatial* location of cells in a sample, leading to insights into cellular and tissue organization. ***Ch**romatin **I**mmuno-**P**recipitation*-sequencing (ChIP-seq) is a method to determine where **transcription factors** bind onto the DNA [18]. *Chromatin accessibility* profiling is a form of single-cell epigenetics, and records which regions of the DNA are actually accessible in a given cell, allowing researchers to understand *why* different cells have different transcriptomic profiles. *Single-cell metabolomics* moves further down the pipeline of the central dogma, and looks at the **metabolites** present in each cell, to understand from a process-based standpoint exactly what reactions are going on in which cells. Combining the information from all these single-cell approaches is a space that is rapidly proving to be extremely informative, and the so-called *multi-omic* datasets combine information from several different data types [9].

transcription factor: protein complexes that attach to the DNA and mediate transcription into RNA, for example accelerating or repressing expression of a gene

metabolite: the small-molecule-sized inputs and outputs of cellular processes

These data are ripe for analyzing computationally, and we will discuss single-cell technologies from a computational standpoint next.

2.2.2 A survey of scRNA-seq problems

The promise of high-resolution insights through single-cell technologies is paired with the unique computational challenges of analyzing these high-dimensional data [19].

- **Preprocessing:** Cleaning up the raw data is itself a non-trivial task when it comes to scRNA-seq data[11]. The experimental techniques used for generating the data can lead to corrupted data — sometimes two cells are mapped as a single-cell (these are called **doublets**) or sometimes the cells captured are dying. Even accounting for these, the sequencing depths of the different cells can be on different scales, and so normalization procedures are required to be able to compare the outputs of different cells.
- **Cell-type identification:** As noted above, one of the key questions that single-cell analyses can answer is, *what makes different cells look different from each other?* The first step is of course to identify *whether* there are different classes of cells within the dataset. The cell-type identification problem involves taking an unlabeled scRNA-seq dataset and determining whether the data can be divided into discrete classes, and what class each cell belongs to. In computational terms, this is done by *clustering* (see Subsection 2.3.1 for an overview of methods), and clustering has become a *de facto* first step for many analysis pipelines. A major difficulty in the clustering of scRNA-seq data in particular is the high-dimensionality of the data — with potentially tens of thousands of genes, figuring out which genes we should focus on to inform clustering remains an open and active area of research (see Chapter 7 for our work on the subject)[12].
- **Gene set enrichment:** Identifying different coherent cell-types, while crucial, is only part of the picture. It is also important to understand what makes the cell-types different transcriptomically, by looking at the transcripts that are disproportionately expressed in each of the cell-types that the clustering methods may have found, a process called **gene set enrichment** or **differential gene expression**. Identifying exactly what makes the expression of a gene in a cell-type distinctive is not necessarily straightforward: methods generally consider the subset

11: We will mostly discuss problems related to RNA-sequencing here, but many of these also apply to the other single-cell methods we've discussed.

12: Another open question, which we touch on in Section 2.7 is whether discrete cell-types exist *at all*. In other words, is the fact that cells have different expression patterns indicative of noise around some core expression profile, or is there value in dealing with the core separately from the periphery around a cluster?

of genes within a cell type that have higher mean expression within the cell type than overall in the dataset, but one might also be interested in *variance* of expression, a concept we consider in Chapter 4.

- **Visualization:** The high-dimensional nature of single-cell data also makes it very difficult to understand, since high-dimensional data in general have some very unintuitive properties[13]. Communicating insight about these datasets thus requires tools for visualization, and the state of the art tools use *dimensionality reduction* in order to reduce the original data to a two- or three-dimensional dataset, which can then be interpreted via scatter plot. See Section 2.6.
- **Trajectory inference:** Understanding the differential gene expression of the different cell types in a dataset motivates another dimension of analysis: temporal variability. Since the multicellular organisms mostly studied in scRNA-seq experiments grow from embryonic cells dividing and differentiating, the temporal dynamics of gene expression can help understand the differentiation process. However, since the act of sequencing destroys the sample, one must infer the the dynamics — the collection of trajectory inference algorithms aims to learn, given the expression vector of the cell, which cell it is moving towards and which cell was closest to its progenitor's expression [21].
- **Integration:** The problems above were all focused on analyzing a *single* dataset. With single-cell sequencing becoming more and more part of the molecular biologist's toolkit, we often are faced instead with multiple datasets, both many scRNA-seq datasets of the same cell-types or multiple types of single-cell data (e.g. ChIP-seq, Slide-Seq, and scRNA-seq), which we call *modalities*. To analyze these datasets together, it is important to figure out how to synthesizes the information they contain. In the single-modality case (i.e. many scRNA-seq datasets), the difficulty is *batch effects*, where experiment-specific noise artifacts can make it seem like the cells within a dataset are more similar to each other than to the proper cell-type. Many methods attempt to correct for these batch effects by trying to "anchor" cells of the same type together across datasets [22, 23]. The case of multiple modalities presents additional problems, as one must try to reconcile the different information provided by each of the different modalities: if two cells are far away from each other in expression space but close by in chromatin-accessibility, what does that tell us about their similarity? We aim to address these questions in Chapter 3.

13: The archetypal example is that a high-dimensional Gaussian point cloud has most of its mass *not* in its center but rather in a ring around it [20].

As we now move towards the mathematical machinery relevant to this work, the reader should keep the above problem types in mind. Notably, the bulk of our mathematical discussion will focus on the theme of *distance metrics*, namely, how to tell whether two expression[14] profiles are close to each other in some biologically meaningful way. We should note that *all* the problems above, with the exception of preprocessing, are heavily reliant on the ability to assign similarity between cells; even the preprocessing step is done with a focus on getting the cleanest possible data to be *able* to make those similarity judgments.

14: or ChIP-seq or slide-seq or any other single-cell vector

2.3 Mathematical Setup

From a mathematician's standpoint, an scRNA-seq dataset[15] with N cells and G unique transcripts[16] of interest can be represented as a matrix $X \in \mathbb{R}^{N \times G}$, where X_{ij} represents the number of transcripts of gene j found in cell i[17]. So a cell can be thought of as a vector in $\mathbb{R}^G$.

Analyzing a scRNA-seq dataset thus entails understanding *distances* between the rows of this data matrix X. A fundamental question — *the* fundamental question in this dissertation — is thus deriving a biologically meaningful similarity score between two cells, given the expression vector of a pair of cells.

15: While we focus on scRNA-seq data here, most single-cell technologies can be easily dealt with analogously.

16: The astute reader familiar with scRNA-seq will recognize that we did not use the word *gene* here — or for that matter, at all so far! This is intentional, as a solid definition of gene is hard to find. For this work, we will use *gene* to mean "unique transcript of interest in an scRNA-seq dataset".

17: One might ask, why $\mathbb{R}$ instead of $\mathbb{N}$, if the data are count data. Often the count data are transformed, so we leave the domain general

2.3.1 Machine learning

Machine learning (ML) has become a crucial tool for the computational biologist, but it is often surprisingly hard to find an operational definition. We generally take the approach of definition by enumeration — discussing *supervised* and *unsupervised* tasks. But we first attempt a more overarching view.

Generally, the unifying theme behind ML algorithms is *data* — if it is about anything, machine learning is about inference through data[18]. We might go even farther and claim that, given some data $\{X_1, \ldots, X_N\} \subset \mathcal{V}$, ML involves learning some function $\phi : \mathcal{V} \to \mathcal{Y}$ such that $y_i = \phi(X_i)$ aids in some decision-making process. While we expand on this in more detail presently, we can quickly see that some canonical machine learning tasks fall into this framework:

18: One might ask, isn't *statistics* the art of inference through data? It is an open question what separates ML from statistics.

Examples of ML tasks

- **Image classification:** Given a set of images of animals, some of which are cats and the others of which are dogs (and several of which are labeled as such), decide if a given unlabeled image is a cat. Here, $\phi : \{\text{Images}\} \to \{\text{cat}, \text{dog}\}$, and the output of the function itself is the decision
- **Clustering:** Given a collection of crabs[19], figure out whether the crabs are from separate species. Here again, ϕ is a function that takes as input several characteristics of the crabs and labels a particular crab, but we additionally have to figure out what the possible set of labels actually is.

19: Yes, the organism — it was in solving a problem very similar to this that the *method of moments*, a classical technique in statistics, was developed

Of course, the above two examples are the archetypes of the split between supervised an unsupervised learning, and without further ado, we now elaborate on these terms.

Supervised learning

The crucial aspect of the data required for supervised machine learning is *labels* — that is, we have some data for which we we *know* the value of the aforementioned function ϕ[20]. The goal of supervised learning is to use the labeled examples to learn the functional form of ϕ, so it can be applied to the unlabeled data whose labels we want to know.

20: The labeled data are often called *training data*

Further organization of the this class is based on the type of the labels: if the labels are discrete (e.g. $\{\text{cat}, \text{dog}\}$ from the example above), then the problem is known as **classification**. If the the labels are continuous (say, predicting the weight of said cat based on the image), then the problem is called **regression**. While the "discrete" versus "continuous" nature of the problem is often used as the baseline to split classification versus regression, and is generally sufficient for our purposes here, the reality is slightly more complicated. A more salient difference is that for a regression problem, the possible labels have a metric[21], and so, some labels can be closer to each other than others.

21: This will be defined in Section 2.4, but it is basically a way to measure distance between values.

The difference is patently obvious in the examples above: weights of 20 lbs and 25 lbs are much closer to each other than weights of 10 lbs and 30 lbs. Conversely, if we label images by animal type, the image either has a particular label or it does not[22].

22: The example of animals is specifically chosen because it introduces an interesting middle-ground — if the labels are **cat**, **dog**, and **spider**, then surely the distance between **cat** and **dog** is closer than that to **spider**. This realm of hierarchical classification is getting closer to regression!

In designing the solution to the supervised learning problem, the key decision that the practitioner must make is the functional form

of ϕ and how it interacts with the input data X. Most well-studied of course is the class of *linear regression*, where we write:

$$\phi(x) = Ax + b$$

and find the matrix A and vector b that best match the labeled x_i to their label[23] y_i. The process by which A is found is called **learning** or **fitting**, and is in general found by minimizing some notion of error, which is usually called a **loss function** or **objective**. For example, in the case of least squares regression, this error is:

$$\min_A \mathcal{O} = \sum_{i=1}^{N} (y_i - [Ax_i + b])^2, \tag{2.1}$$

which is called the **mean-squared error** (MSE).

23: It should be noted that while much of modern ML has moved beyond linear regression, the framework is actually extremely powerful and can accommodate nonlinearities — interested readers should study *ordinary least squares* (OLS)

The flexibility of parametrized supervised learning is thus signposted here: replacing the matrix A with a general function from richer function classes, and tailoring the loss function to achieve the desired objective are the main knobs available for solving a supervised learning problem[24]. In Subsection 2.3.2 we discuss a further degree of freedom, whereby the input points can be transformed into some new nonlinear space using the so-called **kernel trick**.

24: The reader may be curious why *deep learning* hasn't merited a discussion yet. Deep learning is merely one specification of the supervised learning problem, where the function class takes the form a neural network, and the method of optimization cleverly minimizes a given objective function.

Unsupervised learning

Most of the work presented here is in the more nebulous realm of unsupervised learning, where the data we are presented are *not* labeled in any way. The array of tasks in this category focus on finding some kind of inherent structure in the data can be learned without knowledge of ground truth labels. In our generalized framework for ML above, we noted that even in unsupervised learning, the goal is to devise some kind of transformation ϕ that aids with a decision-making process. The key in unsupervised learning is that we do not know the values of ϕ for any of our existing data. While not exhaustive, some of the main tasks in unsupervised learning are clustering, metric learning, and dimensionality reduction.

Unsupervised learning tasks

- **Clustering:** Insofar as one exists, this is the classical unsupervised learning task. The goal is to understand whether your underlying data actually *do* fall into some discrete classes (as one might have in the classification problem

above). Clustering methods have have been an active field of development for decades[25], and the crux of the problem is deciding what constitutes a cluster. Some look at density — clusters are regions of high density surrounded by low-density peripheries [24]; others posit that clusters should be sets of points that are much closer to each other than to other points [25]; still others assume the data come from an underlying mixture of probability distributions and find the assignments that maximize the **likelihood** of the data [26].

- **Dimensionality reduction:** The goal of dimensionality reduction (DR) is to find a lower-dimensional representation of the input data. So here, ϕ is a function that maps the input vector x to a lower dimensional vector y. One can use DR to de-noise the data, to visualize it, or to learn correlations between the input features. We cover DR in detail is Section 2.6.
- **Metric learning:** The metric learning task is to find the right notion of distance between the input data points. This is a really important task in ML that often takes on many names and has many approaches, and we discuss a few of those approaches in Subsection 2.3.2 (which is about kernel methods), Section 2.4 (the general metric learning problem), and Section 2.5 (which is about the specific and crucial case of *manifold learning*).

25: *Centuries* might be more accurate — the "classifying crabs" example above was from the early twentieth century, and the method-of-moments might actually predate this, to, you guessed it, Gauss.

likelihood: given a particular probability distribution, the probability density of seeing a particular data point.

2.3.2 Kernel methods

As alluded to in our discussion of supervised learning, linear regression and its related linear methods have more power than is often ascribed to them. This is because one can transform the input vectors with some nonlinear functions and then apply linear regression on *those* functions. For example, given an input vector:

$$x = (x_1, \ldots, x_k)^T$$

and a label $y \in \mathbb{R}$, the traditional linear regression problem learns a vector w and sets[26] $y = w^T x$. However, if we think that actually, y depends on the *square* of one of the features, and perhaps the product of another pair[27], we could come up with a *new* vector:

$$x' = \psi(x) = (x_1^2, x_1 x_2, x_1, \ldots, x_k)^T \tag{2.2}$$

26: We are ignoring the constant b from above here.

27: This is called an *interaction term* in **econometrics**

and a then learn a *new* weight vector $w' \in \mathbb{R}^{k+2}$ — the key is that under the hood, this remains a linear regression problem, despite the "new" features having non-linear dependencies between each other!

Kernel methods generalize this idea, by allowing the input vector to *first* be transformed into some nonlinear feature space, and the predictor to be applied in that space. The difficulty here is that the data need to be transformed into the feature space, which is potentially very high dimensional. For example, consider the natural extension of (2.2) above to *all* quadratic terms:

$$x' = \psi(x) = (x_i x_j)_{i \leq j \leq N}^T \tag{2.3}$$

This requires learning a weight vector $w' \in \mathbb{R}^{k(k-1)/2}$ — and this scales as $O(k^r)$ for learning interactions up to the power r.

This is where the *kernel trick* comes into play. The linear regression problem can be converted to a problem that is solved only in terms of the matrix $G = XX^T$, which is called the **Gram matrix** — which means only the elements of the Gram matrix need to be learned. Note that

$$G_{ij} = x_i^T x_j$$

In the case of the transformed vector $x' = \psi(x)$ above, the Gram matrix becomes $G = \psi(X)\psi(X)^T$, where

$$G_{ij} = \psi(x_i)^T \psi(x_j)$$

Thus, instead of learning the $O(k^r)$ terms necessary to write out ψ, we need to learn the $k(k-1)/2$ pairwise dot products in the kernel[28]. In fact, just *defining* a kernel implicitly transforms the given problem into some complex feature space.

28: In the case of $r = 2$ (i.e. all quadratic terms), the kernel method is not more efficient, but it will be as r increases

In this work, we rely heavily on kernel methods, as they are key to learning useful distance metrics on biological data. Especially when the *local* distances in a dataset are most important, the kernel trick is closely associated with another crucial structure in ML, the **graph**, which we introduce next.

2.3.3 Graph theory

Long a staple of algorithms in theoretical computer science or for applications to routing and scheduling — namely **combinatorial** tasks — **graph theory** has become a crucial aspect of ML, because of the advent of manifold learning, which we discuss in Section 2.5.

combinatorics: a field of mathematics that deals with counting combinations of objects

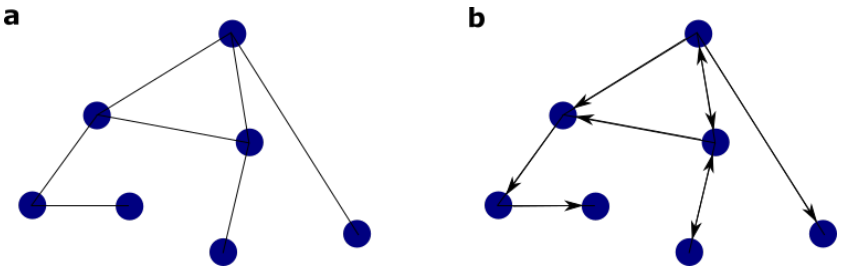

Figure 2.5: An example of an (**a**) directed and (**b**) undirected graph

Here, we lay the groundwork by introducing the mathematical object of a **graph**.

Definition 2.3.1 (graph) *A* graph *is a set of vertices V, which can be any object, and a set of pairs of vertices (v, w), where $v, w \in V$, which we call* edges. *We denote the set of edge E. Two vertices are called* connected *if there is an edge between them.*

Edges can be directed *(if the ordering (v, w) versus (w, v) should be separated) or* undirected *(in which case an edge will be denoted $\{v, w\}$).*

We can also put a weight function $f : E \to \mathbb{R}$ on the edges, in which case the graph is called a weighted graph.

Intuitively, edges denote when two vertices have some sort of relationship or connection.

A schematic of a graph is shown in Figure 2.5. Generally, graphs are extremely useful for depicting relationships between the vertex objects and are well known in biology as well — **trees** (see Figure 2.6 are used for representing the evolution of species, and graphs map out the interactions between different proteins in a regulatory network.

tree: a graph with no cycles, i.e. only one path along edges between any two points in the graph

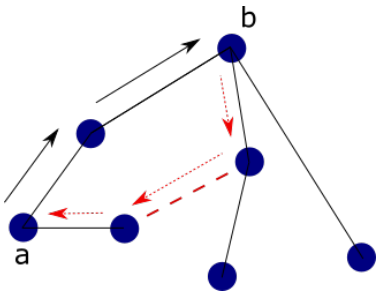

Figure 2.6: The black edges from a tree graph, with only one route from **a** to **b**. Adding the dashed edge creates a **cycle**, and there are now multiple paths.

Our work, which focuses on distances in high-dimensional data, focuses instead on a particular type of graph called a k-nearest neighbors (k-NN) graph, which is induced on a set of points that have some notion of similarity or distance between them, and where points are connected by an edge if they are one of each other's nearest neighbors. Formally:

Definition 2.3.2 (k-nearest neighbors graph) *Assume $(X_1, \ldots, X_N)$ is a set of N points, and let $d_{ij} \in \mathbb{R}^+$ represent the distance between X_i and X_j*[29]*. Then, given a positive integer k, the k-radius of X_i, denoted $r_i^{(k)}$, is the distance to the k-th closest point to X_i:*

$$r_i^{(k)} = \max_{\delta} \left\{ \delta \, : \, \left| \left\{ j \, : \, d_{ij} \leq \delta \right\} \right| \leq k \right\}, \tag{2.4}$$

29: not necessarily Euclidean, but should be a metric

and let $X_{k(i)}$ be the point j that is a distance $r_i^{(k)}$ from X_i. Then, the k-neighborhood of X_i, denoted $N_i^{(k)}$ is:

$$N_i^{(k)} = \left\{ X_j \ : \ d_{ij} \leq r_i^{(k)} \right\} \tag{2.5}$$

Then, the symmetric k-nearest neighbors graph *of X, denoted G_X, is the graph $G_X = (V, E)$, where:*

$$V = \{X_1, \ldots, X_N\}$$
$$E = \left\{ \{X_i, X_j\} \ : \ X_j \in N_i^{(k)} \right\}$$

In words, G_X is a graph where two points are connected if one is in the others neighbor set.

While this is what we are usually referring to when we talk about k-NN graphs, there are a couple of variations that are worth introducing.

The directed k-nearest neighbors graph *of X, denoted D_X, is the graph above but with undirected edges replaced by directed edges:*

$$V_D = \{X_1, \ldots, X_N\}$$
$$E_D = \left\{ (X_i, X_j) \ : \ X_j \in N_i^{(k)} \right\}$$

The mutual k-nearest neighbors graph *of X, denoted M_X, is the restriction of the symmetric k-NN to edges where both vertices are in each other's neighbor set:*

$$V_M = \{X_1, \ldots, X_N\}$$
$$E_M = \left\{ \{X_i, X_j\} \ : \ X_j \in N_i^{(k)} \text{ and } X_i \in N_j^{(k)} \right\}$$

In general, we will drop the k superscript when the value of k is clear.

2.4 Metric Learning

The traditional method for judging the distance between two vectors is the Euclidean distance: $d_E(x, y) = \left(\sum (x_i - y_i)^2\right)^{1/2}$. However, the notion of distance can broadly be generalized: mathematically, a distance metric is a function that satisfies the following conditions:

Definition 2.4.1 (metric) *Given a vector space V, a function $d : V \times V \to \mathbb{R}$ is a* metric *if the following hold:*

1. $d(x, y) \geq 0$ *for all* $x, y \in V$ *(non-negativity)*
2. $d(x, y) = 0$ *iff* $x = y$ *(uniqueness)*
3. $d(x, z) \leq d(x, y) + d(y, z)$ *for all* $x, y, z \in V$ *(triangle inequality)*

Metrics play a key role in many inference algorithms in machine learning: the prediction for a new datapoint is heavily dependent on the predictions of *similar* input points[30].

The field of **metric learning** aims to learn the *correct*[31] metric for a particular dataset or task. In the case where our data have labels, the quality of a distance metric can be based on subsequent performance in classification or regression tasks.

We can thus define the generalized supervised metric learning task:

Definition 2.4.2 (supervised metric learning) *Given a labeled dataset* (X, y)*, where* $X_i \in V$ *a vector space and* $y_i \in \mathcal{Y}$ *is the label for vector* X_i *(and* y_i *can be discrete or continuous), let* $\mathcal{K}$ *be the space of kernel functions, so an element*

$$K \in \mathcal{K} : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$$

defines similarity among input vectors; specifically, let

$$\mathcal{K}_X = \{K(\cdot, x) \ : \ x \in X\}$$

be the set of kernel functions induced by the similarity function given the input dataset X*. Then assume we are given a loss function* $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ *which evaluates a prediction for* X_i *against its label* y_i*; and of course a prediction function* $\mathcal{F} : V \times \mathcal{K}_X \to \mathcal{Y}$*, which, given the kernel function on the dataset* X*, predicts the value for a given input, the* supervised metric learning problem *aims to optimize:*

$$\arg\min_{K \in \mathcal{K}_X} \frac{1}{N} \sum_i \mathcal{L}(\mathcal{F}(X_i, K), y_i) \tag{2.6}$$

Thus, the task is to learn the similarity function between points such that the loss on the input dataset is minimized.

A common use case is *metric learning for nearest neighbors classification*. Here, a test point x is assigned the label of the majority[32] of its k-nearest neighbors for some hyperparameter k. So, the goal is to have an input k-NN graph such that an input point with a label $\ell \in \{0, 1\}$ is surrounded mostly by other training data that have label ℓ as well.

30: This is not *explicitly* true for many parametric methods, but remains *implicitly* true because of commonly used continuity constraints of the learned predictors. It is indeed explicitly true for the large class of *kernel methods* (we will discuss these below), which are built around the importance of similarity between inputs.

31: I have purposely left the definition of "correct" ambiguous, because the notion of a correct distance metric is, as we will see, absolutely essential to problems in computational biology.

32: This is easily extended to the multi-class case, in which the predicted label is the one shared by a *plurality* of neighbors.

Given a distance function d, let $d_k(x)$ be the distance to the k-th farthest point from x. In symbols:

$$d_k(x) = \min_z \left\{ z \in \mathbb{R}^+ \ : \ |\{y \in X \ : \ d(x,y) \leq z\}| = k \right\}$$

In this case, the kernel K between data points can defined, given the distance function d, as:

$$K_d(x,y) = \begin{cases} 1, & d(x,y) \leq d_k(x) \\ 0, & \text{otherwise} \end{cases} \tag{2.7}$$

and the predictor assigns to x:

$$\widehat{y}(x) = \mathscr{F}(x, (K_d)_X) = \mathbb{1}\left(\frac{1}{k} \sum_i y_i K_d(x, X_i) > \frac{1}{2} \right) \tag{2.8}$$

Thus, optimizing the loss function in (2.6) over the kernel means finding the distance function d that minimizes the classification error.

2.4.1 Linear methods

Crucial to solving the metric learning problem is choosing the class of distance functions that are allowed. This is generally the problem of choosing the kernel, which is recognized as a difficult problem [27]. The restriction to *linearity*, however, merits its own consideration since it both allows a straightforward exposition for the problem above and demonstrates the oft underestimated power of linear methods, a theme we will return to in our own work.

In the linear metric learning problem, popularly introduced as neighborhood component analysis (NCA) by Goldberger et al. [28], the goal is to find a matrix A such that transforming the data by A yields a better Euclidean distance metric. For vectors $x, y \in V$, the new distance metric d_A is thus given by:

$$d_A^2(x,y) = d^2(Ax, Ay) = \|Ax - Ay\|^2 = (x-y)^T A^T A (x-y)$$

Letting $Q = A^T A$, the goal of finding the optimal Q becomes learning a **positive semidefinite** matrix that optimizes (2.8).

positive semidefinite (psd): M is psd if $x^T M x \geq 0$ for all x; there are other equivalent definitions

It is worth noting here that in NCA[33] rather than the kernel matrix being strictly binary (X_j is either a neighbor of x or not), a **soft clustering** method is actually used:

$$K_d(x,y) \propto \exp\left(-d^2(x,y)\right)$$

33: and many other k-NN and clustering methods

soft clustering: instead of points being assigned to a single cluster, they are given probabilities of belonging to each possible cluster

where the proportionality constant is chosen so that $\sum_{y \in X} K_d(x, y) = 1$. In the specific case of NCA, we can write the full problem. Set:

$$p_j(x) = \frac{\exp\left(-(x - X_j)^T Q(x - X_j)\right)}{\sum_{y \in X} \exp\left(-(x - y)^T Q(x - y)\right)},$$

which lets us define the predictor (which here, will be a *membership strength* in $[0, 1]$ rather than an assignment):

$$\widehat{y}(x) = \frac{1}{N} \sum_i y_i p_i(x)$$

The goal of the algorithm is thus to maximize this quantity across the training data:

$$\mathscr{L} = -\frac{1}{N} \left(\sum_i y_i \widehat{y}(X_i) + \sum_i (1 - y_i)(1 - \widehat{y}(X_i)) \right) \tag{2.9}$$

(where the negative sign is to make it a minimization problem).

This framework is quite powerful and has been used as the underlying objective function for different optimization methods, with Goldberger et al. [28] following an explicit optimization approach to learn the transform matrix A, and Weinberger, Blitzer, and Saul [29] using the machinery of **semidefinite programming**, both showing excellent performance compared to other methods state-of-the-art at the time.

semidefinite programming: a method for solving constrained optimization problems where the constraints and the argument of optimization expressed as psd matrices

It also makes clear how one could move to nonlinear methods: by allowing A to come from more general function classes. The clear limitation of NCA is that the transformation A is limited to a matrix, and so is insufficient for data that cannot be linearly transformed to a Euclidean space.

2.5 Manifold Learning

Evaluating the quality of a metric in the previous section was possible because we were given labeled data. In other words, there existed a classification or regression task, and a good metric was one for which the predictor performed well on that task. Frequently, we are *not* given labels or a particular objective function to optimize and rather want to learn a metric that is somehow *intrinsically* representative of the data[34]. As one can guess, the notion of an intrinsically "correct" metric for an arbitrary dataset is tricky to define, much less learn,

34: The astute reader will note that we are moving to the realm of *unsupervised* learning as laid out in Subsection 2.3.1.

but the intuition is actually straightforward and best conveyed by an example.

Consider the "swiss roll" dataset shown in Figure 2.7. The generative model for the swiss roll is:

$$p_t = \begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} t\cos(\pi t) \\ t\sin(\pi t) \end{pmatrix} + e_t$$

where e_t is a small Gaussian error. Intuitively, this is a one-dimensional structure that can be parametrized by the distance along the roll. In fact, distance along the roll is intuitively a better metric that Euclidean distance, which, for example, would consider point A closer to C than to B.

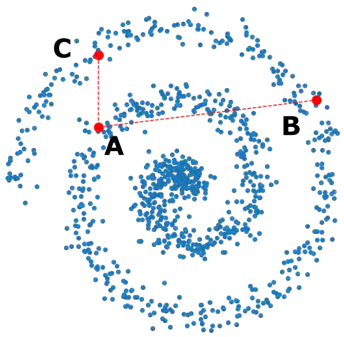

Figure 2.7: A swiss roll in two dimensions. Note that the point A appears closer to C than to B in Euclidean distance, but the opposite is true along the roll.

Learning intrinsically induced distance matrices often takes the form of **manifold learning**, where the data are assumed to lie on an underlying **manifold**. The theory of manifolds is a deep and rich subset of topology, but in our case, we can essentially equate manifold learning to the idea that only small scale Euclidean distances are accurate.

Definition 2.5.1 (manifold hypothesis) *Given a dataset $X \in \mathbb{R}^{N\times G}$ a set of vectors in ambient G-dimensional Euclidean space, assume there exists some (unknown) ground truth distance function d.*

X satisfies the manifold hypothesis *if there exists some small ϵ which, for all i, there is some radius r_i such that $\left|d(X_i, X_j) - \left\|X_i - X_j\right\|\right| < \epsilon$ for all j s.t. $\left\|X_i - X_j\right\| < r_i$.*

manifold: for our highly applied purposes, a manifold can be thought of as a subspace of Euclidean space imbued with its own metric, where, at small enough distance scales, the imbued metric is the same as the ambient Euclidean metric.

We note that this is an informal definition, as no conditions have been put on size of ϵ. One can formalize the definition by assuming X is *sampled* from an underlying manifold, allowing us to take $\epsilon \to 0$ with higher and higher sampling depth.

In general, the work we present is highly applied and will not *prove* that the manifold hypothesis holds for a particular dataset[35], but the machinery we develop implicitly assumes the hypothesis[36]. However, there is much evidence to suggest that biological data *does* satisfy the manifold hypothesis: among the most rigorous is the study of a similar concept, *entropic scaling* by Yu et al. [30], who find that biological data sets often have low-dimensional structure at local scales.

Computationally, a manifold learning problem entails answering the following two main questions:

35: One reason is that the existence of a "ground truth" distance in biological data is hard to show.

36: It should, however, be clear the connection to the formal idea of a manifold: the focus on small-scale distances comports with the idea that *these* distances are "easy" (i.e. Euclidean)

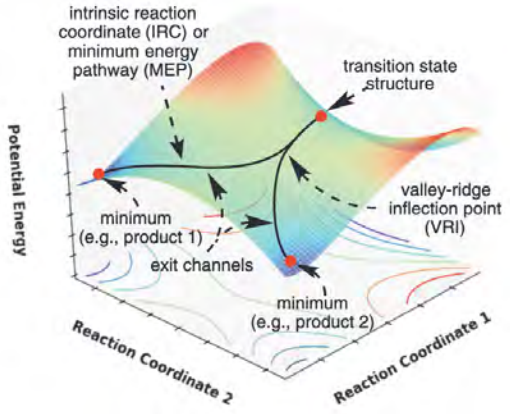

Figure 2.8: A representation of a thermodynamic system using reaction coordinates, taken from [31]

Goals of manifold learning

1. At what scale are the data Euclidean?[37]
2. How do you compute distances *outside* this scale?

37: And, as we will see in Chapter 4, how does this scale vary over the data?

2.5.1 Nearest neighbor graphs

In answering the first question, many methods use *neighborhoods* to determine scale. As introduced in Subsection 2.3.1, a dataset $X \in \mathbb{R}^{N \times G}$ can induce, for $k \in \mathbb{N}$, a k-NN graph, $G_X^{(k)} = (V_X^{(k)}, E_X^{(k)})$ (where we usually drop the k)[38].

Under many manifold learning methods, the parameter k is chosen as a hyperparameter and the radius $d_k(X_i)$ of the nearest neighborhood of point i is implicitly taken as the **local scale**, the scale at which the manifold can be considered Euclidean[39].

38: Unless otherwise specified, the nearest neighbors are calculated using Euclidean distance; we will see in Chapter 7 the benefits of generating k-NN graphs with better distance metrics

39: Note that this means the local scale is not constant over the dataset. Sparse regions have larger (Euclidean) local scales and vice versa. This idea underpins the results of Chapter 4

2.5.2 Density-based distances

To answer the second question, manifold methods frequently use distances determined by density in the dataset. This approach builds from earlier approaches in thermodynamics: a basic tenet of thermodynamics is that the probability of seeing a system in a particular state is proportional to the stability of that state[40]; and so, when looking at a thermodynamic system in terms of its reaction coordinates (see Figure 2.8), tracing the density of the states implies a reaction trajectory.

This motivates an approach to non-local distances which we can describe somewhat informally:

40: As a former physicist, I feel I must clarify — the probability of a state is proportional to the number of microstates that correspond to that state, which we proxy as stability.

Definition 2.5.2 (density distance) *Given a dataset $X \subset \mathbb{R}^G$, we define the* density distance *between $x, y \in X$ as follows. Set*

$$\mathcal{R}(x,y)\,\{(x, x_{r_1}, \ldots, x_{r_m}, y) \,:\, x_{r_i} \in X \text{ and } \|x_{r_{i+1}} - x_{r_i}\| \text{ small}\}\,, \tag{2.10}$$

so $\mathcal{R}$ is the set of possible paths between x and y where each step in the path is small.

Then, we can define the density distance:

$$d_R(x,y) = \inf_{R_r \in \mathcal{R}(x,y)} \sum_{i<|R_r|} \|x_{r_{i+1}} - x_{r_i}\|, \tag{2.11}$$

which intuitively is the distance of the shortest path between x and y

The intuitive idea behind a density distance is that the points in the dataset of interest X "outline" the manifold, and so if a region of $\mathbb{R}^G$ does *not* contain many points in X, then it is likely *not part of manifold*. By jumping from neighbor to neighbor, you are intuitively tracing out a **geodesic** between the points of interest. One possible way to formalize the notion of "small" in (2.10) is to ensure that $x_{r_{i+1}}$ is one of the nearest neighbors[41] of x_{r_i}.

geodesic: on a manifold, the shortest continuous path between two points

41: And in this case, the density distance between x and y as defined is merely the shortest path *in the graph* between the two points

On the swiss roll dataset, the notion of density distances works quite well. The dense regions of the roll trace out the spiral, and you would never jump "across" the open space, the way Euclidean distance would, to find nearest neighbors.

2.5.3 Diffusion map distances

The above formulation, has some notable weaknesses and rigidities, especially if your data are noisy around the underlying manifold. For example, consider the *noisy* swiss roll dataset in Figure 2.9, where there are occasional spikes in noise that put points into the "empty" part of the roll. Here, the density distance takes advantage of these noisy points to traverse the emptiness.

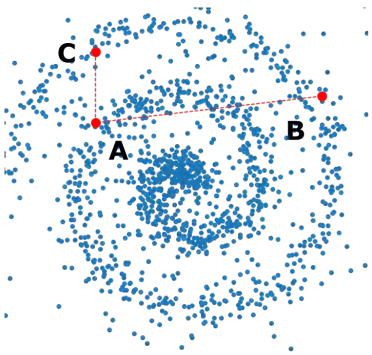

Figure 2.9: Noisy swiss roll dataset, where the empty space in the roll is sparsely populated. Now, one might be able to traverse from A to C across the chasm.

To address these robustness concerns, Coifman and Lafon [32] introduce *diffusion maps*, which, again, drawing from ideas in thermodynamics, model the distance between x and y on a manifold as the time it would take for heat to diffuse from x to y. Analogous to the soft-clustering approach described above, diffusion maps overlay the edges in a k-NN with an **exponential kernel**: the edge between points x and y is given the weight $w(x,z) = \exp\left(-\|x-z\|^2/\sigma^2\right)$, and this weight represents the probability that someone standing at

point x will jump to point z. The diffusion distance between x and y can then be thought of as the probability that someone starting at x will end up at y (after potentially jumping through several other points)[42].

42: Those familiar with the language of **Markov chains** will recognize the flavor of argument here — indeed, the weights on edges are the transitional probabilities between neighbors, and the final distances are determined by the stationary distribution of the Markov chain

2.6 Dimensionality Reduction

The machinery of metric learning attempts to learn distances between data points, but is generally agnostic to the representation of the point itself[43]. However, representations of the data are often quite important — for example, for visualization or for input into downstream tasks. Especially when data are assumed to be intrinsically low-dimensional, as discussed in Section 2.5, the appeal of a low-dimensional representation is well motivated.

43: In other words, metric learning methods don't try to learn new feature vectors.

Often, the representation is chosen so that a straightforward distance metric in the low-dimensional embedding (e.g. Euclidean distance) corresponds to the more complex notion of distance in the original data. This is actually exactly the case in NCA introduced in Section 2.4: if the transform A is chosen to be low rank (more columns than rows), then the NCA-transformed data will have rank(A) as its dimensionality, and the output distance metric is Euclidean in the transformed data. The swiss roll dataset again provides a compelling illustration. An effective one-dimensional representation of the dataset would "unroll" the dataset, mapping $p_t \mapsto t$.

Under this paradigm, dimensionality reduction (DR) is in some ways complementary to manifold learning: the goal of DR is to learn a transformation of the data such that the "correct" distance metric in the transformed data is Euclidean, rather than to learn the "correct" distance on the original data itself.

There are many algorithms for dimensionality reduction[44], making it quite challenging to discuss the field in generality. Nevertheless, we attempt to overlay some organizational hierarchy on existing algorithms to help our exposition. As shown in Figure 2.10, the first level split is between **linear methods** (Subsection 2.6.2), where the transformed dataset is generated by applying a matrix operator; and **nonlinear methods** (Subsection 2.6.3), which cover, well, everything else[45]. In the space of nonlinear methods, we further split them into **parametric** and **nonparametric** methods, although our focus will be on the nonparametric variety. Lastly, a more experimental split that we will explore later in Chapter 7 is the different between **local** and **global** algorithms — briefly, the crux of the split is whether

44: And practitioners tend to have strong preferences in what should be used, which we will discuss in Section 2.7.

45: "Using a term like 'nonlinear science' is like referring to the bulk of zoology as the study of non-elephant animals" — mathematician Stanislaw Ulam

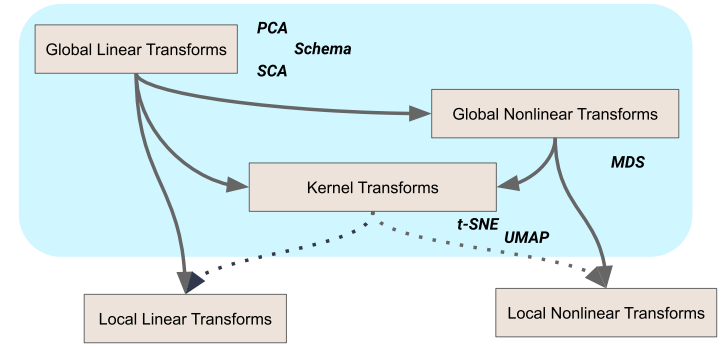

Figure 2.10: Organizing the space of dimensionality reduction algorithms

the algorithm uses the same objective function across the entire dataset.

2.6.1 Objective function

Before working through the hierarchy of DR algorithms, it is worth expanding on the *objective* of these algorithms. Unlike the context of supervised learning, for example, there is no canonical objective function that indicates the quality of an **embedding**. In fact, one of the crucial difficulties in comparing different algorithms is that they often implicitly optimize for different things.

embedding: the output of a DR algorithm; overloading this term, we refer to both the entire transformed dataset and each transformed feature vector as an embedding

Insofar as a "natural" objective exists, it might be the preservation of pairwise distances[46]. If we can create a low-dimensional representation of our input dataset where *every* pairwise distance is preserved, then in some sense, we have preserved all the information of the original dataset in the embedding. Even if Euclidean distance is not the "correct" distance, many kernel based methods that transform the distance metric use Euclidean distance as their input, and so the transformation would not affect their use.

46: When used without qualification, we mean Euclidean distance

We will see that several methods aim to optimize this objective either implicitly (like PCA) or explicitly (like MDS)[47]. However, the dream of perfectly representing your high-dimensional data in low-dimensions by preserving all pairwise distances turns out to be impossible. A classical result from Johnson and Lindenstrauss [33], now known as the Johnson-Lindenstrauss (JL) lemma, gives a tight lower bound on the number of dimensions you can reduce to while accurately preserving pairwise distances:

47: These will be introduced below

Theorem 2.6.1 (Johnson-Lindenstrauss) *Let $\epsilon \in (0, 1/2)$, and let $Q \subset \mathbb{R}^G$ be a set of N points, and set $k = (20 \log N)/\epsilon^2$. There exists a mapping $f : \mathbb{R}^G \to \mathbb{R}^k$ such that for all $x, y \in Q$:*

$$(1 - \epsilon)\|x - y\|^2 \leq \|f(x) - f(y)\|^2 \leq (1 + \epsilon)\|x - y\|^2$$

so f maps $\mathbb{R}^G \to \mathbb{R}^k$ which mostly preserving pairwise distances.

But, crucially, the JL lemma is tight [34], i.e. for any $r < k$, there is *no* embedding function that can preserve all pairwise distances to within a factor of ϵ. Since the optimal embedding function f is essentially achieved with random projections, the upshot is that no DR algorithm can really beat random projections when it comes to preserving pairwise distances.

To get past these fundamental limitations, we again return to the crucial theme underlying this work: that understanding the *correct* distance metric, which is not always[48] Euclidean distance, is the key to analyzing high-dimensional data. Some DR algorithms remain useful past the JL boundary because they remove the influence of features that are just noise and therefore not meaningful[49]. And others, channeling the manifold learning ideas we introduced earlier, only aim to preserve the local-scale distances which are meaningfully Euclidean. As we review the methods below, the reader is advised to keep in mind exactly what notion of distance the methods are focused on.

48: In fact, I would say rarely

49: And so their contribution to pairwise distances should be ignored

2.6.2 Linear methods

As in most algorithmic developments, the earliest methods for DR were linear, with the singular value decomposition (SVD) and the related principal components analysis (PCA) the most popular, and nearly ubiquitous not just in machine learning, but in many other fields of data analysis.

The goal of both methods is to find a better set of axes to represent a dataset — or, equivalently, to *rotate* the data so that each axis represents something meaningful about the data. Specifically, each axis should represent as much *variation* in the dataset as possible. Mathematically, PCA is the eigendecomposition of the covariance matrix:

Definition 2.6.1 (PCA) *Given a data matrix X where the mean of each*

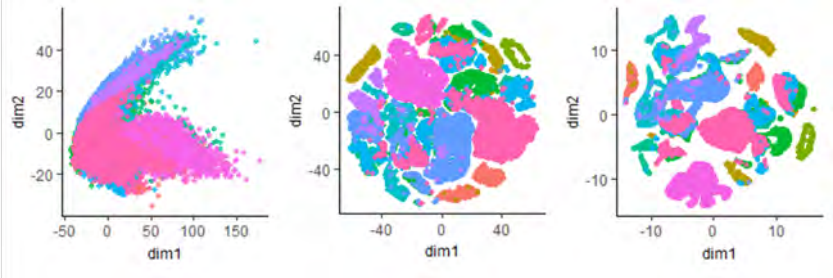

Figure 2.11: Weaknesses of PCA visualization: For an high dimensional dimensional dataset (taken from `http://carbonandsilicon.net/rblogging/2018/02/27/UMAP_plots`)

column is 0, we can write its covariance matrix as X^tX. *Let*

$$X^tX = W\Lambda W^t$$

be the eigendecomposition of the covariance matrix. Then the vectors w_i *are the change-of-basis from standard axes to the PCA axes, and the* PCA-transformed *data is given by:*

$$T(X) = XW \tag{2.12}$$

The SVD is a closely related transformation that works with the data matrix directly rather than the covariance matrix, but its output has the same interpretation.

As a method for DR, since the principal components are ordered by how much variation they describe, one should expect that the top principal components represent most of the important information about the dataset. Indeed, to reduce a G-dimensional dataset to r dimensions, we rewrite (2.12) as:

$$T_k(X) = XW_{(k)} \tag{2.13}$$

where $W_{(k)}$ is the first k columns of W.

It should be noted that despite the "classical" nature of many of these methods and the seemingly onerous restrictions on power imposed by linearity, research continues into developing them. See for example the recent development of surprisal components analysis (SCA) [35].

2.6.3 Nonlinear methods

Especially in the realm of visualization[50], linear methods have often not been successful at effectively communicating structure[51], as evidenced by Figure 2.11.

Our main focus in this work will be **nonparametric** nonlinear meth-

50: By visualization, we generally mean dimensionality reduction down to two or three dimensions, so the data points can be displayed on a scatter plot

51: We will exhaustively discuss what we mean by "structure" below and in future chapters

nonparametric: there are many possible definitions, but we mean here that we do not have an explicit definition for the function that transforms the high-dimensional data, which means we do not know or specify its functional form

ods for DR, because they have become standard in visualizing single-cell datasets and have started to garner interest as pre-processing methods for downstream analyses. These methods are also often built, explicitly or implicitly, around manifold learning principles[52].

52: That is, they assume that the underlying data lie on a manifold.

However, we first consider the setting the JL lemma addresses, and consider the nonlinear methods that attempt to preserve pairwise distances, namely *multidimensional scaling* (MDS). The context here is that the *full* distance matrix of the original data is known, and our goal is to find a new representation. Given a dataset $(X_1, \ldots, X_N)$, the generalized objective function for the various MDS algorithms is:

$$\mathcal{O} = f\left(\left[\mathcal{L}(d_{ij}, \left\|Y_i - Y_j\right\|)\right]_{i,j\leq N}\right) \tag{2.14}$$

where Y_i is the embedding of input data point X_i and d_{ij} is the distance between X_i and X_j in the original data. $\mathcal{L}$ is a loss function that penalizes the embedding distances for being too far from the original distances, and f is any agglomeration function that combines the losses from each pairwise distance into a total loss.

The goal of MDS is then to choose the Y_i such that $\mathcal{O}$ is minimized. For example, in *classical* MDS, the specifics are:

$$\begin{aligned} d_{ij} &= \left\|X_i - X_j\right\| && \text{(Euclidean distance)} \\ \mathcal{L}(d_X, d_Y) &= (d_X - d_Y)^2 && \text{(squared loss)} \\ f((\mathcal{L}_{ij})_{i,j\leq N}) &= \left(\sum_{i\neq j\leq N} \mathcal{L}_{ij}\right)^{1/2} && \text{(root-mean-square)} \end{aligned}$$

But the framework is flexible. In non-metric MDS (NMDS), for example, the same framework as above is used, but the d_{ij} can be an entirely arbitrary metric[53].

53: The advantage of Euclidean MDS is that it can be optimized efficiently, using properties of the Euclidean inner product.

We note one particular modification of the agglomeration function f and loss function $\mathcal{L}$ that we will build on in Chapter 3, Pearson correlation between the distances:

$$\begin{aligned} \mathcal{L}(d_X, d_Y) &= (d_X - \mu_{d_X})(d_Y - \mu_{d_Y}) && \text{(covariance)} \\ f((\mathcal{L}_{ij})_{i,j\leq N}) &= \frac{1}{\sigma_{d_X}\sigma_{d_Y}} \sum_{i\neq j\leq N} \mathcal{L}_{ij} && \text{(squared correlation)} \end{aligned}$$

where μ_{d_X} and σ_{d_X} are the mean and standard deviation of the set of pairwise distances (and analogously for d_Y). The rationale behind correlation, rather than exact equality among distances is the assumption that only the *ordering* of the distances matter, rather

than their actual value. This softens the necessity that the original distance metric is correct — all that is necessary now is that the ordering of distances in the original space is correct.

Diffusion-based methods

MDS traditionally performs poorly on visualization tasks and does not scale well to large datasets. Unsurprisingly, these methods struggle with manifold data, since large-scale distances are not easy to learn on manifolds. The diffusion-based methods we discussed in Subsection 2.5.3 use the diffusion-map distances, rather than Euclidean distances, as their input for dimensionality reduction. As we noted, the distance between points X_i and X_j is the stationary probability that a random walk with restart (RWR) starting at X_i will end at X_j. This allows for a density-based traversal of the dataset, but also defines two points as close by if there are many ways to get from X_i to X_j.

As a tool for DR (rather than just for metric learning), diffusion maps apply an eigendecomposition to the *random walk matrix* and use *that* as the reduced representation [32, 36]. Recalling that PCA is the eigendecomposition of the covariance matrix of the data, we can interpret the output of a diffusion map as a version of PCA that considers the random-walk distance as the covariance matrix of the data. In fact, we can consider the RWR matrix as a kernel, so diffusion maps become a form of kernelized PCA[54].

54: We note here that Ham et al. [37] show that nearly *all* diffusion-based DR methods are essentially a kernelized version of PCA

Manifold projections

We have built up now to the nonlinear, nonparametric DR algorithms that have become the *de facto* standard in dimensionality reduction: *t*-distributed stochastic neighborhood embedding (t-SNE) [38] and the very recent uniform manifold approximation and projection (UMAP) [39]. These algorithms both really focus on the manifold assumption, eschewing the global intuition in PCA or the (kernel-PCA) diffusion methods and dialing down on aligning only local distances.

We leave a thorough discussion of these algorithms to Chapter 4, where we present our own improvements to the methods. However, it is instructive to outline them here, especially as their objective functions contrast with previous methods. Notably, they combine a crucial aspect of diffusion maps — their focus on locality — while not restricting the transformation to a linear one. In fact, the embedding of a point is *not* actually the output of a parametric function but

rather is dynamically chosen to minimize an objective function, as in MDS.

The fundamental unit of analysis for both algorithms is the k-NN graph we defined in Subsection 2.3.3. Rather than consider *all* pairwise distances, only the edges *in* the graph are considered significant[55]. A weight function (that rapidly decays with distance) is placed on the edges between the points, further emphasizing the focus on local distances.

55: In fact, both algorithms actually institute a *repulsive* force between all pairs of points that do not have an edge

After the points are initialized in low-dimensional space[56] an analogous low-dimensional weight function on distances in low-dimensions is compared to the weight function in high-dimensions. The objective of the algorithm is thus to minimize the distortion between the weight functions — the objective tends to be minimized when the neighbors of a point in low-dimensions are the same as the neighbors in high-dimensions.

56: There are several methods for doing this initialization, which we won't worry about here

Crucially, both methods are implicitly kernel methods (recall Subsection 2.3.2), so their input depends *only* on the definition of distance in the original space; this scenario is basis for our method for generating locally informed k-NN graphs in Chapter 7. Similarly, the embedding generated by these algorithms is *also* only dependent on distances in the low-dimensional space. That is, the coordinate-values themselves have no real meaning, only the distances induced *by* those coordinates[57].

57: As we will discuss further in Chapter 4, this is part of why using UMAP or t-SNE embeddings as feature vectors for downstream analysis is fraught

2.7 Evaluating Algorithms

Developing the algorithms we have so exhaustively discussed here is only one part of the task of answering the biological questions that initially motivated this chapter. It is just as important to figure out how we can know whether the algorithm has actually solved the desired problem. In this section, we will discuss briefly the philosophy of evaluation, which is itself a multidimensional problem. Some angles that must be considered are discussed below:

Dimensions of evaluation

We note that the following are not necessarily entirely independent of each other; nor are they exhaustive of all the potential ways to think about evaluation.

- **Appropriateness of translation:** Do the mathematical objects accurately represent the important aspects of their

biological counterparts?

- **Appropriateness of objective:** Is the objective function that the algorithm aims to optimize actually the correct objective to answer the biological question? This is the key question for a lot of our work — what kind of distortion is acceptable when transforming biological data?
- **Appropriateness of data model:** Statistical models make assumptions about the processes that generate their data[58]. It is important to ask whether the assumed data model is actually a reasonable representation of the data. Recent work in scRNA-seq, for example, considers whether a type of model called a *zero-inflated negative binomial* model[59] appropriately represents expression data.
- **Interpreting results:** While the above examples are generally asking the question, "Is the model good?" the *purpose* of creating these models is actually to ask, "What do the results tell me about biology?" We spend a lot of time in all the work presented in this dissertation asking exactly that question: how can we translate the numerical results our transformations yield to new biological insights and questions.

58: Most notable is the i.i.d. assumption — that each data point is generated from the same distribution, independently

59: briefly, a model that allows for more zeroes than expected

2.7.1 Classification of single-cell data

Supervised learning tasks rely on labeled data, and the labels need to be accurate[60]. The particular case of *classification* tasks, where the goal is to assign a discrete label to a data point, demands special attention.

60: While robust methods exist for dealing with mislabeled data, these considerations are not centered in most existing methods

In the realm of scRNA-seq, the discrete labels we are concerned with are often cell-type assignments. Cell types can determined in different ways — sometimes, known **marker genes** can be evaluated while doing the sequencing itself. However, using marker-gene-based cell-type assignment requires foreknowledge of the cell-types that one expects to find in the dataset, which, especially when working with new or under-studied datasets[61], is not always possible.

marker gene: a gene that is only expressed — and always expressed — in one type of cell, and so presence of that gene's transcripts in a cell allows its cell-type to be determined

61: which is, after all, one of the main appeals of single-cell analyses.

In those scenarios where marker genes are not known, researchers often have to turn to computational methods to determine cell-type labels. Primarily, the main computational methods are built around the clustering methods we discussed in Subsection 2.3.1: the scRNA-seq data are processed[62] and then a clustering algorithm is run. Each resulting cluster is then representative of a discrete cell-type.

62: i.e. normalized, and potentially fed through a dimensionality-reduction algorithm for noise reduction.

While cell-type discovery is itself a task in its own right, the reason it is important to discuss evaluation here is that the assignment of cell-types is often used as a baseline step for evaluating or developing *downstream* algorithms — if cell-type assignment is poorly done, the errors can propagate through those downstream algorithms.

Cell-type dependent methods

There are several methods that depend on cell-type assignment for either evaluation or as a baseline for their algorithms

- **Differential gene expression:** As discussed in Subsection 2.2.2, one of the major benefits of single-cell data is the ability to understand the differences in different types of cells, which necessitates deciding which cells are of different types. Methods to find marker genes [40, 41] are evaluated based on their abilities to "rediscover" known marker genes for known cell-types; methods to find local **coexpression** patterns specific to cells [42] build their networks based on cell-type assignments.
- **Rare cell-type detection:** As single-cell datasets become larger and larger, the resolution with which cell-types can be found increases. Rare cell-type detection methods [43–47] are necessarily evaluated on their ability to detect sub-clusters within existing clustering methods. Even methods that do not explicitly search for rare cell-types but are built around understanding the distribution or density within expression space of data [35, 48] are evaluated based on how closely cells of the same cell-type are connected.
- **Integration:** As discussed in Subsection 2.2.2, combining multiple datasets has become an important problem in scRNA-seq. The crucial challenge in integration is to remove *batch effects* that are specific to the dataset from "actual" biological variation, which should be preserved. The way that integration methods [22, 23, 49] are able to separate these effects is by seeing whether the algorithms correctly interweave cells that are of the same ground truth cell-type but in different datasets.

It is therefore important to ensure that the discrete cell-types that underlie these algorithms are reasonable!

coexpression: the correlation between the expression-levels of genes in a given [sub-]population

For methods reliant on discrete cell-type classifications, an underlying assumption is often that *all* the cells within that cell-type come from the same generative model. Under this paradigm, the cells at

the peripheries of their assigned cluster are just very noisy versions of that cluster. Whether using clusters or marker-genes for identification, decisions about thresholds have to be made about when a cell belongs to a cell-type. Even works that evaluate methods for assigning cell-types [50] rely on *known* marker genes as ground truth. Moreover, many works [51, 52] consider the cell-type assignments when evaluating even *unsupervised* algorithms — the underlying idea behind these evaluations is that within a cell-type, the cells should be considered "close" enough that Euclidean distance is reasonable[63].

We engage in this discussion not to oppose the notion that cell-types exist, but rather to alert the reader that all evaluations[64] of cell-type-based methods need to consider the possibility that their cell-type assignments are not accurate. In Chapters 4 and 7, for example, our work disambiguates existing cluster labels to find substructures within a given label.

63: It should be noted that the papers cited do not attempt to *prove* — nor is it obvious how or whether they *could* prove — that Euclidean distance within a cell-type is appropriate or accurate.

64: including our own

2.7.2 Metrics for unsupervised algorithms

Our work in developing better distance metrics for single-cell data is, based on the discussion in the previous section, going to be mostly *unsupervised* — we do not generally rely on cell-type labels when coming up with metrics for single-cell data[65].

Under the aegis of manifold learning (see Section 2.5), our goal in the work we present here is to come up with notions of distance that *do not* depend on cell-type labels. This is not an uncontroversial mode for inquiry. In evaluating various methods for unsupervised methods, practitioners in the field have struggled to develop "objective" criteria for evaluation. The question to answer — that remains open, in our view — is exactly what objective function an unsupervised method in single-cell analysis should optimize. Here we discuss briefly some of the ways that the field has attempted to evaluate unsupervised methods[66].

Among the most pervasive of critiques relies on the preservation of *Euclidean* pairwise distances in the original high-dimensional space. At face value, this is absolutely a reasonable critique of methods — the Euclidean distance between high-dimensional vectors is as close as one can get to a "natural" method for comparison, and a method that scrambles Euclidean orderings of distance should naturally be treated with suspicion. One of the most recent high-profile papers that takes this approach is by Chari, Banerjee, and Pachter [51], who find the state-of-the-art dimensionality reduction methods UMAP

65: We do however evaluate our methods based on the so-called ground truth of cell-type labels. Our philosophy here is that if our unsupervised methods can recreate cell-type labels, then we can trust the other types of structure that our methods show.

66: We will delve more deeply into the critiques in the chapters where they are most appropriate.

and t-SNE (discussed in Section 2.6) *severely* lacking in this pairwise density preservation. In fact, the original paper motivating UMAP for use in visualizing scRNA-seq data [52] touted its performance on this metric as a reason for using it.

As Section 2.5 makes clear, the importance of pairwise distances outside of some small local-scale is doubtful if one subscribes to the manifold hypothesis. But, continuing the string of caveats that, more than anything define this work, it *does* remain an open question exactly *how* to determine this local scale. For example, the aforementioned critique [51] considers, in one of their analyses, the *intra*-cluster distance preservation in a UMAP embedding of an scRNA-seq dataset — the implicit assumption here is that the scale at which scRNA-seq data is Euclidean is within each cluster. It is not clear that this true, especially when substructure inside a cluster exists.

We do not attempt to — or claim to — prescribe a correct method to evaluate unsupervised learning methods. Rather, our contention is that there is *no* universal method for evaluating an unsupervised learning method. Our aim, as we discuss in the following section, is instead to use what is known experimentally to buttress our unsupervised learning-based claims.

2.7.3 Evaluation in this work

In the works presented here, we generally attempt to combine intuition that is motivated by theory and performance on real data. Unlike other methods, we generally do not have existing objective functions and ground truth labels to compare our performance to. Thus, developing the *objective function* itself will be a major part of our discussion, and motivating the choice of objective will figure prominently in our evaluation.

From the theory-side, we do not generally have airtight proofs underlying our methods, but rather develop intuitions for the generative model of the data, and show how our method is well-suited to that intuition.

Theory examples

While we of course leave the details to the proper sections, we signpost some examples of theoretical evaluation to demonstrate how we combine intuition and rigor:

- **Manifold reshaping:** When demonstrating why we are interested in using correlation between distances as a way to faithfully preserve global distances in Chapter 3, we use a toy example where the manifold distance and global Euclidean distance do not agree, and prove that our approach "fixes" Euclidean distance.
- **Density across dimensions:** We motivate the use of a logarithmic transform in the dimensionality reduction method we develop by considering the extremely stylized case of a **ball** in high-dimensions being "squished" to low-dimensions without affecting its density, and show that a logarithmic transform is necessary for maintaining density.

ball: given a radius r, the set of points $\{x : \|x\| \leq r\}$

Because the theory developed is focused on highly-simplified generative models, theory alone is not enough to "prove" that the methods we have developed are useful. To that end, we combine empirical evaluation, trying to be cognizant of the difficulties inherent in empirical valuation of unsupervised learning methods discussed above.

Empirical evaluation

We briefly highlight some empirical examinations that buttress our claims.

- **Differential gene expression**: Our methods often "create" new clusters that do not correspond to a "known" cell-type. In order to "validate" the new clusters we are seeing, we find the differentially expressed genes in the new cluster and search the literature for evidence that such a subtype exists.
- **Fidelity:** While we are cautious about the idea that cells of a given cell-type label should *always* be embedded together, we do concede that, usually, it is the case that cells of a given cell-type are more similar to each other than those with other labels. Thus, when evaluating the local decompositions in Chapter 7, we show that our work actually makes sure that cells of the same label are embedded closer together *without using the labels themselves* to develop the method.

3 Metric Alignment of Multimodal Data*

In our attempts to understand distance metrics for single-cell data, our first contribution concerns the multi-modal scenario considered in Subsection 2.2.1, where we have multiple modalities of data about the same single-cell dataset.

3.1 The Promise of Multimodal Data

High-throughput assays can now measure diverse cellular properties, including transcriptomic, genomic , epigenomic , proteomic, functional, and spatial data modalities.

Types of multimodal data

We summarize the promise of some of the multimodal data types from above:

- **Transcriptomic:** This is type of data we are most familiar with, scRNA-seq data which counts the amount of the mRNA transcripts in each cell in the sample [19, 55, 56].
- **Genomic:** Analogously, one can profile the *genome* itself in each cell — this allows researchers to find mutations in the genome, especially useful for studying tumors or datasets representing multiple individual organisms [15, 57].
- **Epigenomic:** Methods have been developed for profiling the environmental and physical factors that modulate gene expression. For example, chromatin accessibility profiling [58]

* The work in this section is drawn from the preprint 'SCHEMA: A general framework for integrating heterogeneous single-cell modalities' by Singh et al. [53], which is focused on developing the method itself, and the following *Genome Biology* publication [54], which focuses on results when the methods are applied.

measures which subsections of chromatin can be accessed by transcriptional machinery and methylation profiling finds which histone proteins are methylated, a well understood epigenetic mechanism [59]; and ChIP-seq [60] focuses on regions of the genome where transcription factors can bind.

- **Proteomic:** Analogous to genomic and transcriptomic sequencing, single-cell proteomic sequencing [61] profiles polypeptide counts at single-cell resolution.
- **Functional:** Understanding a cell's products, as the above methods are able to do, is enriched by the ability to understand its interaction with stimuli. Functional sequencing [15] profiles the metabolites — the products and inputs to cellular reactions — also at single-cell resolution.
- **Spatial:** The importance of physical organization of tissues has always been a focus of anatomy and physiology, but the development of the Slide-seq sequencing platform [62] allows researchers to understand physical organization at the cellular level.

Excitingly, single-cell experiments increasingly profile multiple modalities simultaneously within the same experiment [15, 58, 61, 62], enabling researchers to investigate covariation *across* modalities; for instance, researchers can study epigenetic gene regulation by correlating gene expression and chromatin accessibility across the same population of cells. Importantly, since the underlying experiments provide us with multimodal readouts per cell, we do not need to integrate modalities across different populations of cells [1].

1: The problem of integrating datasets that consist of *different* cells is a complementary problem that is also well-studied [22, 23, 63–67]

Simultaneous multimodal experiments present a new analytic challenge of synthesizing agreement and disagreement across modalities. For example, how should one interpret the data if two cells look similar transcriptionally but are different epigenetically[2]? Moreover, given the rapid biotechnological progress that continues to enable novel measurement modalities and easier simultaneous multimodal profiling, a multimodal analysis paradigm should scale to massive single-cell datasets, be robust to noise and sparsity in the data, and be able to synthesize two or more arbitrary modalities in an interpretable way. Many existing methods, however, struggle with scalability, **overfitting**, or are specialized to specific multimodal tasks (such as just spatial transcriptomic [68–70] or only gene-set estimation [71, 72]).

2: In our language of distance metrics, what if we have two different metrics on the same dataset, where they disagree over a pair of points?

overfitting: a model is *overfit* if it performs well in training but not in "the wild" — it has learned characteristics extremely particular to the testing set

We therefore present Schema, a method that synthesizes multimodal data based on a conceptual framework that accommodates any

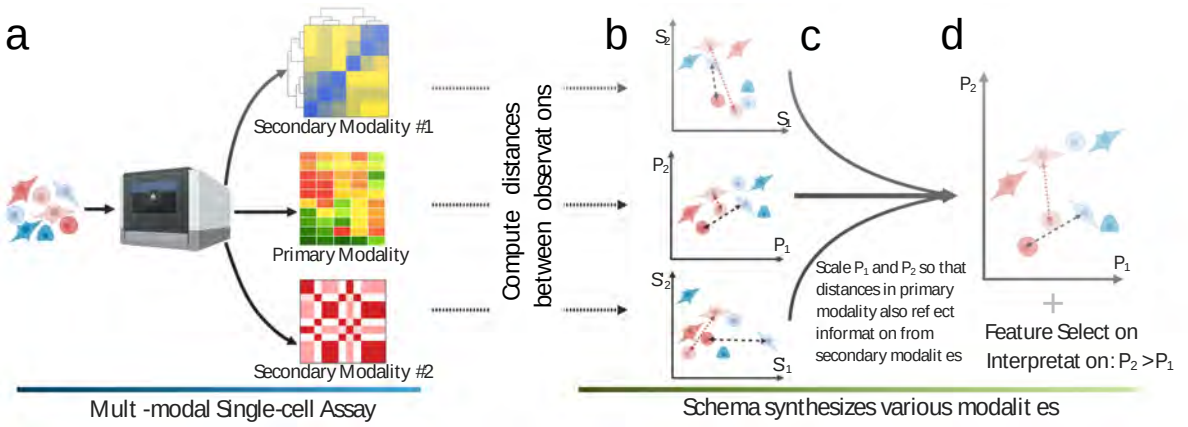

Figure 3.1: Integration of simultaneously assayed modalities using Schema. a. Schema is designed for assays where multiple modalities are simultaneously measured for each cell. The researcher designates one high-confidence modality as the primary (i.e., reference) and one or more of the remaining modalities as secondary. **b.** Each modality's observations are mapped to points in a multi-dimensional space, with an associated distance metric that encapsulates modality-specific similarity between observations. Across the three graphs, the dashed and dotted lines indicate distances between the same pairs of observations. **c.** Schema transforms the primary-modality space by scaling each dimension so that the distances in the transformed space have a higher (or lower, as desired) correlation with corresponding distances in the secondary modalities; arbitrary distance metrics are allowed for the latter. Importantly, the transformation is provably guaranteed to limit the distortion of the original space, thus ensuring that information in the primary modality is preserved. **d.** The new point locations represent information synthesized from multiple modalities into a coherent structure. To compute the transformation, Schema weights features in the primary modality by their importance to its objective; we have found this feature-selection aspect very useful in biological interpretation of its results.

number of arbitrary modalities. Schema draws from metric learning[3] [73–76]. Our critical insight is to interpret each modality as describing its own measure of distance between the underlying cells; we can then newly formulate the synthesis problem as reconciling the information implied by these different distance measures.

3: Recall from Section 2.4 that this is the subfield of machine learning concerned with computing an accurate measure of similarity (equivalently, distance) on a dataset

Schema achieves this multimodal synthesis through an interpretable and principled **quadratic programming** formulation to compute the optimal reweighting of a modality's features that maximizes its agreement with other modalities. Thus, a key advantage of our approach is that it provides feature weights that enable a researcher to understand where different modalities agree and where they do not. Our constrained optimization approach also improves Schema's robustness to outliers and to overfitting.

quadratic programming: a framework for optimization that allows a quadratic term in the objective function; see Subsection 3.4.3

3.2 Multimodal Analysis as Metric Learning

Before the advent of multimodal single-cell experiments, computational analysis has focused on variation within a single modality. In contrast, analysis of simultaneous[4] multimodal single-cell experiments[5] critically requires reasoning about information across modalities in a mutually consistent way. Our key intuition is that

4: We emphasize again that the crucial simultaneity here is that the *same* set of cells are used *across* the modalities

5: where two or more modalities are available per cell

each modality gives us information about the biological similarity among cells in the dataset, which we can mathematically interpret as a modality-specific distance metric. For example, in RNA-seq data, cells are considered biologically similar if their gene expression profiles are shared; this may be proxied as the Euclidean distance between normalized expression vectors, with shorter distances corresponding to greater similarity.

To synthesize these distance metrics, we draw inspiration from metric learning. Given a reference modality, Schema transforms this modality such that the Euclidean distances in the transformed space agree with a set of supplementary distance metrics from the other modalities, while also limiting the distortion of the original reference modality. Analyses on the transformed data will thus incorporate information from all modalities (Figure 3.1). For instance, with RNA-seq data as the reference modality, Schema can transform the data so that it incorporates information from other modalities but limits the distortion from the original data so that the output remains amenable[6] to standard RNA-seq analyses (e.g., cell-type inference, trajectory analysis, and visualization).

6: and by default, remains the same dimensionality

In our approach, the researcher starts by designating one of the modalities as the primary (i.e., reference) modality, consisting of observations that are mapped to points in a multi-dimensional space. In the analyses presented here, we typically designate the most informative or high-confidence modality as the primary or the reference modality, with RNA-seq being a frequent choice[7] . The coordinates of points in the primary modality are then transformed using information from secondary modalities. Importantly, the transformation's complexity is constrained by limiting the distortion of the primary modality below a researcher-specified threshold. This acts as a regularization, preventing Schema from overfitting to other modalities and ensuring that the high-confidence information contained in the primary modality is preserved. We found this constraint to be crucial to successful multimodal syntheses. Without it, an unconstrained alignment of modalities using, for instance, canonical correlation analysis (CCA), a common approach in statistics for inferring information from cross-covariance matrices, or autoencoders, a deep learning approach for mapping multiple datasets to a shared latent space [77–80], is prone to overfitting to sample-specific noise, as we show in some of the case studies in Chapter 5.

7: The notion of "highest-confidence" modality is certainly somewhat arbitrary, but generally the most high-dimensional and least sparse modality is a good rule of thumb; one could also potentially derive measures of information content, which we have not yet attempted

To see how Schema's transformation synthesizes modalities, consider the case where the primary dataset is gene expression data. While the points close to each other in Euclidean space are likely to be

biologically similar cells with shared expression profiles, longer Euclidean distances are less informative[8].

Schema's constrained optimization framework is designed to preserve the information contained in short-range distances, while allowing secondary modalities to enhance the informativity of longer distances by incorporating, for example, cell-type metadata, differences in spatial density, or developmental relationships. To facilitate the representation of complex relationships between modalities, arbitrary distance metrics and kernels are supported for secondary modalities.

Schema's measure of inter-modality alignment is based on the **Pearson correlation** of distances, which is optimized via a quadratic programming algorithm, for which further details are provided in Subsection 3.4.3. An important advantage of Schema's algorithm is that it returns coefficients that weight features in the primary dataset based on their agreement with the secondary modalities (for example, weighting genes in a primary RNA-seq dataset that best agree with secondary developmental age information)[9]. These feature weights enable greater interpretability into data transformations — this is not immediately achievable by more complex, nonlinear transformation approaches [77–83]. We demonstrate this interpretability throughout our applications of Schema.

8: This is of course, the underlying intuition behind the *manifold hypothesis* discussed in Section 2.5. Schema does not *explicitly* take advantage of the manifold hypothesis but we show that intuitively, short-term distances are the ones that are conserved.

Pearson correlation: for random variables X, Y, the correlation is $\rho = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y}$

9: The interpretation as feature weights is due to an extremely stringent limitation on the class of functions that transformation can take — only scaling transforms; this provides great computational and interpretation advantages

3.3 Manifold Intuition

We now discuss first at a high level the technical details motivating the Schema algorithm before an in-the-weeds analysis in Section 3.4. To begin with, suppose we have N observations; in a single-cell setting these would correspond to cells. Next, for each observation we have multiple types (i.e., modalities) of data, $D_1, D_2, \ldots, D_r$[10]. If these datasets all represent views of the same underlying biology, they should be in some kind of agreement. But noise, experimental artifacts and, importantly, unknown biological factors make this hard to discern. Our approach to the heterogeneous integration task is to produce a new dataset D^* that combines the information from D_1, D_2, D_3 and is in some agreement with each of them.

To crystallize this intuition of "agreement", we build upon an idea common to many machine learning techniques and single-cell analyses [39, 84, 85]: analyze the data exclusively in terms of distances between points in the data. Biologically, this is well justified. For example, cells with similar expression profiles typically belong to the same cell group/cluster[11].

10: For example, dataset D_1 may record transcriptional information (represented as a $N \times k$ matrix, k being the number of genes), dataset D_2 may have the 2-dimensional location of the cell in a tissue slice (an $N \times 2$ matrix), and dataset D_3 might be cell-line information (a $N \times 1$ vector of labels)

11: And of course, the centrality of distances is a key theme of this dissertation

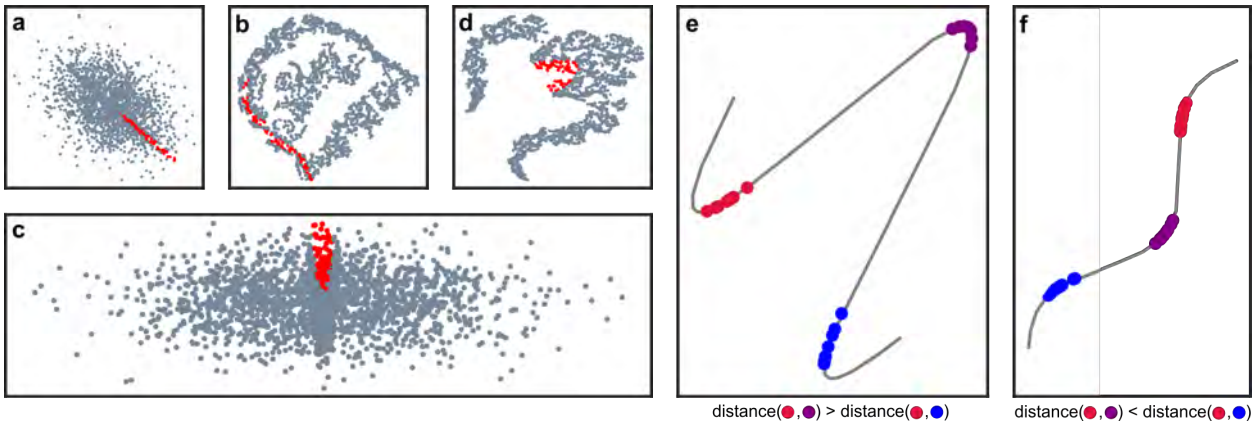

Figure 3.2: Demonstration of Schema on a toy dataset. The original dataset (**a**) has a small set of points that are differently colored (red). An analogy here would be to rare cell-types in a single-cell dataset. There is enough noise that a t-SNE plot of the raw data (**b**) fails to clearly separate the red points. Running Schema produces (**c**), with the data now rotated and scaled to accentuate the red points. Note how the t-SNE plot for this transformed data (**d**) clearly separates the red points from the rest. **(e,f) Manifold-reshaping interpretation of Schema**. We depict three clusters of cells R (red), P (purple), and B (blue) in a high-dimensional space (**e**). The cells actually lie on a manifold (the gray curve) and Euclidean distance on the manifold is misleading, since it makes it appear that R is closer to B than to P, when, on the manifold, P should be between R and B. Schema can transform the data so that Euclidean distance is more reasonable. If we have some secondary data that tells us that distance(B, P) and distance(R, P) are less than distance(B, R), we can rotate the dataset and scale the axes so that the manifold distance agrees better with Euclidean distance.

We say that two datasets[12] agree if they have a similar *neighborhood structure*. Formally, we view each dataset as describing the same N points in two different spaces and say that they have identical *neighborhood structures* if for any point P and any k $(1 \leq k \leq N-1)$, the k-nearest neighbors of P are the same in both datasets. In other words, a nearest-neighbor algorithm can not distinguish between the two datasets[13]. Note that the distances need not be Euclidean, a feature we will exploit later.

This is an extremely strong[14] property, and we weaken it by considering instead the *correlation* of pairwise distances between the datasets. For this, consider the set of $\binom{N}{2} = \frac{N(N-1)}{2}$ pairwise distances in each dataset. We say the two datasets are in agreement if pairwise distances in one are highly correlated to the the other.[15]

Our goal is to combine the information from the input datasets (D_1, D_2, D_3 here) into a single dataset D^* that is in agreement with each of them. First, we designate one dataset (say, D_1 of shape $N \times k$) as the primary and the remaining datasets (here, D_2 and D_3) as secondary.[16] Specifically, we pose the following constrained optimization problem: find an embedding D^* (of shape $N \times k$) that is a linear transformation of D_1, where the pairwise squared Euclidean distances in D^* are i) maximally correlated to the corresponding distances in D_2 and D_3; and ii) correlated to the corresponding distances in the primary dataset D_1 above some user-specified

12: We will use *dataset*, *view*, and *modality* interchangeably

13: Alternatively, we can say that the ordering of all pairwise distances is preserved

14: and not necessarily desirable, due to brittleness

15: For computational tractability, we approximate the Spearman correlation with the Pearson correlation. The difference between the two is that the former works with ranks, while the latter works with raw values.

16: Choosing which dataset should be primary is sometimes obvious (e.g., when using meta-data). Typically, the dataset where short-range Euclidean distances would be most meaningful should be chosen (see the discussion related to manifold reshaping in Section 3.4). This is not a "reference" dataset in the sense of Stuart et al. [86]; here, the researcher is explicitly allowed to distort the primary dataset.

threshold. By tuning the threshold, the user can trade off distortion from the primary dataset D_1 against agreement with the secondary data in D_2 and D_3.

Introducing this constraint while keeping the optimization problem feasible is a key conceptual contribution of this work: methods like CCA, as well as standard approaches in metric learning, place no such limits on the distortion of D_1. Biologically, that is deeply problematic because it de-emphasizes information in D_1 that does not co-vary with D_2 and D_3[17].

17: And as we will see in Chapter 5, we are often confronted with highly uncertain information in the secondary modalities in real biological data — allowing a focus on the high-confidence modality lets us selectively use information from the lower-confidence views

We now describe our solution to this optimization problem. Any affine transform can be broken into three components: translation, scaling (i.e., stretch or shrink the axes), and rotation. Distances are invariant with respect to translation, so only the latter two are of interest here. *The key algorithmic contribution of this paper is a quadratic programming (QP) formulation that computes the optimal scaling transform for the optimization problem above*.

We pair this scaling transform with a rotation produced by a change-of-basis technique, thus producing an affine transform. In particular, we use methods like principal component analysis (PCA) or non-negative matrix factorization (NMF) for this change of basis. We have found this to be surprisingly effective: doing a PCA and NMF rotates the data so that dimensions with high variance or information become axis-aligned. The QP-computed scaling transform then acts as a feature selection mechanism on top of this, identifying which axes are the most useful in maximizing agreement between the datasets.

3.3.1 Motivating the choice of correlation as an objective

As a measure of the alignment between our transformation and a dataset, correlation of pairwise distances is a flexible and robust measure. Given a pair of dataset, the connection between their pairwise-distance Spearman rank correlation and the neighborhood-structure similarity is deep: if the correlation is greater than $1 - \epsilon$, the fraction of misaligned neighborhood-relationships will be less than $O(\sqrt{\epsilon})$. There is a *manifold* interpretation that is also compelling: assuming the high-dimensional data lie on a low-dimensional manifold, small distances are more accurate than large distances, so the *local* neighborhood structure is worth preserving. We can show intuitively that optimizing the correlation aims to preserve local neighborhood structure[18]. Using correlation in the objective also affords the flexibility to broaden $\text{Corr}(w, \rho_j)$ in (3.2) to any function

18: In the next section, a toy example where short-range distances are selectively preserved is shown

f_j of the metric: i.e. $\text{Corr}(w, f_j \circ \rho_j)$[19]; this allows us to, for example, invert the direction of alignment or more heavily weigh local distances[20].

As scRNA-seq dataset sizes reach millions of cells, even calculating the $O(N^2)$ pairwise distances becomes infeasible. In this case, we sample a subset of the pairwise distances. As an estimator, sample-correlation is a robust measure. This allows Schema to perform well even with relatively small subsets; in fact, we only need a sample-size *logarithmic* in our desired confidence level to generate high-confidence results (Appendix A). This enables Schema to continue scaling to more massive scRNA-seq datasets.

19: This is a slightly stronger version of the *kernel trick* from Subsection 2.3.2; specifically, f_j is a kernel function that depends only on the given distance between points (rather than a general kernel)

20: We explore a *bandwidth-limited* version of Schema below

3.4 Mathematical Formulation

Recall that we are assuming we have N observations across r datasets D_j, $1 \le j \le r$, where $D_j = \{x_i^{(j)}, 1 \le i \le N\}$ contains data (categorical or continuous) for each observation. We will refer to D_1 as the *primary* dataset and the rest as secondary. Each dataset's dimensionality and domain may vary. In particular, we assume D_1 is k-dimensional, i.e., $x_j^{(1)} \in \mathbb{R}^k$ for all j. For notational convenience, we drop the superscript when referring to the primary dataset and its data. Each dataset D_j must also have some notion of distance between observations attached to it, which we will denote ρ_j, so $\rho_j(x_n^{(j)}, x_m^{(j)})$ is the distance between observations n and m in D_j. Actually, since our entire framework below deals in *squared* distances[21], for notational convenience we will let ρ_j be the *squared* distances between points in D_j; also, we drop the superscript in $x_j^{(1)}$ when referring to the primary dataset D_1 and its data.

21: This is for computational reasons that will become apparent in Subsection 3.4.3

The goal is to find a transformation Ω such that $\Omega(D)$ generates a dataset D^* for which the Euclidean metric ρ^* on D^* "mediates" between the various metrics ρ_j, each informed by its respectively modality. Note that none of the ρ_j need to be Euclidean. The above setup is quite general[22], and we now specify the form of the transformation Ω and the criteria for balancing information from the various metrics. Here, we limit Ω to a *scaling transform*. That is, $\Omega(D) = \{\text{diag}(\omega)x \mid x \in D\}$ for some $\omega \in \mathbb{R}^k$[23]. The scaling transform ω acts as a feature-weighting mechanism: it chooses the *features* of D_1 that align the datasets best[24]. We note here that a natural extension would be allowing *general linear* transformations for Ω; however, in that context, the fast framework of quadratic programming would need to be substituted for the much slower framework of semidefinite programming.

22: Notably, other choices of both integration and other classes of transformation are easily incorporated

23: $\text{diag}(\omega)$ is a $k \times k$ diagonal matrix with ω as its diagonal entries

24: In other words, ω_i being large means that the ith coordinate of D_1 is important

To measure quality of integration between the modalities' metrics ρ_j, our approach here is to learn a metric ρ^* that preserves the neighborhood structure in each modality as well as possible. Our measure of the alignment between ρ^* and ρ_j is given by the Pearson correlation between pairwise squared distances under two metrics. Intuitively, maximizing the correlation coefficient encourages distances under ρ^* to be large when the corresponding ρ_j distances are large and vice versa. This can be seen from the formula:

$$\text{Corr}(\rho^*, \rho_j) = \frac{\text{Cov}\left[\rho^*, \rho_j\right]}{\left(\text{Var}\left[\rho^*\right]\text{Var}\left[\rho_j\right]\right)^{1/2}} \tag{3.1}$$

To deal with multiple modalities, we try to maximize the correlation between ρ^* and the distances on each of the metrics, allowing the user to specify how much each modality should be weighted. The theory can also allow hard constraints, whereby the correlation between the transformed data and some D_j has to be at least some value[25]. Our goal is thus to find:

25: But our current implementation only allows a hard constraint on the primary modality

$$\begin{aligned} &\max_{\omega \in \mathbb{R}^k} \sum_{j=1}^{r} \gamma_j \text{Corr}(\rho^*(\omega), \rho_j) \\ \text{subject to} \quad &\text{Corr}(\rho^*(\omega), \rho_j) \geq \phi_j \text{ for } j \in \{1, \ldots, r\} \end{aligned} \tag{3.2}$$

where γ_j and ϕ_j are hyperparameters that determine the importance of the various metrics. We have also highlighted that ρ^* is a function of ω and is determined entirely by the solution to (3.2). In the rest of our discussion, we will primarily refer to ω, rather than ρ^*.

In order to make this optimization feasible, we use the machinery of *quadratic programming*.

3.4.1 Setting up the quadratic program

Quadratic programming (QP)is a framework for constrained convex optimization problems that allows a quadratic term in the objective function and linear constraints. The general form is:

$$\begin{aligned} &\min_{v \in \mathbb{R}^s} v^T Q v + q^T v \\ \text{subject to} \quad &\\ &Gv \preceq h \\ &Av = b \end{aligned} \tag{3.3}$$

where Q is a positive semidefinite (psd) matrix, and the notation $y \preceq z$ means the inequality is true for each coordinate (i.e., $y_i \leq z_i$

for all i).

To put our optimization (3.2) in a QP formulation, we expand the covariance and variance terms in (3.1), and show that the covariance is *linear* in the transformation and variance is *quadratic*:

$$\text{Cov}(w, \rho_\ell) = \left(\frac{1}{|P|}a_\ell - \frac{1}{|P|^2}b_\ell\right) w \tag{3.4}$$

$$\text{Var}(w) = w^T \left(\frac{1}{|P|}S - \frac{1}{|P|^2}T\right) w \tag{3.5}$$

where a_ℓ and b_ℓ are k-dimensional vectors that depend only on D_ℓ; and S and T are $N \times k$ matrices that depend only on D_1; and P is the set of pairs of observations. It is also not hard to show that $(|P|^{-1}S - |P|^{-2}T)$ is psd, as required. For details of the derivation, see Subsection 3.4.3.

There is one more difficulty to address. The correlation is the *quotient* of the covariance and the standard deviation, and the QP framework cannot handle quotients or square roots. However, maximizing a quotient can be relaxed to maximizing the numerator (the covariance), minimizing the denominator (the variance), or both.

We now have the ingredients for the QP and can frame the optimization problem as:

$$\max_{w \in \mathbb{R}^k} \sum_{j=1}^{r} \gamma_r \text{Cov}(w, \rho_j^2) - \alpha \text{Var}(\rho^*) - \lambda ||w - \mathbf{1}||^2 \tag{3.6}$$

subject to

$$\text{Cov}(w, \rho_j) \geq \beta_j \text{ for } 1 \leq j \leq r$$
$$w \succeq \mathbf{0}$$

regularization: a technique to avoid overfitting by penalizing a predictor for being too "complex" and learning random noise in the training set

where $\mathbf{0}$ and $\mathbf{1}$ are the all-zeros and all-ones vectors (of the appropriate length) respectively. Here, λ is the hyperparameter for **regularization** of w, which we want to penalize for being too far away from the all-ones vector (i.e. equal weighting of all the features). One could also regularize the ℓ_2 norm of w alone (i.e. incorporate $-\lambda||w||^2$) which would encourage w to be small; we have found that empirically the choices yield similar results.

This program can be solved by standard QP solvers (see Subsection 3.4.3 for the full details of how to put the above program in canonical form for a solver), and the solution w^* can be used to transform unseen input data, using $\omega^* \in \mathbb{R}^k$, where $\omega_i^* = \sqrt{w_i^*}$.

3.4.2 Hyperparameters

A well-known challenge for machine learning algorithms is interpretability of hyperparameters. Here, the QP solver needs values for λ, α, and β, and specifying these in a principled way is a challenge for users. Our approach is thus to allow the user to specify more natural parameters. Specifically, we allow the user to specify minimum correlations between the pairwise distances in D^* and each of the D_i; and also the ratio of the largest value of w to its average value, Formally, the user can specify s_i and $\bar{w}$ such that:

$$\begin{aligned} \text{Corr}(\rho^*, \rho_i) &\geq s_i \text{ for } 1 \leq i \leq r \\ \frac{||w||_\infty}{||w||_1} &\leq \frac{\bar{w}}{k} \end{aligned} \tag{3.7}$$

While these quantities are not directly optimizable in our QP formulation (3.6), we can access them by varying the hyperparameters α, β, λ. We note that, in its current implementation, Schema supports the constraint s_i only for $i = 1$, i.e., the primary dataset. In the future, we intend to support all s_i.

Intuitively, we note that the choice of λ controls whether w satisfies $\bar{w}$; and α and β control whether the correlation constraints s_i are satisfied. To satisfy these constraints, we simply *grid search* across feasible values of $\{\alpha, \beta, \lambda\}$: we solve the QP for fixed values of α, β, λ, keeping only the solutions for which the $\{s_i, \bar{w}\}$ constraints are satisfied. Of these, we choose the most optimal. The efficiency of quadratic programming means that such a grid search is feasible, which gives users the benefit of easily interpretable and natural hyperparameters.

Preprocessing transforms

Standard linear decompositions, like PCA or NMF are useful as preprocessing steps for Schema, as they transform the features into a more meaningful basis[26]. PCA is a good choice in this regard because it decomposes along directions of high variance; NMF is slower, but has the advantage that it is designed for data that is non-negative (e.g., transcript counts). The transform ω that we generate can be interpreted as a *feature-weighting* mechanism, identifying the directions (in PCA) or factors (in NMF) most relevant to aligning the datasets. The user can also employ a feature-set that is a union of features from two methods (e.g., PCA and CCA).

26: And Schema's feature selection then acts on the *transformed* basis

3.4.3 Details of the quadratic program

For the interested reader, we fully specify the quadratic program here — proving that the covariance and variance are indeed linear and quadratic in the transformation respectively as claimed in (3.4) and (3.5)[27].

27: For those not interested in a somewhat tedious derivation, this section can be skipped without loss of continuity.

We introduce some notation to condense the expressions. Define $w \in \mathbb{R}^k$ where $w_i = \omega_i^2$, $\delta_{ij} \in \mathbb{R}^k$ with $(\delta_{ij})_s = ((x_i)_s - (x_j)_s)^2$ (i.e. squared elements of $x_i - x_j$) and, for convenience, let P be the set of pairs of observations $P = \{\{i, j\} : 1 \leq i \leq j \leq N\}$. Using the fact that the covariance between variables X and Y is given by $\mathrm{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$, and the variance as $\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}^2[X]$, we can expand:

$$\begin{aligned}
\mathrm{Cov}(w, \rho_\ell) &= \frac{1}{|P|} \sum_{\{i,j\} \in P} \rho_\ell(x_i^{(\ell)}, x_j^{(\ell)}) \delta_{ij}^T w \\
&\qquad - \frac{1}{|P|^2} \left(\sum_{\{i,j\} \in P} \delta_{ij}^T w \right) \left(\sum_{\{i,j\} \in P} \rho_\ell(x_i^{(\ell)}, x_j^{(\ell)}) \right) \\
&= \left(\frac{1}{|P|} a_\ell - \frac{1}{|P|^2} b_\ell \right)^T w \\
\mathrm{Var}(w) &= \frac{1}{|P|} \sum_{\{i,j\} \in P} w^T \delta_{ij} \delta_{ij}^T w - \frac{1}{|P|^2} \left(\sum_{\{i,j\} \in P} \delta_{ij}^T w \right)^2 \\
&= w^T \left(\frac{1}{|P|} S - \frac{1}{|P|^2} T \right) w
\end{aligned}$$

where a_ℓ and b_ℓ are k-dimensional vectors that depend only on D_ℓ; and S and T are $N \times k$ matrices that depend only on D_1.

Explicitly, we derive:

$$\begin{aligned}
a_\ell &= \sum_{\{i,j\} \in P} \rho_\ell(x_i^{(\ell)}, x_j^{(\ell)}) \delta_{ij} \\
b_\ell &= \left(\sum_{\{i,j\} \in P} \rho_\ell(x_i^{(\ell)}, x_j^{(\ell)}) \right) \sum_{\{i,j\} \in P} \delta_{ij} \\
S &= \sum_{\{i,j\} \in P} \delta_{ij} \delta_{ij}^T \\
T &= \left(\sum_{\{i,j\} \in P} \delta_{ij} \right) \left(\sum_{\{i,j\} \in P} \delta_{ij}^T \right)
\end{aligned}$$

We recall the general optimization problem (3.6) that needs to be

mapped to this framework:

$$\max_{w \in \mathbb{R}^k} \sum_{j=1}^{r} \gamma_r \text{Cov}(w, \rho_j) - \alpha \text{Var}(\rho^*) - \lambda ||w - \mathbf{1}||^2 \quad (3.8)$$

subject to

$$\text{Cov}(w, \rho_j) \geq \beta_j \text{ for } 1 \leq j \leq r$$
$$w \geq \mathbf{0}$$

and the framework for quadratic programming in (3.3) that this needs to be mapped to:

$$\min_{v \in \mathbb{R}^s} v^T Q v + q^T v \quad (3.9)$$

subject to

$$Gv \leq h$$
$$Av = b$$

The mapping is straightforward:

$$v = w$$
$$Q = \frac{1}{|P|} S - \frac{1}{|P|^2} T + \lambda I_k$$
$$q = -2\lambda \mathbf{1} - \sum_{j=1}^{r} \gamma_\ell \left(\frac{1}{|P|} a_\ell + \frac{1}{|P|^2} b_\ell \right)$$

We also require that Q be positive semidefinite (psd). This is also straightforward to show. We can write:

$$Q = \lambda I_k + \frac{1}{|P|} \sum_{\{i,j\} \in P} (\delta_{ij} - \mu)(\delta_{ij} - \mu)^T$$

where $\mu = \frac{1}{|P|} \sum_{\{i,j\} \in P} \delta_{ij}$, so it is a sum of psd matrices.

For the linear constraint, we express G as a block matrix:

$$G = \begin{pmatrix} H & \mathbf{0} \\ \mathbf{0} & -I_k \end{pmatrix}$$

where each row in H is given by:

$$H_j = -\frac{1}{|P|} a_j - \frac{1}{|P|^2} b_j \text{ for } 1 \leq j \leq r$$

and h is an $r + k$-dimensional vector, where:

$$h_j = \begin{cases} -\beta_j & \text{for } 1 \leq j \leq r \\ 0 & \text{for } r+1 \leq j \leq r+k \end{cases}$$

If $\beta_j = 0$ for some j (i.e. no correlation constraint) the corresponding rows can be deleted from H and h. We note here that while the framework can handle as many correlation constraints as the user desires, our current implementation only works with a correlation constraint on the primary dataset (i.e. $\beta_j = 0$ for all $j > 1$); we believe for most use cases, this will be sufficient.

We have no equality constraints in our optimization, so A and b from (3.9) are not needed.

3.4.4 Manifold reshaping

Recent work by Yu et al. [30] has provided evidence that even very high-dimensional biological data often actually lie on some implicit low-dimensional manifold. Thus, the proper way to measure the similarity between points is using *manifold's* metric and not the Euclidean metric of the high-dimensional ambient space. For machine learning applications that operate on distances between points, it is therefore crucial that we can access the *correct* distances, even if we do not know the manifold. This is, of course, the premise of manifold learning, a main theme running through this dissertation.

Assuming the data manifold is "smooth" (as practitioners usually do), the Euclidean distances between *nearby* points closely approximate the distances on the manifold. Therefore, we can assume that the small distances in the primary dataset are accurate and should be preserved, and as importantly, the ranking of the distances should be preserved. Concretely, if $\rho(x_i, x_j) > \rho(x_i, x_\ell)$, then we should encourage $\rho^*(x_i, x_j) > \rho^*(x_i, x_\ell)$; in other words, the local neighborhoods of points should not be distorted.

Suppose our dataset consists of well-separated clusters (common in scRNA-seq datasets, where the clusters could potentially represent different cell types/lines) and lies on a manifold as in Figure 3.2. The within-cluster distances are well represented by the Euclidean distance in the ambient space, but the between-cluster distances are not. The goal for our algorithm is to encourage *global* movement of the *entire* clusters, rather than distorting the neighborhoods within each cluster[28].

28: Another way to put this is that we want to transform *global* Euclidean distances in the primary modality but *not* local distances

We consider a simplified dataset and show that optimizing the Spearman rank coefficient encourages global and not local moves. Suppose we have two clusters, one centered at the origin (call it O), and the other (call it M) drawn from a Gaussian with mean μ and variance $\sigma^2 I_k$ (the analysis does not require an isotropic Gaussian, but it is cleaner)[29].

29: Again, this section is quite in the weeds — the reader can skip it without loss of continuity

We define here our notion of *local* and *global* movements. In the local scenario, we perturb the points in M randomly to generate T:

$$T = \{x + z \mid x \in M, z \sim \text{Normal}(\mathbf{0}, \tau^2 I_k)\}$$

In other words, the points in the cluster get some isotropic Gaussian noise with variance τ^2.

In the global scenario, we want all the points in the cluster to move in generally the same direction. The simplest way to achieve this is to choose a Gaussian with variance only in one direction. To that end, we fix a random vector v such that $\|v\| = 1$. Then, we generate another perturbation S analogously:

$$S = \{x + z \mid x \in M, z \sim \text{Normal}(\mathbf{0}, k\tau^2 v v^T)\}$$

Thus, S is the points in M perturbed in the direction of v. We multiply the variance by k to make sure that norms of variances of the Gaussians for both S and T are the same.

We consider the *inter*-cluster distances between S or T and O and compare them to the distances between M and O. The goal is to show that the global perturbation has a higher Spearman rank correlation with the original distances than the local perturbation.

First, we simplify the model: assume that μ is large with respect to the variances of the clusters, so we can approximate distances between *any* point in O and some x in any of M, S, T as $\|x\|$. We can also consider the clusters as random variables:

$$\begin{aligned} M &\sim \text{Normal}(\mu, \sigma^2 I_k) \\ S &= M + Z_S \\ Z_S &\sim \text{Normal}(\mathbf{0}, k\tau^2 v v^T) \\ T &= M + Z_T \\ Z_T &\sim \text{Normal}(\mathbf{0}, \tau^2 I_k) \end{aligned}$$

Note that S and T are both Gaussian as well. To evaluate which perturbation weakens the Spearman rank correlation the most, we determine the probability that the perturbation *swaps* the norms of

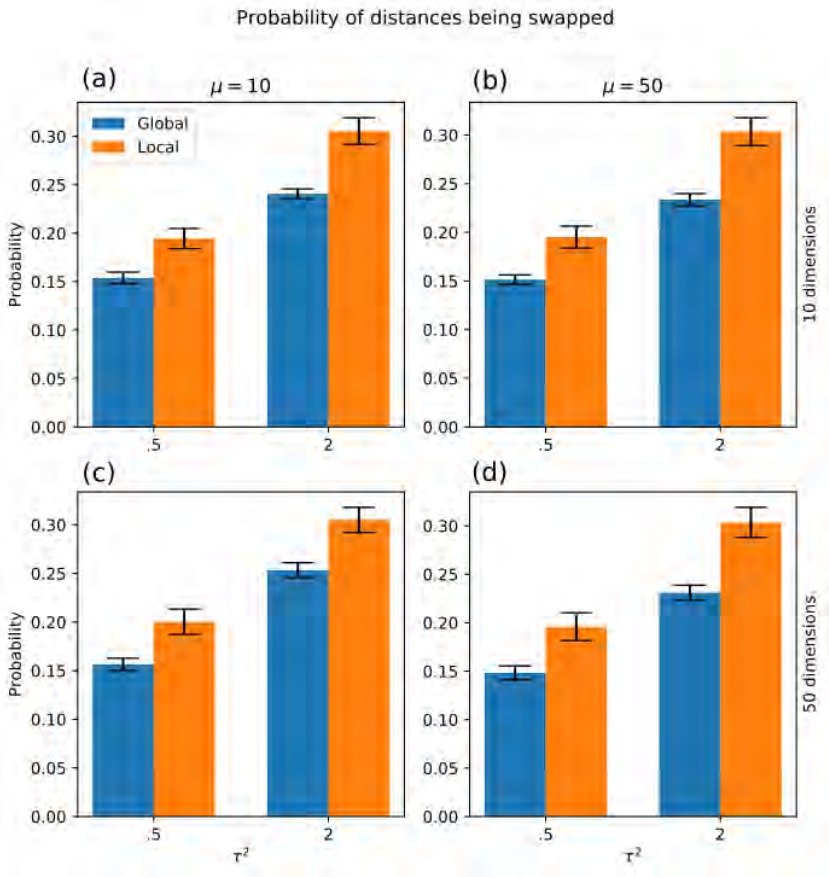

Figure 3.3: Empirical evaluation of global v. local perturbations. Comparing the effect of global and local perturbations of two point clouds on the Spearman correlation (comparing before and after perturbation). Note that across the parameter choices shown here, the global perturbation results in a higher Spearman correlation. This supports the manifold learning interpretation of Schema

two points in the cluster. Probabilistically, we compare:

$$\Pr(\|M_1 + (Z_T)_1\|^2 > \|M_2 + (Z_T)_2\|^2 \text{ given } \|M_1\|^2 < \|M_2\|^2)$$
$$\Pr(\|M_1 + (Z_S)_1\|^2 > \|M_2 + (Z_S)_2\|^2 \text{ given } \|M_1\|^2 < \|M_2\|^2)$$

Note that the norm of a multivariate Gaussian follows a χ^2-distribution, so the above probabilities can be simulated using that distribution. Figure 3.3 illustrates the outcome of this simulation. Across a wide range of parameter choices, the *global* perturbation causes fewer swaps or rank and therefore a higher Spearman rank correlation. In future work, we will aim to prove this fact analytically[30].

30: Essentially, this comes down to an analysis of non-standard χ^2 distributions, which are generally tricky but can usually be handled with clever tricks — the requisite bolt of inspiration has not yet struck

3.5 Limitations of Existing Metric Learning Tools

Unfortunately, existing metric learning methods [73–76]. are not well suited to the challenge of synthesizing multi-modal single-cell data. These methods, some of which list below, are designed to synthesize two datasets at a time, necessitating an ad hoc approach to integrating additional modalities. Like CCA, standard metric

learning approaches do not limit the distortion of the primary modality. Setting a researcher-specified limit on this distortion is an important regularization mechanism in Schema (as we will see in Chapter 5, when we evaluate the algorithms on real data), increasing the robustness of its results and ensuring that insights from the primary modality are not lost.

We designed Schema so it could scale to the large and ever-growing single-cell datasets. Towards that end, Schema deviates from existing metric learning approaches in computing a scaling transform and not a general affine transform. While affine transforms potentially offer more general alignment, Schema's ability to accept arbitrary distance metrics on the secondary modalities partly compensates for this limitation on the primary modality transform. Additionally, the reach of a scaling transform is enhanced by featurizing the primary modality so that each feature represents a source of variance that is axis-aligned and orthogonal to others (e.g., using PCA or NMF). Most crucially for our needs, scaling transforms can be computed efficiently; they can be optimized using fast quadratic programming methods whereas an affine transform would need to be optimized using the much slower framework of semi-definite programming[31]. Moreover, the ability to implement a *kernel* version of distance drastically increases the flexibility of even a scaling transform . Additionally, our choice of correlation as the measure of agreement allows for a sampling approach that further enhances scalability while producing provably accurate results (see Appendix A).

31: And other transforms like PCA and NMF with interpretable results can be used as pre- and post-processing steps

These design choices allow Schema to scale up to large single-cell datasets. We ran a set of metric learning algorithms on one of the Slide-seq samples [62] (puck 180430_1: 22943 cells $\times$18133 genes), using implementations made available on the creators' websites or in the Python package `metric_learn` [87]. We tested the following methods: neighborhood component analysis (NCA) [73]; metric learning for kernel regression (MLKR) [88]; local Fisher discriminant analysis (LFDA) [89]; large margin nearest neighbors (LMNN) [75]; and information theoretic metric learning (ITML) [74]. On a Linux server with 24 Intel Xeon 2.40 GHz cores and 386 GB RAM, each of these methods either crashed or failed to produce a meaningful output within 6 hours. In contrast, the aggregate runtime of an ensemble of Schema runs over different choices of the minimum correlation hyperparameter was 34 minutes on this dataset (see Table H.1); in each run, Schema sampled a subset of the pairwise distances between points.

3.6 Discussion

We designed Schema to be a powerful approach to multimodal data analysis. Schema is based on an elegant conceptual formulation in which each modality is defined using a distance metric. A key conceptual advance of this work is to formulate the synthesis task as a constrained optimization problem, allowing Schema to robustly accommodate noisy and sparse modalities. The strength of this intuition enables analysis of an arbitrary number of modalities and applicability to any modality, so long as it is possible to define an appropriate distance metric. Importantly, the synthesis is interpretable, with Schema identifying the features of the primary (i.e., reference) modality that drive the integration.

Our approach enables the researcher to supervise the synthesis by choosing which modality to transform, the degree to which it can be distorted, and the desired level of agreement between modalities. While existing methods like Seurat v3 [86] and LIGER [67] are designed for unsupervised discovery of common patterns across experiments, Schema's supervised formulation facilitates a broader set of investigations, enabling us to not only infer cell types and identify gene sets but also, for instance, rank amino acids by selection pressure.

When choosing a primary modality, we generally recommend selecting the most high-confidence modality or the one for which feature selection will be most informative, though it can sometimes be productive to integrate insights across multiple invocations of Schema with varying primary modality choices. In many of our demonstrations, we chose RNA-seq as the primary modality since it is often the modality where preprocessing and normalization are best understood, boosting our confidence in it; additionally, transformed RNA-seq data lends itself to a variety of downstream analyses. Once a primary modality has been designated, Schema can synthesize an arbitrary number of secondary modalities with it. In contrast, methods designed around pairwise modality comparison need *ad hoc* adaptations to accommodate additional modalities. Schema's approach is advantageous not only for datasets with more than two modalities[15, 90] but also in cases where metadata[32] can be productively incorporated as additional modalities.

32: for example, batch information and cell age

Intuitively, our correlation-based alignment approach has parallels to kernel canonical correlation analysis (kernel CCA), a generalization of CCA where arbitrary distance metrics[33] can be specified when correlating two datasets. While Schema offers similar flexibility for

33: as motivated by the general kernel methods discussed in Subsection 2.3.2

secondary modalities, it limits the primary modality to Euclidean distances. Introducing this restriction enhances scalability, interpretability and robustness. Unlike kernel CCA, the optimization in Schema operates on matrices whose size is independent of the dataset's size, enabling it to scale sub-linearly to massive single-cell datasets. Also, the optimal solution is a scaling transform that can be naturally interpreted as a feature-weight vector[34]. Perhaps most importantly, Schema differs from kernel CCA in performing a constrained optimization, thus reducing the distortion of the primary dataset and ensuring that sparse and low-confidence secondary datasets do not drown out the primary signal.

34: Of course, generalized kernel CCA has more power to express arbitrary functions

The constrained optimization in Schema acts as regularization, helping ensure that the computed transformation and feature selection remain biologically meaningful. By choosing a high-confidence modality as the primary modality and bounding its distortion when incorporating the secondary modalities, Schema enables information synthesis while retaining high-confidence insights. This bound on the distortion is an important parameter, directly controlling how much the secondary modalities inform the primary dataset[35]. Therefore, we recommend that studies using Schema for feature selection should aggregate the results over a range of values of this parameter while analyses that utilize only a single parameter should keep it high[36] to preserve fidelity with the original dataset. If sufficient data is available, cross-validation can also be used to tune this parameter. We strongly recommend that studies with a single parameter should report the value of this parameter alongside their results.

35: That is, values approaching 1 will increasingly limit the influence of the secondary modalities

36: at least 0.9; the default setting in our implementation is 0.99

3.6.1 Future explorations

Interesting future methodological work could explore alternative formulations of the Schema objective, potentially including more complex nonlinearities than our quadratic-program formulation. Schema can also be used in conjunction with data-integration methods [67, 86] designed for cases where each modality was assayed on different cells: after a cross-modality cell-to-cell correspondence has been computed, Schema can be applied to interpret the integrated data. It can also guide further biological experiments that profile only the highly-weighted features based on other data modalities, enabling efficient, targeted follow-up analysis.

We are also keen to build upon the connection with kernel CCA mentioned previously, with the perhaps obvious goal of combining the power of the arbitrary kernel with the efficiency of Schema. A major difficulty in adapting Schema to *general* kernel methods[37] is

37: Note that kernels can already be applied to the secondary modalities

that the *features* of the primary modality are weighted — and these features independently are meaningless for kernel methods. One approach would be to explicitly understand the way the kernel matrix varies as a function of the weights on each feature[38]. However, more directly, and very related to our discussion on manifold learning, would be to use a so-called *bandwidth-limited* kernel, which essentially only considers pairwise distances within a certain bandwidth. Intriguingly, some preliminary work emphasizing, for example, the small-scale distances[39] in the primary modality revealed interesting conclusions about semantics and global distances.

Given the current pace of biotechnological development, we anticipate that high-throughput experiments, and their conclusions, will increasingly rely on more than one data modality, underscoring the importance of Schema and its conceptual framework. Schema is publicly available for use[40] and as the Python package `schema_learn`.

38: This probably would not be possible for general kernels

39: which, recall, are those most important under the manifold hypothesis

40: `http://schema.csail.mit.edu`

4 Density-preserving Dimensionality Reduction*

Moving from the global transformations in the previous chapter, our second contribution is more squarely in the realm of *manifold learning*, as introduced in Section 2.5; and, while Schema *can* be used for dimensionality reduction (see Chapter 5), we now turn to methods explicitly designed for that purpose.

4.1 Analysis Begins with Visualization

Exploratory analyses of large-scale biological datasets typically begin with visualizing the data in low dimensions, in the hopes of revealing high-level structural insights to be probed in downstream analyses. This approach has been especially critical in the rapidly emerging field of single-cell transcriptomics (see Section 2.2), where high-throughput single-cell RNA sequencing (scRNA-seq) technologies are empowering researchers to study gene expression at an unprecedented resolution across diverse tissues, organisms, and biological conditions. Driven by the high-dimensionality of scRNA-seq datasets (thousands of different transcripts per cell) and their increasingly large-scale (hundreds of thousands of cells), many researchers rely on two- or three-dimensional data visualizations for quickly and intuitively finding structural patterns[1] and communicating biological insights with the scientific community [19, 92].

Two of the most popular techniques for high-dimensional data visualization are t-stochastic neighborhood embedding [38] (t-SNE) and uniform manifold approximation and projection [39] (UMAP), both of which have been widely adopted in scRNA-seq analysis [52,

1: We keep the notion of "structural" vague here, because, as we will see, the notion of structure worth preserving in high-dimensions is non-trivial; two well-accepted types of structure, though, are clusters and trajectories

* The text in this section is from the publication 'Assessing single-cell transcriptomic variability through density-preserving data visualization' by Narayan, Berger, and Cho [91]

93]. In contrast to traditional methods for dimensionality reduction, e.g. principal component analysis (PCA) (see Subsection 2.6.2), both t-SNE and UMAP learn a *nonlinear* embedding[2] of the original space by optimizing the embedding coordinates of individual data points using iterative algorithms. Both methods aim to accurately preserve the original local neighborhood of each data point in the visualization, while being more permissive of distortions in long-range distances. Because of the expressiveness of nonlinear embeddings, t-SNE and UMAP are well-regarded for their empirical performance at elucidating sophisticated manifold structures and clustering patterns in high-dimensional data [19, 92].

2: As noted in Subsection 2.6.3, this just means the embedding coordinates are not a linear function of the original coordinates

Despite their strengths, t-SNE and UMAP suffer from a major, often-overlooked pitfall: they neglect information about the local density of data points in the source dataset. In other words, data points whose neighbors are close-by in the original data are not distinguished in the visualization from those whose neighbors are far away[3]. This limitation leads to misleading visualizations where the apparent size of a cluster largely reflects the number of points in the cluster rather than its underlying heterogeneity. In scRNA-seq data, this omitted information about heterogeneity corresponds to the *variability* of gene expression within a subpopulation of cells. Thus, accurately portraying differences in local density in visualizations could provide another "dimension" of information, reflecting heretofore hidden insights into the transcriptomic landscape of single cells.

3: Importantly, here we are using *close-by* and *far away* in the *absolute* sense, i.e. the Euclidean distance is large, not the *ordinal* sense of close-by *compared to other points*

In this chapter, we introduce **density-preserving** data visualization methods den-SNE and densMAP that build upon t-SNE and UMAP, respectively, to enable researchers to more accurately visualize and extract deeper biological insights from the growing compendium of single-cell transcriptomic experiments. Our methods leverage the insight that, since both t-SNE and UMAP construct their embeddings by iteratively optimizing an objective function, we can augment that objective function with an auxiliary term that measures the distortion of **local density** at each data point in the visualization. To this end, we develop a general, differentiable measure of local density, called the **local radius**, which intuitively represents the average distance to the nearest neighbors of a given point. Our design of this measure enables efficient optimization of the density-augmented visualization objective. The algorithmic techniques we introduce could be used to augment other visualization tools based on iterative optimization and thus are of general interest.

local density: we define this precisely later on, but essentially, this measures how close the nearest neighbors of a point are to it

In Chapter 6, we demonstrate the utility of density-preserving visualization by applying den-SNE and densMAP to a diverse range

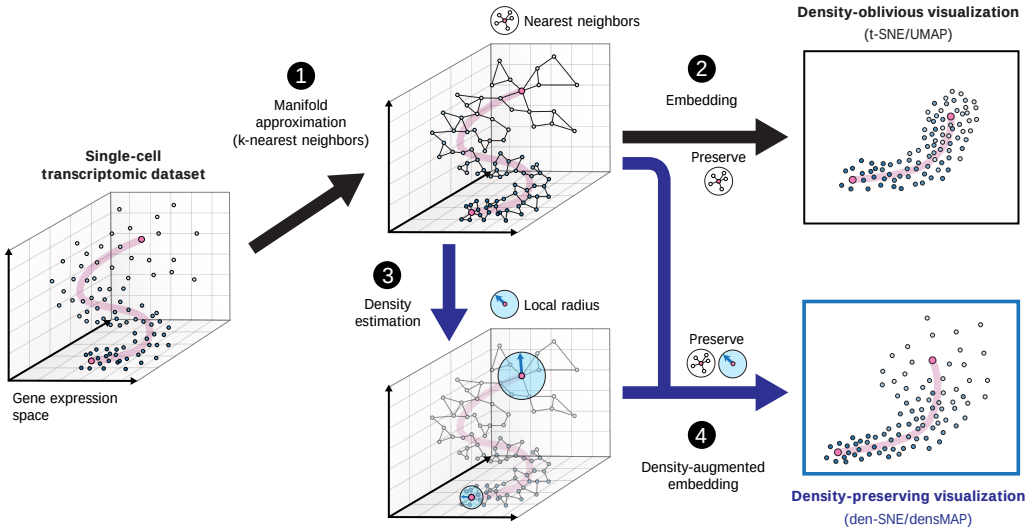

Figure 4.1: Overview of density-preserving data visualization. Given a set of points in a high-dimensional space as input (e.g. gene expression profiles from single-cell RNA-seq experiments), the goal of data visualization is to embed these points in 2D or 3D while preserving the structure of the original data. To this end, standard visualization tools t-SNE and UMAP first construct the k-nearest neighbor (KNN) graph as a compact summary of the data manifold (**1**). These methods then optimize the visualization coordinates of the points to maximally preserve local distances between neighbors in the graph (**2**). However, because t-SNE and UMAP adaptively choose length-scale to normalize local distances within each neighborhood, they produce visualizations that neglect information about density in the original space, thus omitting a key structural feature of the data. To enhance data visualization by incorporating density information, we introduce a general, differentiable measure of density called the local radius (Methods), which is efficiently calculated on the same KNN graphs that t-SNE and UMAP leverage (**3**). By augmenting the original visualization objective with a new term that encourages local radii to be consistent between the original space and the visualization, we transform both t-SNE and UMAP into density-preserving counterparts, den-SNE and densMAP, which more accurately portray the structure of the underlying data (**4**).

of published scRNA-seq datasets. Our work shows that density-preserving data visualization can unveil unforeseen patterns in single-cell transcriptomic landscapes and enrich our understanding of biology.

4.2 Overview of Density-preserving Data Visualization

Our density-preserving visualization methods, den-SNE and densMAP, augment t-SNE and UMAP respectively, generating embeddings that preserve both local structure and variability in the original data (Figure 4.1). To capture the local structure of the data, t-SNE and UMAP both create a nearest-neighbors graph (see Subsection 2.3.3) and preserve only the distances between neighboring points in this graph. We use the same nearest-neighbors graphs underpinning each of the original methods to calculate a *local radius* around each point, which represents the average distance from the point to its

nearest neighbors; this conveys the density of that point's neighborhood. The two original algorithms have an objective function that quantifies the agreement between a given embedding and the original nearest-neighbors graph, and they rearrange the embedding to maximize this agreement. We augment these objective functions with an additional term that measures the agreement between the local radii in the original dataset and in the embedding, ensuring that local structure is still preserved in the embedding while also conveying information about local variability. Our techniques have strong theoretical foundations, enable efficient optimization, and are easily generalized to other data visualization algorithms that similarly use gradient-based optimization.

Applying our methods to simulated datasets featuring heterogeneous density landscapes revealed the misleading visual conclusions that could be made without density preservation (Figure 4.2). Visualizing a mixture-of-Gaussian point clouds with different variances, t-SNE and UMAP generate clusters that are all similarly sized, while den-SNE and densMAP accurately depict the different variances (Figure 4.2a). When the point clouds are translated linearly with overlap, reflecting a trajectory, lack of density preservation in t-SNE and UMAP obscures dynamic changes in variability over the trajectory (Figure 4.2c). Conversely, when size is constant but a region is oversampled, t-SNE and UMAP overrepresent this oversampled region, giving the impression of increased variability and downplaying undersampled regions[4] (Figure 4.2b and d). Our following results show that these considerations are critical in biological analyses.

4: As the sizes of scRNA-seq datasets become ever-larger, the potential for representing rare transcriptomic states increases — they get ignored if underrepresented visually!

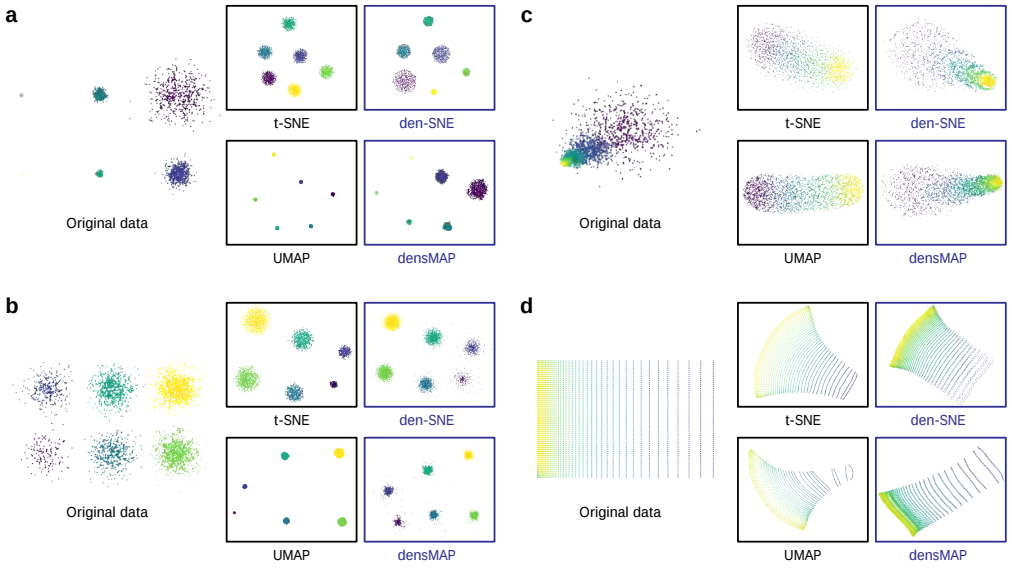

Figure 4.2: Density-preserving visualization more accurately captures the true underlying shape of synthetic datasets than existing tools. We compared the visualizations of our density-preserving methods den-SNE and densMAP to those of t-SNE and UMAP on different synthetic datasets: mixture-of-Gaussian point clouds with (**a**) increasing variances with the same sampling rate; (**b**) same variance, but with increasing sampling rates; (**c**) increasing variances in a linear translational motion with overlap, representing a temporal trajectory; and (**d**) a grid of points, whereby the density grows linearly in one direction. The synthetic datasets are generated in twenty dimensions for the point clouds and two dimensions for the grid, and the depictions of the original data in the figure represent two-dimensional linear projections for the former. While t-SNE and UMAP produce misleading visualizations where the apparent size of a cluster of points (marked by different colors) is unrelated to the amount of space it occupies in the original space and is biased by sampling rate, den-SNE and densMAP more accurately portray the shape of the original data by preserving density information.

4.3 Method Details

We now delve deeply into the details of our augmented methods, first laying a solid groundwork for both t-SNE and UMAP[5], highlighting precisely why they do *not* preserve density, and then explaining how we fix the problem.

5: and emphasizing especially their similar underlying philosophy

4.3.1 Review of t-SNE and UMAP.

The most widely-used nonlinear visualization algorithms in single-cell transcriptomic analysis are t-SNE [38] and UMAP [39], and both follow a similar methodology.

Outline of t-SNE and UMAP

Both t-SNE and UMAP build their embeddings using the following steps:

1. They first compute a nearest-neighbor graph of the high-dimensional data and introduce a type of probability distribution on the edges of this graph that assigns larger weights on smaller distances[6].
2. They then initialize an embedding in two-dimensions and define a similar distribution on pairwise distances in the embedding.
3. They then optimize the coordinates of the points in the embedding to minimizes the distance between this original probability distribution and a embedding distribution.

The key differences between the two algorithms lie in their choices of these distributions and the objective function quantifying the difference between the two distributions.

6: To be pedantic, for t-SNE, this distribution is over *all* edges, and for UMAP, it is over *each* edge

Let $X = \{x_i\}_{i=1}^n$ be our input dataset with n data points, where each $x_i \in \mathbb{R}^d$[7]. Let E be the set of edges (i, j) in the (directed) k-nearest neighbor graph constructed on this dataset[8], where j is one of the k points closest to i. For t-SNE, the probability distribution on the original data, $P_{ij}^{\text{t-SNE}}$, is given by normalizing and symmetrizing **Gaussian kernel distances**:

$$\begin{aligned} \tilde{P}_{j|i} &= \exp\left(-\left\|x_i - x_j\right\|^2 / \sigma_i^2\right) \\ Z_i &= \sum_{j:(i,j)\in E} \tilde{P}_{j|i} \\ P_{ij}^{\text{t-SNE}} &= \frac{1}{2n}\left(\frac{\tilde{P}_{j|i}}{Z_i} + \frac{\tilde{P}_{i|j}}{Z_j}\right) \end{aligned} \tag{4.1}$$

7: For our purposes, x_i will generally be the gene expression profile of a cell

8: See Subsection 2.3.3 to recall the difference between directed and undirected graphs

Gaussian kernel distance: given x and y, the gkd is the density value of x for a Gaussian distribution centered around y; equivalent to exponential kernel introduced in Subsection 2.5.3

where σ_i is chosen adaptively[9] for each i and corresponds the length-scale at x_i.

9: i.e. chosen at runtime dependent on the data — we detail exactly how below

UMAP uses a slightly different kernel, representing a rescaled exponential distribution:

$$\begin{aligned} \tilde{P}_{j|i} &= \exp\left(-(\left\|x_i - x_j\right\| - \text{dist}_i)/\gamma_i\right) \\ P_{ij}^{\text{UMAP}} &= \tilde{P}_{j|i} + \tilde{P}_{i|j} - \tilde{P}_{j|i}\tilde{P}_{i|j} \end{aligned} \tag{4.2}$$

where γ_i is chosen adaptively and also corresponds to the length-scale, and dist_i is the distance from x_i to its nearest neighbor. We expand on the role of σ_i and γ_i in the next section.

For the probability distributions computed on the embedding, both t-SNE and UMAP use a heavy-tailed distribution (e.g. Student's t-distribution for t-SNE), which emphasizes preserving local structure in the original dataset while being more lenient towards longer distances (see the original papers [38, 39] for a thorough explanation).

Formally, the probability distributions $Q_{ij}^{\text{t-SNE}}$ and Q_{ij}^{UMAP} in the embedding are defined as

$$\tilde{Q}_{ij}(a,b) = (1 + a d_{ij}^{2b})^{-1} \tag{4.3}$$

$$\mathcal{I}_i(a,b) = \sum_{j \neq i} \tilde{Q}_{ij}(a,b) \tag{4.4}$$

$$Q_{ij}^{\text{t-SNE}} = \tilde{Q}_{ij}(1,1) \left(\sum_k \mathcal{I}_k(1,1) \right)^{-1} \tag{4.5}$$

$$Q_{ij}^{\text{UMAP}} = \tilde{Q}_{ij}(a,b) \tag{4.6}$$

where d_{ij} represents the distance between points i and j in the embedding (Euclidean for both methods), and a and b are additional shape parameters UMAP introduces to control the spread of the distribution according to a user parameter. In the following, we omit the superscripts of P and Q when they are clear from the context.

The goal of both algorithms is to generate an embedding that minimizes the difference between P and Q. The loss function used by t-SNE to quantify this difference is the Kullback-Leibler (KL) divergence:

$$\text{KL}(P \| Q) = -\sum_{ij} P_{ij} (\log P_{ij} - \log Q_{ij}).$$

UMAP instead uses the cross-entropy (CE) loss summed over all the edges:

$$\text{CE}(P \| Q) = -\sum_{ij} P_{ij} \log Q_{ij} + (1 - P_{ij}) \log\big(1 - Q_{ij}\big).$$

Both methods optimize the embedding coordinates to minimize the respective loss functions using standard gradient descent optimization techniques (see Appendix C for details). Notably, the fact that UMAP does not require Q to be renormalized over all edges allows UMAP to use **stochastic gradient descent**, making it more computationally efficient than t-SNE in general.

stochastic gradient descent: a modification of the gradient descent optimization algorithm whereby the embedding coordinates are updated for a small chunk of data at a time

4.3.2 Adaptive length-scale selection in t-SNE and UMAP erases density information

The length-scale parameters σ_i and γ_i play an important role. The exponentially-decaying tails of the P distribution in both t-SNE and UMAP mean that the points a few multiples of the length-scale away from x_i are effectively omitted from the **conditional** distribution $P_{\cdot|i}$. Thus, the choice of the length-scale at point x_i determines the

conditional: briefly, the conditional distribution gives the probability of an event happening *given* that another event has happened

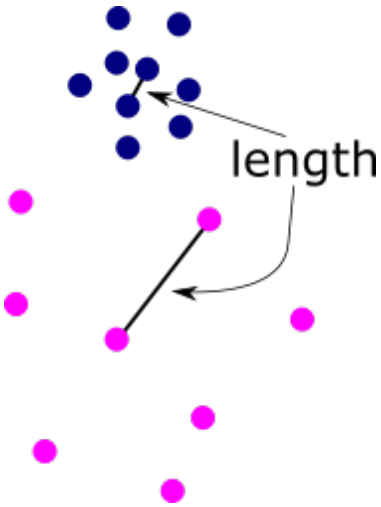

Figure 4.3: The need for different length-scales in a dataset; the top point cloud is much denser than the bottom

10: The concept of "smooth" here is imprecise — essentially, if a point's local neighborhood has only $k-1$ points rather than k points, then using k neighbors could result in poor performance; perplexity discounts points that are extremely far away even if k is chosen as the desired value.

11: i.e. more points are significantly represented in $P_{\cdot|i}$ as σ_i increases

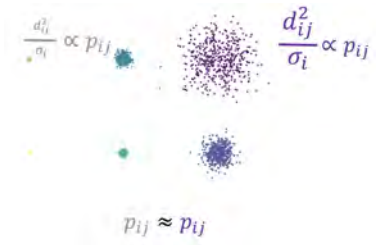

Figure 4.4: Adaptive length-scales cancel density information; since the length-scale term σ is proportional to the variance of the point cloud and the pairwise distances are divided by that term.

radius of the local structure around x_i that the embedding aims to preserve. Since different points in the dataset can have vastly different distribution of distances to their respective nearest neighbors, it is desirable to use a different σ_i or γ_i for each point x_i in order to evenly capture the local structure across all parts of the data (see Figure 4.3).

In t-SNE, the σ_i's are chosen by setting the *perplexity* of each conditional distribution $P_{\cdot|i}$ constant. Perplexity can be thought of as a "smooth" analog of the number of nearest neighbors[10] and is formally defined as $\text{Perp}_i = 2^{H_i}$, where H_i denotes the entropy of the conditional distribution $P_{\cdot|i}$:

$$H_i = -\sum_j P_{j|i} \log_2 P_{j|i}. \tag{4.7}$$

Since perplexity monotonically increases in[11] σ_i, t-SNE performs a binary search on each σ_i to obtain a constant perplexity for all i. UMAP's length-scale selection is analogous, but instead of fixing the value of perplexity, it fixes the marginal sum of probabilities at each point i, $\sum_j P_{ij}$, by choosing an appropriate γ_i.

Although it is effective for capturing local structure, adaptive choice of length-scale has the undesirable consequence of canceling out differences in density around each point in the original data, as t-SNE (implicitly) and UMAP (explicitly) both assume the data points are distributed *uniformly* on an underlying manifold. Note that, in both t-SNE and UMAP, a sparse neighborhood of x_i leads to a large length-scale, whereas a dense neighborhood leads to a small length-scale, as shown in Figure 4.4. Since the distance between points is divided by the length-scale parameter in the computation of P, we can intuitively see that this normalization removes density information from the data.

We can actually make this example more formal

Invariance under data dilation

More formally, consider a dataset of points $X = \{x_i\}_{i=1}^n$ with Euclidean pairwise distances $\{d_{ij}\}_{i,j=1}^n$. Suppose we dilate the data space by a factor of $\alpha > 1$ to generate a sparser dataset $Z = \{z_i\}_{i=1}^n$ with the same underlying structure, where the new pairwise distances are scaled by α, i.e. $\|z_i - z_j\| = \alpha d_{ij}$. A key observation is that the distribution P computed on X by t-SNE or UMAP will be *identical* to P computed on Z, even though Z represents a more heterogeneous set of points than X. Intuitively,

this is because obtaining the same perplexity/marginal sum of probabilities on Z requires that the respective length-scales be scaled by α, which cancels out the increase in distances and leaves the resulting P *unchanged*. Since P is the only information about the dataset provided as input to the embedding step of each algorithm, the original differences in density in different regions of the data space are entirely lost in the embedding.

We provide a more detailed description of this property and its generalization to a broader class of generative models for the underlying data in Section 4.5.

4.3.3 Capturing density information using the local radius

To generate embeddings that retain information about the density at each point, we introduce the notion of a **local radius** to make concrete our intuition of spatial density. Intuitively, a point is in a *dense* region if its nearest neighbors are very close to it, and in a *sparse* region if its nearest neighbors are far away. Thus, we use average distance to nearest neighbors[12] as a measure of density for a given point.

12: The skeptical reader may wonder how to decide how many nearest neighbors to use for this; this is indeed non-trivial and expanded on below

To formalize this notion, for a point x_i, we require two components: (i) a pairwise distance function $d(x_i, x_j)$, and (ii) a probability distribution $\rho_{j|i}$ that weighs each x_j based on its distance from[13] x_i, with faraway points having lower weights. We can now define the local radius:

13: That is, ρ is a function of the distance metric d

Definition 4.3.1 (local radius) *Given a distance metric d and a probability distribution ρ that is decreasing in d, the* local radius *at x_i, denoted $R_\rho(x_i)$, as the expectation of the distance function over x_j with respect to $\rho_{j|i}$, thus capturing the average distance from x_i to nearby points:*

$$R_\rho(x_i) := \mathbb{E}_{j\sim\rho_{j|i}}\left[d(x_i, x_j)\right]. \tag{4.8}$$

In the following, we let the distance function be the squared Euclidean distance, i.e. $d(x_i, x_j) = \left\|x_i - x_j\right\|^2$, which we found to have better empirical performance than standard Euclidean distance[14]. Other choices of distance function can be easily incorporated into our framework.

14: We do not yet have a theory for why this might be, and experimenting with other distance metrics might be helpful!

In den-SNE and densMAP, we take advantage of the probability distributions $P^{\text{t-SNE}}$ and P^{UMAP} which already capture local relationships; for the local radius in the original embedding, we renormal-

ize the edge probabilities P_{ij} to obtain a conditional distribution $\rho_{j|i} = P_{ij}/\sum_j P_{ij}$ and calculate the local radius as

$$R_P(x_i) = \frac{1}{\sum_j P_{ij}} \sum_j P_{ij} \|x_i - x_j\|^2$$

for both methods. Note that P vanishes rapidly outside the neighborhood of each x_i and is thus well-suited for density estimation. We can show in fact that this representation of density (inversely-related) has the desirable property that it scales with the *variance* of a range of data-generating distributions and increases when the length-scale term σ_i increases (Section 4.5).

Next, we define the local radius in the embedding. Let y_i be the embedding coordinates of the point x_i given by the algorithm of choice. We need a distribution analogous to P for calculating the expected distance between y_i and its neighbors in the embedding. It would still be desirable for this distribution to have adaptive length-scales like P in order to ensure that a comparable number of nearest neighbors are taken into consideration for calculating the local radius at different points in the dataset. However, this would present a major hurdle for optimization because the binary search used to determine σ_i and γ_i is not differentiable[15]. Instead, we leverage the embedding distribution Q computed by t-SNE and UMAP as an approximation for the adaptive scheme[16] Letting $\rho_{j|i} = Q_{ij}/\sum_j Q_{ij}$ and $d(y_i, y_j) = \|y_i - y_j\|^2$, the local radius in the embedding is given as

$$R_Q(y_i) = \frac{1}{\sum_j Q_{ij}} \sum_j Q_{ij} \left\|y_i - y_j\right\|^2. \tag{4.9}$$

Note that we adopt the squared Euclidean distance for consistency with local radius computation in the original space.

For ease of notation, we denote the local radius in the original data as R_o and the local radius in the embedding as R_e in the following sections.

15: In other words, we cannot determine the adaptive length scale in the process of the optimization

16: It is worth noting that, in the case of t-SNE, Q is based on a Cauchy distribution, which can be interpreted as the marginalization of a Gaussian distribution over an unknown variance [94]. Thus, Q intuitively reflects an average over all length-scales.

4.3.4 Augmenting the visualization objective to induce density preservation.

To preserve density, we aim for a *power-law* relationship between the local radius in the original dataset and in the embedding, i.e. $R_e(y_i) \approx \alpha\,[R_o(x_i)]^\beta$ for some α and β, inspired by the exponential scaling of density with respect to dimensionality (see Subsection 4.3.5). This

can be reframed as an *affine* relationship between the logarithms of the local radii, i.e.,

$$r_e(y_i) \approx \beta r_o(x_i) + \alpha,$$

where we define $r_o(x_i) := \log R_o(x_i)$ and $r_e(y_i) := \log R_e(y_i)$. The goodness-of-fit of this relationship can be measured by the *correlation coefficient*

$$\mathrm{Corr}(r_e, r_o) = \frac{\mathrm{Cov}(r_e, r_o)}{(\mathrm{Var}(r_e)\mathrm{Var}(r_o))^{1/2}}, \tag{4.10}$$

which is invariant to the parameters α and β[17]. $\mathrm{Cov}(\cdot,\cdot)$ denotes the covariance function, and $\mathrm{Var}(\cdot)$ denotes the variance function[18].

17: The invariance is useful because we do not need to know the parameters of the power law, just that one exists

18: Note that these quantities are estimated by considering the tuples $\{(x_i, y_i)\}_{i=1}^n$ as n independent samples from the same distribution; e.g., the mean of r_e is estimated as $\frac{1}{n}\sum_{i=1}^n r_e(y_i)$

Our density-preservation objective is to choose the embedding $\{y_i\}_{i=1}^n$ such that correlation between the log local radii of the original dataset and the embedding is maximized. This approach is closely related to canonical correlation analysis [95] (CCA), which finds a linear transformation of a dataset that maximizes its correlation with another. We are further motivated by recent work that extends CCA to nonlinear transformations [96] — our procedure can be interpreted as a nonlinear CCA, where we specify the nonlinear transform as the computation of the local radius.

Augmenting the loss functions of t-SNE and UMAP with this density-preservation objective yields the den-SNE and densMAP objectives, respectively:

$$\mathscr{L}^{\text{den-SNE}} = \mathrm{KL}(P^{\text{t-SNE}} \| Q^{\text{t-SNE}}) - \lambda \mathrm{Corr}(r_o^{\text{t-SNE}}, r_e^{\text{t-SNE}}), \tag{4.11}$$

$$\mathscr{L}^{\text{densMAP}} = \mathrm{CE}(P^{\text{UMAP}} \| Q^{\text{UMAP}}) - \lambda \mathrm{Corr}(r_o^{\text{UMAP}}, r_e^{\text{UMAP}}), \tag{4.12}$$

where λ is a user-chosen parameter[19] that determines the relative importance of the density-preservation term compared to the original objective.

19: In Chapter 6, we give some empirical guidelines for *how* one might choose this parameter, and show that, for the data sets we considered, the output embeddings are generally robust

4.3.5 Motivating the power-law relationship between embedded and original local radii

We motivate the connection between density-preservation and a power-law relationship between the original and embedding local radii with an example[20]. Suppose for a point $x \in \mathbb{R}^d$ in the original d-dimensional dataset, the K points in its neighborhood are uniformly distributed in a ball of radius γ_d and volume $V \propto \gamma_d^d$, as shown in Figure 4.5.

20: an extremely stylized example, but nevertheless helpful

Now, suppose we want to embed the dataset into $s < d$ dimensions while preserving structure and density. This means we want x and its neighbors to be mapped to an s-dimensional ball of uniform density with radius γ_s, and, to preserve the *density* of the K-neighborhood of x, the volume of the s-dimensional ball should still be V (see Figure 4.5). Since $V \propto \gamma_s^s$, this suggests a power law relationship between γ_s and γ_d, i.e. $\gamma_s \propto \gamma_d^{d-s}$. Taking logarithms, $\log \gamma_s = (d-s)\log \gamma_d + \beta$ for some β.

Figure 4.5: Scaling of density in a ball. As the dimensions increase, the volume of a ball increases for a given radius

Drawing the analogy between the local radius we defined and γ above, density preservation thus corresponds to a power law relationship between the local radii in the original and the embedded datasets.

4.3.6 Optimizing the embedding with respect to density-augmented objectives

Our differentiable formulation of the local radius enables us to optimize the density-augmented objective functions (4.11) and (4.12) using standard gradient descent techniques. Since both t-SNE and UMAP are also based on gradient descent, it suffices for us to calculate the contribution of the density-preservation objective to the overall gradient and add it to the existing t-SNE and UMAP gradients.

The gradient of the density-preservation objective with respect to the embedding coordinates y_i is given by

$$\nabla_{y_i} \mathrm{Corr}(r_e, r_o) = \sum_{j \neq i} \left[\frac{\partial}{\partial d_{ij}^2} \mathrm{Corr}(r_e, r_o) \right] (y_i - y_j),$$

where $d_{ij} = \|y_i - y_j\|$. To simplify the notation, let $\mu_e = \mathbb{E}[r_e]$, $r_i^e = r_e(y_i)$, and $r_i^o := \left(r_o(x_i) - \frac{1}{n}\sum_i r_o(x_i)\right) / \mathrm{Var}^{1/2}(r_o)$[21]. Now, the inner gradient term with respect to d_{ij}^2 can be calculated as

21: Note that the centering of r_i^o and normalizing by standard deviation does not depend on the embedding and thus can be precomputed

$$\frac{\partial}{\partial d_{ij}^2} \mathrm{Corr}(r_e, r_o) = \frac{\mathrm{Var}(r_e)\left(r_i^o \frac{\partial r_i^e}{\partial d_{ij}^2} + r_j^o \frac{\partial r_j^e}{\partial d_{ij}^2}\right) - \mathrm{Cov}(r_e, r_o)\left((r_i^e - \mu_e)\frac{\partial r_i^e}{\partial d_{ij}^2} + (r_j^o - \mu_e)\frac{\partial r_j^e}{\partial d_{ij}^2}\right)}{(n-1)\mathrm{Var}(r_e)^{\frac{3}{2}}},$$

where

$$\frac{\partial r_i^e}{\partial d_{ij}^2} = \frac{\tilde{Q}_{ij}^2(a,b)}{\mathfrak{Z}_i(a,b)} \left[abd_{ij}^{2(b-1)} + e^{-r_i^e}(1 + a(1-b)d_{ij}^2) \right].$$

The terms $\tilde{Q}_{ij}(a,b)$ and $Z_i(a,b)$, defined in (4.3) and (4.4), respectively, are quantities computed by t-SNE and UMAP to capture the local structure of the embedding[22]. Setting the parameters $a = b = 1$ results in the t-SNE formulation, whereas UMAP sets these two parameters as a function of a user parameter. A detailed derivation of our gradients above is provided in Appendix C.

22: $Z_i(a,b)$ is required only in t-SNE

Optimizing the densMAP objective requires special consideration because UMAP uses stochastic gradient descent (SGD), whereby edges are sampled according to P_{ij} and the gradient update is performed for one edge at a time. Since the gradient formula (4.10) involves a sum over its neighbors with equal weights, edges sampled from P must be re-weighted to obtain unbiased estimates of our gradient. To this end, we multiply the density term in the gradient for an edge $\{i,j\}$ by Z/nP_{ij} where $Z = \sum_{\{k,\ell\}\in E} P_{k\ell}$, to correct for sampling bias. In addition, there are a number of global terms that are computationally burdensome to update for every edge, which include $\mathrm{Var}(r_e)$, $\mathrm{Cov}(r_e, r_o)$, and μ_e. We compute these terms in the beginning of each **epoch** and consider them as fixed during the epoch. This can be viewed as a form of coordinate descent, where the objective is optimized with respect to a subset of variables at a time while conditioning on the rest. We describe these techniques in detail in Appendix C.

epoch: a round of edge-wise updates for the entire dataset

4.4 Implementation Details

To ensure that our methods find good local optima of (4.11) and (4.12) that are as effective as t-SNE and UMAP in separating clusters, we take a two-step approach where we run the original algorithms *without* the density-preserving objective for the first q fraction of iterations, then optimize the full objective for the remaining $1-q$ fraction of iterations[23]. We note that an alternative approach is to smoothly activate the density-preserving objective, but because any non-zero weight on this term incurs all of the associated computational overhead with little benefit, we opted for the two-step approach instead.

23: This approach is akin to t-SNE's "early exaggeration", whereby the first several iterations of the optimization emphasize attractive forces to help guide the direction of the optimization; it can also be thought of as a type of *simulated annealing*

For computational efficiency, we approximate the embedding distribution Q used in our local radius computation (4.9) by allowing Q_{ij} to be non-zero only when P_{ij} is non-zero (i.e. i and j are k-nearest neighbors in the original space), thus inducing sparsity in Q. This technique is especially well-suited for the aforementioned two-step approach, since the embedding already closely follows

the nearest-neighbor structure in P when this approximation takes effect.

There are several parameters of den-SNE and densMAP that the users can modify to tailor the behavior of these algorithms. We inherit all of the parameters from t-SNE and UMAP, including perplexity (t-SNE) or number of neighbors (UMAP), number of iterations/epochs, and the "min-dist" parameter for UMAP[24]. We refer the readers to the original publications for a detailed discussion of these parameters. There are two additional parameters we introduce in den-SNE and densMAP: the weight $\lambda \geq 0$ given to the density-preserving objective, and the fraction $q \in [0, 1]$ of iterations that take the density term into account. All of our experimental results are based on the following default parameter settings that we recommend. For den-SNE, we use perplexity of 50 and 1000 iterations (same as the default setting of t-SNE), along with $q = 0.3$ and $\lambda = 0.1$. For densMAP, we use 30 neighbors, 750 epochs, $q = 0.3$, and $\lambda = 2$. We note that changing the value of λ leads to qualitatively different embeddings that achieve different trade-offs between the original visualization objective and the density-preservation term (Figure G.20). For MNIST, we took advice from the scientific community and Kobak and Berens [97] to increase the early exaggeration parameter for t-SNE and den-SNE to 1,000, which resulted in better clustering of the digits [97].

24: This controls the a and b parameters in Q_{ij}; see (4.6)

4.4.1 Quantitative evaluation of density preservation

25: As noted above, assessing performance of visualization techniques is hard; here, our "assessment" is meant to show that density differences depicted are actually found in the original data

To assess the performance[25] of visualization algorithms at preserving density, we compute the correlation between the log local radii in the original dataset and two measures of visual density in the embedding generated by the algorithm.

26: Notably, we do not compute an adaptive length scale

The first measure of visual density is the local radius computed in the same manner as in the original space. Recall that during the optimization, we compute the local radius in the embedding *approximately* using the heavy-tailed distribution Q computed by t-SNE or UMAP and consider only the edges present in the nearest-neighbors graph of the original data[26]. For accurate evaluation, here we compute the local radius more directly as follows. Given the embedding points $\{y_i\}_{i=1}^n$, we compute the analog of the P matrix on the original data on these embedding points, denoted P'. For

t-SNE and den-SNE, we define P' as:

$$\begin{aligned}\tilde{P'}_{j|i} &= \exp\left(-\left\|y_i - y_j\right\|^2/\sigma_i\right)\\ Z'_i &= \sum_j P'_{j|i}\\ P'_{j|i} &= P'_{j|i}/Z'_i\end{aligned}$$

where σ'_i, the length-scale parameter, is chosen to achieve the same perplexity as in the original P matrix.

For UMAP and densMAP, we define P' as:

$$\begin{aligned}\tilde{P}'_{j|i} &= \exp\left(-(\left\|y_i - y_j\right\| - \mathrm{dist}_i)/\gamma'_i\right)\\ P'_{ij} &= (\tilde{P}'_{j|i} + \tilde{P}'_{i|j} - \tilde{P}'_{j|i}\tilde{P}'_{i|j})\\ P'_{j|i} &= P'_{ij}/\sum_{j\neq i}^{n} P'_{ij}\end{aligned}$$

where dist_i is the distance to the nearest neighbor of y_i, and γ'_i is chosen to achieve the same constant marginal $\sum_j P'_{j|i}$ as the original P matrix.

Since P' more explicitly focuses on the local neighborhoods of points in the embedding than Q by adaptively choosing the length-scale, calculating the local radius using this distribution more accurately reflects the actual density of each point in the embedding:

$$R_{P'}(y_i) = \sum_{j\neq i} P'_{j|i}\left\|y_i - y_j\right\|^2.$$

Note that the adaptive length-scale ensures that a similar number of neighbors are considered when computing the local radius for both dense and sparse neighborhoods in the embedding. Our quantitative metric of density preservation is then the Pearson correlation coefficient (R^2) between $\log R_{P'}(y_i)$ and $r_o(x_i) = \log R_P(x_i)$, where the latter is the log local radius in the original data space.

The second measure of visual density in the embedding is the *neighborhood count*, which is motivated by the visual perception of density as the number of points in a given area[27]. For a point y_i in the embedding and a radius ℓ, the ℓ-neighborhood count of y_i is the number of points y_j that are within a distance of ℓ from y_i in the embedding[28]. Thus, dense regions will have large neighborhood counts and sparse regions, small counts. This natural notion of local density has been extensively used in the psychology of vision [98, 99].

27: Of course, this is also the usual definition of density

28: An obvious question is why we do not use neighborhood count in the original space; this is because choosing ℓ in high-dimensions is extremely difficult — if the dataset exhibits vast variation in density, a particular choice of ℓ will not work across the data

To systematically choose ℓ for each dataset, we first compute the area A of the smallest bounding box of the embedded points, then calculate an average length-scale $\ell_{ave} = \sqrt{A/n}$[29], where n is the number of points in the dataset. To assess density preservation across different length-scales, we tested different multiples of ℓ_{ave}; for den-SNE and t-SNE, we chose ℓ from $\{\ell_{ave}, 2\ell_{ave}, 4\ell_{ave}\}$, and for densMAP and UMAP, from $\{\frac{1}{2}\ell_{ave}, \ell_{ave}, 2\ell_{ave}\}$[30]. For each chosen ℓ, we calculate the ℓ-neighborhood count for each point in the embedding and calculate the correlation (in log space) with the local radii in the original space as a quantitative metric of density preservation. A strong negative correlation is desirable, which indicates that points with a higher neighborhood count (higher visual density) tend to have a smaller local radius in the original dataset (smaller underlying variability).

29: We use the square root because the embeddings we consider are two-dimensional

30: We chose smaller values for densMAP and UMAP because those embeddings are more compact in general than those of den-SNE and t-SNE for our parameter choices

4.4.2 Additional metrics for evaluating visualization quality

We additionally evaluated the performance of our methods on three previously proposed metrics[31] of visualization quality on scRNA-seq data [52]: classification score (CS), mutual information score (MIS), and pairwise distance score (PDS). Intuitively, CS and MIS measure clustering accuracy based on the visualization, and PDS measures the preservation of pairwise distances among the datapoints.

31: Essentially, we needed to show that our notion of density preservation does *not* come at the cost of performance on existing methods

Additional metrics of visualization quality

- **Classification score (CS):** evaluates the accuracy of classifiers that assign each datapoint to one of the known[32] clusters based on the visualization coordinates. Following prior work [52], we trained a random forest classifier on the visualization[33] to predict the cluster labels from the original dataset using the `RandomForestClassifier` class in Python `scikit-learn` package with default parameters. We then calculated the CS as the accuracy of the trained classifier on a held-out test set[34]. We averaged the results across three trials of cross-validation to produce the final score.
- **Mutual information score (MIS):** measures the agreement between the output of a clustering algorithm in the original and the embedding space. As previously proposed [52], we used agglomerative clustering with $k = 100$ clusters to generate a high-resolution clustering of the original dataset, then applied the same procedure to obtain a clustering based

32: in other words, assuming a ground truth clustering

33: using 60% of the data

34: the remaining 40% of the data

on the visualization. We performed the clustering using `scikit-learn`'s `AgglomerativeClustering` class with the default **Ward linkage**. MIS is calculated as the **mutual information** between the two cluster assignments, which measures their agreement. To produce a robust estimate of the score, we computed MIS on three 60% subsamples of the original dataset and averaged the results.

- **Pairwise distance score (PDS):** we sampled $1,000$ points at random from the dataset and calculated the score as the squared correlation coefficient (R^2) between the pairwise distances among the chosen points in the original space and those in the visualization, again following the previously proposed approach [52]. Note that this score equally considers all pairs of points regardless of their distance, even though the nonlinear data visualization algorithms like t-SNE and UMAP are designed to focus on preserving distances within local neighborhoods[35]. To more comprehensively assess the preservation of pairwise distances at different scales in the original dataset, we calculated PDS for different subsets of pairwise distances with an increasing upper limit on their original distance in the dataset. More precisely, we calculated the PDS for the bottom x% of pairwise distances in the original space for x ranging from 0 to 100.

Ward linkage: briefly, the condition the clustering method uses to join two clusters together (hence *agglomerative*) is finding the pair that would least increase the variance of the new cluster

mutual information:given two random variables, X, and Y, their mutual information is a quantity from information theory that measures how the conditional distribution of X given Y is different from that of X independently

35: Not to put too fine a point on it, but this is essentially a metric UMAP and t-SNE *should not* perform well on — and under the manifold hypothesis, its utility is questionable too

4.4.3 Code availability

We provide the software for den-SNE and densMAP in the densVis package available at: `http://densvis.csail.mit.edu/` and `https://github.com/hhcho/densvis`. Our densMAP implementation is also available as part of the Python `umap` package (`https://github.com/lmcinnes/umap`) on the 0.5dev branch.

4.5 Theoretical Motivation for the Local Radius

Here[36], we motivate density-preservation by more rigorously showing that t-SNE does not preserve density due to its use of a constant perplexity for choosing the length-scale[37]. The setup is as follows.

Assume that for a given point X, we draw its n-nearest neighbors[38] as iid random variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ where each $X_i \in \mathbb{R}^d$ is drawn from a Gaussian distribution with mean X and covariance matrix[39] Σ. Let P_X be a row of the un-symmetrized probability

36: The following sections consist of our attempts to motivate our techniques with some more rigorous theory, for the interested reader; for those not inclined to following mathematical proofs, they can be skipped

37: Unfortunately, UMAP's use of a non-standard distribution for its P matrix (the re-scaled exponential distribution) precludes this analysis from being extended to UMAP

38: These are the neighbors used in the computation of t-SNE's P matrix

39: We note that we put no restrictions on the covariance; Σ can be any psd matrix

matrix induced by t-SNE, as in (4.1):

$$(P_X)_j = Z_X^{-1} \exp\left(-\left\|X - X_j\right\|^2 / \sigma_X^2\right)$$
$$Z_X = \sum_{j=1}^{n} \exp\left(-\left\|X - X_j\right\|^2 / \sigma_X^2\right).$$

The length-scale term σ_X is chosen to make the perplexity, Perp, constant as in (4.7):

$$\log \text{Perp} = \mathscr{H}_X = -Z_X^{-1} \sum_{j=1}^{n} (P_X)_j \log(P_X)_j + \log Z_X, \tag{4.13}$$

entropy: in physics, entropy is a measure of disorder or uncertainty; we do not concern ourself with the formal thermodynamic definition here, but a high entropy means more neighbors should be involved in the computation

where $\mathscr{H}_X$ is the **entropy** of P_X.

We showed above in Subsection 4.3.2 that dilating a set of points $X = \{x_i\}_{i=1^n}$ by multiplying the coordinates by some $\alpha > 1$ does not change the input probability distribution P for t-SNE or UMAP. We first observe that one can extend this result to a *collection* of sets of points $X_1, X_2, \ldots, X_k$, where, for some K, all the K nearest neighbors of a point $y \in X_\ell$ are *also* in[40] X_ℓ. Now, assume that each *collection* X_ℓ is scaled by some α_ℓ, and suppose we choose $s < K$ nearest neighbors to construct the nearest neighbors graph for the input distribution P (for either t-SNE or UMAP). Then, P is a *block diagonal* matrix: $P_{ij} = 0$ whenever x_i and x_j are from different collections. Thus, the length-scale terms for t-SNE and UMAP are computed on *each block* independently, and so each block is invariant to scaling the points in that collection by α_ℓ. Since each block of P does not change, the full matrix P does not change when each cluster is scaled independently, meaning that the density differences between the clusters after scaling are lost when the dataset is embedded. Thus, for our results below, we analyze the case with one point cloud, and note that the results generalize to well-separated[41] point clouds where each cloud is individually scaled.

40: In other words, these collections are clusters of points that are well separated from each other

41: While this can probably be more general, we use well-separated to refer to this block diagonal structure

4.5.1 Scaling of σ in t-SNE

We can consider σ_X as a function of the covariance of the generative model for $\mathbf{X}$, i.e. $\sigma_X = \sigma(\Sigma)$, since it is based on the length-scale, so σ_X itself can be thought of as a random variable. Notably, Vladymyrov and Carreira-Perpiñán [100] show that σ has a unique value that satisfies (4.13), so we just need to find *any* σ that satisfies the equation.

We first show that σ scales as the variance.

Proposition 4.5.1 *Suppose, we draw* $\mathbf{X} = \{X_1, \ldots, X_n\}$, $X_i \in \mathbb{R}^d$ *from a Gaussian distribution* $\mathcal{N}_X$ *with mean* X *and covariance* Σ, *and given* $\alpha > 0$, *we now draw* $\mathbf{Y} = \{Y_1, \ldots, Y_n\}$, $Y_i \in \mathbb{R}^d$ *from a Gaussian distribution* $\mathcal{N}_Y$ *with mean* Y *and covariance* $\alpha\Sigma$. *Let* σ_X *be chosen so that* $\mathscr{H}_X = \log \text{Perp}$ *for a constant value of* Perp. *Then, for any* $\phi, \delta \in (0, 1/4)$, *with* $n = O(\log(1/\delta)/\phi^2)$, *setting* $\sigma_Y = \alpha\sigma_X$ *will yield* $(1-\phi)\mathscr{H}_X < \mathscr{H}_Y < (1+\phi)\mathscr{H}_X$ *with probability at least* $1 - \delta$.

This is the proof that we go into in the most detail, and so we break it down into the following lemmas.

Proof outline

In order to prove that the length-scale σ scales as the variance of the point cloud, we show the following:

1. **Lemma 4.5.2** shows that an idealized version of entropy scales with the length-scale term
2. **Lemma 4.5.3** shows that this idealized version of entropy is not much different from the actual entropy calculated on the sample.
3. Combining the two results shows that the actual entropy also approximately scales with the length scale selection

Our proof involves approximating the sums on the right-hand side of (4.13) by expectations over the generating Gaussian distribution. To that end, we define:

$$\mathcal{G}_X(\sigma^2) = \frac{\mathbb{E}_{z\sim\mathcal{N}_X}\left[\exp\left(-\|z-X\|^2/\sigma^2\right)\|z-X\|^2/\sigma^2\right]}{\mathbb{E}_{z\sim\mathcal{N}_X}\left[\exp\left(-\|z-X\|^2/\sigma^2\right)\right]} + \log \mathbb{E}_{z\sim\mathcal{N}_X}\left[\exp\left(-\|z-X\|^2/\sigma^2\right)\right]. \tag{4.14}$$

We will show that $\mathscr{H}_X - \log n \to \mathcal{G}_X$ in the large-sample limit. We first show the behavior of $\mathcal{G}$ under dilations of the length-scale, i.e. $\mathcal{G}(\alpha\sigma^2)$:

Lemma 4.5.2 *Given the distributions* $\mathcal{N}_X$ *and* $\mathcal{N}_Y$ *defined above, and length-scale term* σ*:*

$$\mathcal{G}_Y(\alpha\sigma^2) = \mathcal{G}_X(\sigma^2).$$

That is, scaling σ *by the same amount as the covariance results in a constant value for* $\mathcal{G}$.

Proof. (Lemma 4.5.2) Introducing the simplifying notations $d(z) :=$

$\|z - X\|, p(z, \sigma) := \exp\left(-d(z)^2/\sigma^2\right)$, and $\mathbb{E}_X[\cdot] := \mathbb{E}_{z\sim\mathcal{N}_X}[\cdot]$, we write:

$$\mathcal{G}_X(\sigma^2) = \frac{\mathbb{E}_X\left[p(z,\sigma)\frac{d^2(z)}{\sigma^2}\right]}{\mathbb{E}_X[p(z,\sigma)]} + \log \mathbb{E}_X[p(z,\sigma)]. \tag{4.15}$$

Letting $f(z; \mu, \Sigma)$ denote the pdf of the multivariate normal distribution[42] with mean μ and covariance Σ, we can thus compute the expectations explicitly:

42: Throughout, we urge the reader to consider how one might generalize from this case; specifically, the scaling of the pdf with the variance appears to be the key factor

$$\begin{aligned}\mathbb{E}_X\left[p(z,\sigma)\frac{d^2(z)}{\sigma^2}\right] &= \frac{1}{\sigma^2}\int f(z; X, \Sigma)p(z,\sigma)d^2(z)\,\mathrm{d}z \\ &\propto \frac{1}{\sigma^2 \det \Sigma^{1/2}}\int \exp\left(-\frac{1}{2}((z-X)^T(\Sigma^{-1} + 2\sigma^{-2}I)(z-X))\right)d^2(z)\,\mathrm{d}z\,.\end{aligned} \tag{4.16}$$

The factors hidden by the proportionality symbol throughout are constants that depend only on the dimension and not on X, σ, or Σ.

We show now that we can, without loss of generality, assume that the covariance matrix Σ is diagonal. Intuitively, this is because the calculation of σ relies only on *distances*, which are invariant under orthogonal transformations, and we can transform to a coordinate system where Σ is diagonal. More formally, assume we have a non-diagonal covariance matrix. Then we can take its singular value decomposition $\Sigma = U\Lambda U^T$ where Λ is diagonal and U orthonormal[43]. Then, replacing the integration variable in (4.16) with $\theta = Uz$ and $\Phi = UX$, we see:

43: Note that the SVD takes the form of an eigendecomposition because Σ is psd

$$\mathbb{E}_X\left[p(z,\sigma)\frac{d^2(z)}{\sigma^2}\right] \propto \frac{1}{\sigma^2 \det \Lambda^{1/2}}\int \exp\left(-\frac{1}{2}((\theta-\Phi)^T(\Lambda^{-1} + 2\sigma^{-2}I)(\theta-\Phi))\right)d^2(\theta)\,\mathrm{d}\theta\,, \tag{4.17}$$

which is identical to (4.16).

The exponential term in the integral in (4.16) is the (unnormalized) pdf of a normal distribution with mean X and covariance matrix[44] $(\Sigma^{-1} + 2\sigma^{-2}I)^{-1} = \sigma^2\Sigma(\sigma^2 I + 2\Sigma)^{-1}$. Thus, (4.16) is:

44: by the Woodbury matrix identity

$$\propto C(\sigma, \Sigma)\frac{1}{\sigma^2 \det \Sigma^{1/2}}\int f(x; X, \sigma^2\Sigma(\sigma^2 I + 2\Sigma)^{-1})\|x - X\|^2\,\mathrm{d}x\,,$$

where we define C as the normalization factor for the normal distribution f inside the integral. The integral is thus the expectation

of the sum of the variances in each dimension of the distribution, i.e. the total variance, given by the trace of the variance matrix:

$$\mathbb{E}_X\left[p(x,\sigma)\frac{d^2(x)}{\sigma^2}\right] \propto \underbrace{\det\left[\sigma\Sigma^{1/2}(\sigma^2 I + 2\Sigma)^{-1/2}\right]}_{C(\sigma,\Sigma)} \frac{1}{\sigma^2 \det\Sigma^{1/2}} \underbrace{\operatorname{Tr}\left[\sigma^2\Sigma(\sigma^2 I + 2\Sigma)^{-1}\right]}_{\text{total variance of } f}. \tag{4.18}$$

Similarly,

$$\mathbb{E}_X\left[p(x,\sigma)\right] \propto \frac{1}{\det\Sigma^{1/2}} \int \exp\left(-\frac{1}{2}((x-X)^T(\Sigma^{-1} + 2\sigma^{-2}I)(x-X))\right) \mathrm{d}x\,,$$

and thus the integral is just the normalization of $f(x; X, \sigma^2\Sigma(\sigma^2 I + 2\Sigma)^{-1})$, so:

$$\mathbb{E}_X\left[p(x,\sigma)\right] \propto \underbrace{\det\left[\sigma\Sigma^{1/2}(\sigma^2 I + 2\Sigma)^{-1/2}\right]}_{C(\sigma,\Sigma)} \frac{1}{\det\Sigma^{1/2}}. \tag{4.19}$$

We thus have the form of $\mathcal{G}_X(\sigma^2)$ by plugging (4.18) and (4.19) into (4.15):

$$\mathcal{G}_X(\sigma^2) = C_1 \operatorname{Tr}\left[\Sigma(\sigma^2 I + 2\Sigma)^{-1}\right] + C_2 \log\left(\frac{\det\left[\sigma\Sigma^{1/2}(\sigma^2 I + 2\Sigma)^{-1/2}\right]}{\det\Sigma^{1/2}}\right), \tag{4.20}$$

where C_1 and C_2 are constants that do not depend on X, σ, or Σ[45]. Now, we turn to Y, whose neighbors are drawn from a normal distribution with mean Y and covariance $\alpha\Sigma$. Repeating the analyses above, it is easy[46] to see then that:

$$\mathcal{G}_Y(\tau^2) = C_1 \operatorname{Tr}\left[\alpha\Sigma(\tau^2 I + 2\alpha\Sigma)^{-1}\right] + C_2 \log\left(\frac{\det\left[\tau(\alpha\Sigma)^{1/2}(\tau^2 I + 2\alpha\Sigma)^{-1/2}\right]}{\det(\alpha\Sigma)^{1/2}}\right). \tag{4.21}$$

Plugging in $\tau = \alpha\sigma$, we see:

$$\begin{aligned}
\mathcal{G}_Y(\alpha\sigma^2) &= C_1 \operatorname{Tr}\left[\alpha\Sigma(\alpha\sigma^2 I + 2\alpha\Sigma)^{-1}\right] + C_2 \log\left(\frac{\det\left[\alpha\sigma(\alpha\Sigma)^{1/2}(\alpha\sigma^2 I + 2\alpha\Sigma)^{-1/2}\right]}{\det(\alpha\Sigma)^{1/2}}\right) \\
&= C_1 \operatorname{Tr}\left[\cancel{\alpha}\Sigma\cancel{\alpha^{-1}}(\sigma^2 I + 2\Sigma)^{-1}\right] + C_2 \log\left(\frac{\cancel{\alpha^{d/2}}\det\left[\sigma\Sigma^{1/2}(\sigma^2 I + 2\Sigma)^{-1/2}\right]}{\cancel{\alpha^{d/2}}\det(\Sigma)^{1/2}}\right) \\
&= \mathcal{G}_X(\sigma^2).
\end{aligned}$$

45: This will be used throughout, but can take on different values

46: in the way that mathematicians say things are easy, or an "exercise for the reader"

□

Thus, when the variance of the underlying distribution is scaled by α, the length-scale selection needs to also scale by α to keep $\mathcal{G}$ constant.

Next, we now show that, for sufficiently large n, the entropy $\mathscr{H}_X$ calculated on a sample and scaled appropriately by n, does not differ much from $\mathcal{G}_X$.

Lemma 4.5.3 *Suppose we draw* $\mathbf{X} = \{X_1, \ldots, X_n\}$ *i.i.d. from* $\mathcal{N}_X$. *Then, for any* $\alpha, \delta \in (0, 1/4)$ *and* $n > O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$, *we have:*

$$(1-\alpha)\mathcal{G}_X + \log(1-\alpha) < \mathscr{H}_X - \log n < (1+\alpha)\mathcal{G}_X + \log(1+\alpha)$$

with probability at least $1 - \delta$.

Proof. (Lemma 4.5.3) Using our notation from above, we have:

$$\mathcal{G}_X(\sigma^2) = \frac{\mathbb{E}_X[p(z,\sigma)d^2(z)]}{\sigma^2\mathbb{E}_X[p(z,\sigma)]} + \log \mathbb{E}_X[p(z,\sigma)]$$
$$\mathscr{H}_X = \frac{\sum_{j=1}^n p(X_j,\sigma)d^2(X_j)}{\sigma^2 \sum_{j=1}^n p(X_j,\sigma)} + \log\left(\sum_{j=1}^n p(X_j,\sigma)\right).$$

We use Hoeffding's inequality to show that:

$$\mathbb{E}_X[p(z,\sigma)d^2(z)] \approx \frac{1}{n}\sum_{j=1}^n p(X_j,\sigma)d^2(X_j)$$
$$\mathbb{E}_X[p(z,\sigma)] \approx \frac{1}{n}\sum_{j=1}^n p(X_j,\sigma).$$

Given $X_1, \ldots, X_n$ i.i.d. random variables drawn from a distribution F, bounded in $[0, s]$, Hoeffding's inequality quantifies how far the sample mean $\overline{X} = \frac{1}{n}\sum_i X_i$ deviates from its expectation $\mu = \mathbb{E}_{X\sim F}[X]$:

$$\Pr[\overline{X} > (1+\epsilon)\mu] < \exp\left(-\frac{\delta^2 n\mu}{3s}\right)$$
$$\Pr[\overline{X} < (1-\epsilon)\mu] < \exp\left(-\frac{\delta^2 n\mu}{2s}\right).$$

Taking the weaker bound (the first one) and setting the probability to δ, we can solve for n to see that $\overline{X}$ will be between $(1 \pm \epsilon)\mu$ when $n > \frac{3s\log(1/\delta)}{\mu\epsilon^2}$. Now, we note that the random variable $p(z, \sigma)$ lies

within $[0, 1]$, and the random variable $p(z, \sigma)\frac{d^2(z)}{\sigma^2}$ lies within $[0, 1/e]$, so we can set the range s to be 1. Thus, with sufficiently large n, we have that:

$$(1-\epsilon)\mathbb{E}_X[p(z,\sigma)d^2(z)] < \frac{1}{n}\sum_{j=1}^{n} p(X_j,\sigma)d^2(X_j) < (1+\epsilon)\mathbb{E}_X[p(z,\sigma)d^2(z)], \tag{4.22}$$

and

$$(1-\epsilon)\mathbb{E}_X[p(z,\sigma)] < \frac{1}{n}\sum_{j=1}^{n} p(X_j,\sigma) < (1+\epsilon)\mathbb{E}_X[p(z,\sigma)]. \tag{4.23}$$

Now, the above two equations (4.22) and (4.23) provide bounds for the numerator and the denominator of the first term of $\mathcal{G}_X(\sigma^2)$, respectively. Thus, in the worst case[47], we see that:

47: in other words, maximizing the numerator and minimizing the denominator

$$\begin{aligned}
&\frac{(1-\epsilon)\mathbb{E}_X[p(z,\sigma)d^2(z)/\sigma^2]}{(1+\epsilon)\mathbb{E}_X[p(z,\sigma)]} + \log(1-\epsilon) + \log\mathbb{E}[p(z,\sigma)] \\
&< \mathcal{H}_X - \log n \\
&< \frac{(1+\epsilon)\mathbb{E}_X[p(z,\sigma)d^2(z)/\sigma^2]}{(1-\epsilon)\mathbb{E}_X[p(z,\sigma)]} + \log(1+\epsilon) + \log\mathbb{E}[p(z,\sigma)].
\end{aligned}$$

To write these bounds in terms of $\mathcal{G}_X$, we note that $1+2\sqrt{\epsilon} > \frac{1+\epsilon}{1-\epsilon}$ and $1-2\sqrt{\epsilon} < \frac{1-\epsilon}{1+\epsilon}$ for $\epsilon < 1/4$, we can combine the above and compare $\mathcal{H}_X$ and $\mathcal{G}_X$:

$$(1-2\sqrt{\epsilon})\mathcal{G}_X + \log\left(1-2\sqrt{\epsilon}\right) < \mathcal{H}_X(\sigma) - \log n < (1+2\sqrt{\epsilon})\mathcal{G}_X + \log\left(1+2\sqrt{\epsilon}\right)..$$

Setting $\alpha = 2\sqrt{\epsilon}$ gives us the desired bounds. □

With these two results, we can prove Proposition 4.5.1.

Proof. (Proposition 4.5.1) By Lemma 4.5.2, $\mathcal{G}_Y(\alpha\sigma_X^2) = \mathcal{G}_X(\sigma_X^2)$. By our concentration bounds in Lemma 4.5.3, we take enough samples so that $\mathcal{H}_X(\sigma_X^2)$ is within a $(1+\beta)$ multiplicative factor of $\mathcal{G}_X$. Similarly, with as many samples, we know that $\mathcal{H}_Y(\alpha\sigma^2)$ is within a $(1+\beta)$ multiplicative factor of $\mathcal{G}_Y$. Thus in the worst case:

$$(1-\beta)^2\mathcal{H}_X < \mathcal{H}_Y < (1+\beta)^2\mathcal{H}_X$$

Taking $\beta = \frac{1}{3}\phi$ (which allows us to discard the quadratic terms), we see the above can be relaxed to

$$(1-\phi)\mathcal{H}_X < \mathcal{H}_Y < (1+\phi)\mathcal{H}_X,$$

as required. □

For the remainder of this section we will assume that the quantities calculated can be well approximated by their expectations, as we do above by approximating $\mathscr{H}_X$ with $\mathcal{G}_X$. For all the below propositions, Hoeffding's inequality can be used as in Lemma 4.5.3 to achieve arbitrarily close approximations, logarithmic in the number of samples[48].

48: We should note that the proportionality factors that are subsumed might be significant, but, even from a practical standpoint, the datasets we consider are large enough that asymptotics are probably reasonable

We extend the above analysis to the case of *uniformly* distributed data.

Proposition 4.5.4 *Let X be such that its n-nearest neighbors are distributed uniformly in a ball B_X of radius γ, and Y another point whose neighbors are distributed uniformly in a ball B_Y of radius $\sqrt{\alpha}\gamma$. Then, $\mathcal{G}_Y(\alpha\sigma^2) = \mathcal{G}_X(\sigma^2)$ where $\mathcal{G}$ is as defined analogously to* (4.15)*:*

$$\mathcal{G}_X(\sigma^2) = \frac{\mathbb{E}_{z\sim B_X}\left[p(z,\sigma)\frac{d^2(z)}{\sigma^2}\right]}{\mathbb{E}_{z\sim B_X}[p(z,\sigma)]} + \log \mathbb{E}_{z\sim B_X}[p(z,\sigma)].$$

Proof. As in Proposition 4.5.1, we then explicitly compute the terms $\mathbb{E}_X\left[p(z,\sigma)\frac{d^2(z)}{\sigma^2}\right]$ and $\mathbb{E}_X\left[p(z,\sigma)\right]$, assuming a uniform distribution now instead of a Gaussian:

$$\mathbb{E}_X\left[p(z,\sigma)\frac{d^2(z)}{\sigma^2}\right] \propto \frac{1}{\gamma^d\sigma^2} \underbrace{\int \exp\left(-\|z-X\|^2/\sigma^2\right) d^2(z)\,\mathrm{d}z}_{\text{total variance of (unnormalized) } \mathcal{N}(0,(1/2)\sigma^2 I)}$$

$$\propto \frac{\sigma^d}{2^{d/2}\gamma^d\sigma^2}\frac{d}{2}\sigma^2 \tag{4.24}$$

$$\propto \frac{\sigma^d}{\gamma^d}, \tag{4.25}$$

where the factors of 2 and d are absorbed into the proportionality

constant. Similarly:

$$\begin{aligned}\mathbb{E}_X\left[p(z,\sigma)\right] &\propto \frac{1}{\gamma^d}\underbrace{\int \exp\left(-\|z-X\|^2/\sigma^2\right)\mathrm{d}z}_{\text{normalization factor of } \mathcal{N}(0,(1/2)\sigma^2 I)} \\ &\propto \frac{\sigma^d}{2^{d/2}\gamma^d} \\ &\propto \frac{\sigma^d}{\gamma^d}.\end{aligned}$$

Repeating the above calculations for Y shows that:

$$\mathbb{E}_Y\left[p(y,\sigma)\frac{d^2(y)}{\sigma^2}\right] \propto \mathbb{E}_Y\left[p(x,\sigma)\right] \propto \frac{\sigma^d}{\alpha^d\gamma^d}.$$

Choosing σ_X^2 to achieve target perplexity for X, we see that setting $\sigma_Y^2 = \alpha\sigma_X^2$ makes the above terms equal to their corresponding terms in X and thus achieves the target perplexity. □

4.5.2 Scaling of the local radius with variance and length-scale

We now turn to the definition of local radius. Recall from Equation 4.8 in Subsection 4.3.3 that our local radius is defined as:

$$R_{P_X} = Z_X^{-1}\sum_j d^2(X_j)p(X_j,\sigma_X).$$

So, assuming the samples are drawn from generating distribution F_X, when we approximate this in the large-sample limit by $T_X(\sigma^2)$, as we did for the length-scale, we have:

$$T_X(\sigma^2) = \frac{\mathbb{E}_{z\sim F_X}[d^2(z)p(z,\sigma)]}{\mathbb{E}_{z\sim F_X}[p(z,\sigma)]}. \tag{4.26}$$

For the Gaussian and uniform generating distributions discussed in Propositions 4.5.1 and 4.5.4, it is straightforward to show that the local radius scales with the variance of the underlying distribution[49].

Proposition 4.5.5 *Let F_X and F_Y be a Gaussian or a spherical uniform distribution centered at X and Y respectively. For the Gaussian case, assume X has a covariance matrix Σ, and for the uniform distribution assume a radius γ; Y has covariance $\alpha\Sigma$ or radius $\sqrt{\alpha}\gamma$. Then, given σ_X^2,*

49: This is a slightly indirect way of showing that the local radius is related to the density — for a given sampling depth, a larger variance indicates lower density

the length-scale for X, and $\sigma_Y^2 = \alpha\sigma_X^2$ as in Propositions 4.5.1 and 4.5.4, $T_Y(\sigma_Y^2) = \alpha T_X(\sigma_X^2)$.

Proof. Note that for a given distribution,

$$T_X(\sigma_X^2) = \sigma_X^2 \mathcal{G}_X(\sigma_X^2) - \log \mathbb{E}_X[p(z, \sigma_X)].$$

If F is Gaussian, then plugging the value of $\mathcal{G}_X(\sigma_X^2)$ from (4.20):

$$T_X(\sigma_X^2) \propto \sigma_X^2 \operatorname{Tr}\left[\Sigma(2\sigma_X^2 I + \Sigma)^{-1}\right],$$

and if F is uniform, then from (4.25):

$$T_X(\sigma_X^2) \propto \sigma_X^2 \frac{\sigma_X^d}{\gamma^d \sigma_X^2} \sigma_X^2 \propto \frac{\sigma_X^d}{\gamma^d} \sigma_X^2.$$

Now, plugging in $\sigma_Y^2 = \alpha\sigma_X^2$ into corresponding equations for $T_Y(\sigma_Y^2)$, we see, in the Gaussian case:

$$T_Y(\alpha\sigma_X^2) = C_1 \alpha\sigma_X^2 \operatorname{Tr}\left[\alpha\Sigma(\alpha\sigma^2 I + 2\alpha\Sigma)^{-1}\right] = \alpha T_X(\sigma_X^2),$$

where the proportionality constant is the same for T_X and T_Y (as it does not depend on the covariance). In the uniform case:

$$T_Y(\alpha\sigma_X^2) = C_1 \left(\frac{\alpha\sigma_X}{\alpha\gamma}\right)^d \alpha\sigma_X^2 = \alpha T_X(\sigma_X^2),$$

as required. □

50: for now . . .

For more general distributions we are unable[50] to show the explicit *linear* scaling of local radius with the variance of the underlying distribution as above. However, we can still connect the local radius to the length-scale parameter σ chosen by t-SNE, which itself is known to empirically proxy the length scale at each point.

We show that the local radius R_o in the original space is an increasing function of the length-scale parameter σ. Thus, the local radius recaptures the length-scale information lost when normalizing by σ. The proposition below builds off the connection made by Vladymyrov and Carreira-Perpiñán [100] between the t-SNE Gaussian kernel and the partition function from thermodynamics[51].

51: We again do not expand deeply on the thermodynamic connection, but the partition function is crucial in understanding the entropy of a system

Here, we assume again that given a point X, its n-nearest neighbors are given by $X_1, \ldots, X_n$, but unlike before, we assume *no* knowledge of the distribution of the X_j. Thus, we cannot use expectations of known distributions for large-sample approximations and instead

consider a discrete distribution. To elucidate, define $\beta = \frac{1}{\sigma^2}$ and rewrite $R_X(\beta) := R_{P_X}(\sigma^2)$ from above suggestively[52] as:

52: ;)

$$R_X(\beta) = \sum_{j=1}^{n} \frac{\tilde{q}(X_j, \beta)}{Z_X(\beta)} d^2(X_j),$$

where $\tilde{q}(z, \beta) = \exp\left(-\beta d^2(X_j)\right) = p(z, \sigma)$, and we make clear the sum Z_X's dependence on β. Now, we note that since $Z_X(\beta) = \sum_j \tilde{q}(X_j, \beta)$, we can treat $q(X_j, \beta) := \tilde{q}(X_j, \beta) Z_X^{-1}(\beta)$ as probabilities from a *discrete* distribution, i.e. since $\sum_j q(X_j, \beta) = 1$[53]. Using the moments of this discrete distribution, we can show that the local radius is a decreasing function of β (and so an increasing function of the length-scale σ).

53: For the physicists in the room, β plays the role of inverse temperature in this system

Proposition 4.5.6 *Let $R_X(\beta) = \sum_j q(X_j, \beta) d^2(X_j)$, where $\sigma = \beta^{-1/2}$ is chosen as before, to ensure constant entropy*[54]*. Then $\frac{\partial R_X}{\partial \sigma} > 0$.*

54: The way β has to change to match a given entropy is akin to how systems with different intrinsic order need to vary their temperature to achieve a given entropy

Proof. Consider the expectation of d^2, taken over the distribution q:

$$\mathbb{E}_{X_j \sim q(\beta)}[d^2] = \sum_{j=1}^{n} q(X_j) d^2(X_j) = R_X(\beta). \tag{4.27}$$

Thus, R_X is expectation of the variable d^2 over the discrete distribution q.

Now, we aim to understand the derivative of R_X with respect to β. First it is straightforward[55] to verify:

55: Here, the computation is tedious but direct

$$R_X(\beta) = Z_X^{-1}(\beta) \sum_j d^2(X_j) \exp\left(-\beta d^2(X_j)\right) = -\frac{1}{Z_X} \frac{\partial Z_X}{\partial \beta}.$$

Now, take the derivative:

$$\begin{aligned} R_X'(\beta) &= \frac{1}{Z_X^2} \left(\frac{\partial Z_X}{\partial \beta} \right)^2 - \frac{1}{Z_X} \frac{\partial^2 Z_X}{\partial \beta^2} \\ &= R_X^2 - \frac{1}{Z_X} \frac{\partial^2 Z_X}{\partial \beta^2}. \end{aligned}$$

By (4.27), we see that the first term is $\left(\mathbb{E}_{q(\beta)}[d^2]\right)^2$. Expanding the second term we see:

$$\begin{aligned} \frac{1}{Z_X} \frac{\partial^2 Z_X}{\partial \beta^2} &= \frac{1}{Z_X} \sum_j d^4(X_j) \exp\left(-\beta d^2(X_j)\right) \\ &= \mathbb{E}_{X_j \sim q(\beta)}[(d^2)^2], \end{aligned}$$

so this term is the second moment of the variable d^2 over the distribution q.

Thus, we have

$$R'_X(\beta) = \left(\mathbb{E}_{q(\beta)}[d^2]\right)^2 - \mathbb{E}_{q(\beta)}[(d^2)^2] = -\mathrm{Var}_q(d^2) < 0.$$

where $\mathrm{Var}_q(d^2)$ is the variance of d^2 calculated over q and is therefore positive.

Since R_X is a decreasing function of β, that means it is an increasing function of σ, since $\sigma = 1/\sqrt{\beta}$ is monotonic.

□

4.6 Discussion

Effective tools for visualizing the single-cell landscapes captured by ever-larger single-cell experiments are pivotal for accelerating and disseminating discoveries. den-SNE and densMAP overcome a major limitation of the state-of-the-art tools t-SNE and UMAP: that they neglect differences in the *local variability* of gene expression across the transcriptomic landscape. While t-SNE and UMAP remain useful for revealing clustering or trajectory patterns, we demonstrated on a range of datasets that the local density information we incorporate into our visualizations harbors insights that can enrich our understanding of biology beyond what existing visualization tools offer. Our density-preservation techniques are broadly applicable to other visualization algorithms, including recent extensions of t-SNE [101, 102] and force-directed layout embedding [31, 103] (FDLE), and also to other types of biological data where visualization has been useful, such as scATAC-seq [104] and metagenomics [105].

In theory, targeted analyses could also capture the changes in transcriptomic variability made apparent by our visualizations[56]. However, by visualizing this information over the entire dataset, our approach allows easier interpretation and understanding. This methodological shift is akin to how t-SNE and UMAP have streamlined cell-type identification workflows by visually revealing clustering patterns in the dataset, despite the fact that clustering algorithms could be applied independently of visualization. Similarly, our methods can help researchers to easily grasp variability changes in their data and, consequently, to generate new biological hypotheses.

56: for example, by comparing the variance of gene expression between cell types [106]

Its analytical benefits aside, density-preserving visualization, as our results illustrate, more faithfully represents the underlying structure

of the dataset. Even as the community becomes increasingly aware of the intricate limitations of existing visualization tools, inaccurate visualizations will continue to expose researchers to potential biases in data interpretation. A large body of work in the social sciences highlights the problematic nature of inaccurate visualizations: for example, even though distortions in Mercator projections of the world map are well-known, they still suggest biased conclusions to viewers [107, 108]. Our density-preserving visualization tools will reduce such distortions and can help prevent unintentional biases and misdirection when researchers interpret and share insights from these data.

Our work motivates a number of interesting directions for further research, as we will detail in Chapter 6. First, the changes in transcriptomic variability we discovered in tumor-infiltrating immune cells suggest *differential variability* as a general tool for characterizing different cell states. A change in variability likely reflects underlying alterations of gene regulatory programs, and identifying the key drivers of this pattern and their roles merits further exploration. Our visualizations also motivate local density measures for noise reduction, as they often reveal fine-grain structure within a cell type, typically a dense "core" surrounded by a sparse cloud of cells with more divergent expression patterns. By focusing on only this core, one could obtain crisper canonical representations of cell states and developmental trajectories. Lastly, other popular tools for scRNA-seq analysis based on the nearest-neighbors representation of the transcriptomic landscape may also benefit from information about local variability, motivating density-augmented algorithms for tasks such as clustering [109], trajectory analysis [21], and data integration [23]. Our work represents a key step forward in understanding the dynamic structure of complex single-cell transcriptomic landscapes.

5 Results for Metric Alignment of Multimodal Data*

In this chapter, we demonstrate the generality and utility of Schema on a range of published datasets. We synthesize RNA-seq and ATAC-seq modalities from multimodal data on 11,296 mouse kidney cells to infer cell types, with Schema enabling an 11% increase in accuracy over previously described approaches. On a dataset of 62,468 spatially-resolved transcriptomes in the mouse cerebellum, we use Schema's feature selection capabilities to identify genes differentially expressed between sparsely and densely packed granule cell neurons. We demonstrate how UMAP and t-SNE visualizations can be made more informative by infusing additional information, like cellular age, into the visualizations. Going beyond gene expression, we perform a feature selection analysis on a dataset of 62,858 T cells to estimate the locations and residues in the T-cell receptor's complementarity-determining region 3 (CDR3) important to its binding specificity. Schema is thus designed to support the continually-expanding breadth of single-cell technologies while retaining the power, tunability and interpretability required for effective exploratory analysis.

5.1 Inferring Cell Types by Synthesizing Gene Expression and Chromatin Accessibility

We first sought to demonstrate the value of Schema by applying it to the increasingly common and broadly interesting setting in which researchers simultaneously profile the transcriptome and chromatin accessibility[1] of single cells [58]. Focusing on cell type inference, a

1: As discussed in Subsection 2.2.1, chromatin accessibility assays measure which sections of chromatin are *physically* accessible in a given cell

* The work in this section is drawn from the *Genome Biology* publication [54], which focuses on results when Schema was applied. I was not the primary author on any of these results; they are included to demonstrate the power of the method.

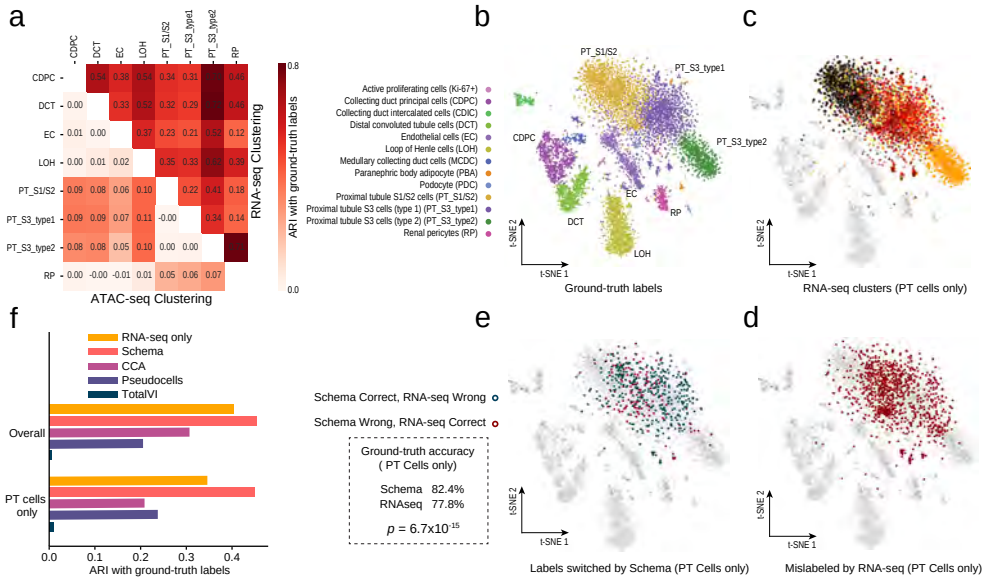

Figure 5.1: Synthesis of RNA-seq and ATAC-seq information leads to more accurate cell type inference. a. Leiden clustering [110] of per-cell transcriptional profiles results in greater agreement (measured as the adjusted Rand index, ARI) with ground-truth cell type labels when featurizing cells by RNA-seq profiles alone compared to featurizing with ATAC-seq profiles alone. ATAC-seq does provide relatively more information when distinguishing PT cells. **b.** Ground truth labels from Cao et al. [58]. **c, d, e.** To assess the ground-truth accuracy of Leiden clustering, we assigned each cluster to the cell type most frequently seen in the ground-truth labels of its members. Clusters where labels are more mixed will thus have lower accuracy. Clustering on RNA-seq profiles alone results in many PT cells assigned to such clusters. Schema synthesis of RNA- and ATAC-seq features, followed by Leiden clustering, results in significantly greater concordance with ground-truth on PT cell types when compared to Leiden clustering on the RNA-seq features alone (One-sided binomial test, $p = 6.7 \times 10^{-15}$). **f.** ARIs of clusters from Schema-synthesized data are higher, especially for PT cells. Synthesizing the modalities using canonical correlation analysis (CCA), totalVI (an autoenconder-based deep learning approach), or a "pseudocell" approach described in the original study results in lower ARI scores.

key analytic step in many single-cell studies[2], we applied Schema on a dataset of 11,296 mouse kidney cells with simultaneously assayed RNA-seq and **ATAC-seq** modalities and found that synthesizing the two modalities produces more accurate results than using either modality in isolation (Figure 5.1F; Figure G.23).

With RNA-seq as the primary (i.e., reference) dataset and ATAC-seq as the secondary, we applied Schema to compute a transformed dataset in which pairwise RNA-seq distances among cells are better aligned with distances in the ATAC-seq peak counts data while retaining a very high correlation with primary RNA-seq distances ($\geq$ 99%)[3]. We then clustered the cells by performing Leiden community detection [110] on the transformed dataset and compared these clustering assignments to the Leiden clusters obtained without Schema transformation. We measured the agreement of these fully-automated clusterings with expertly-defined ground-truth cluster labels[4] (from Cao et al. [58]), quantifying this agreement with the adjusted Rand index (ARI), which has a higher value if there is

2: notwithstanding the discussion in Subsection 2.7.1 about the existence or lack thereof of discrete cell types

ATAC-seq: the methodology for assaying chromatin accessibility, which works by degrading the sections of chromatin in a cell that are accessible

3: See Chapter 3 for a refresher on the method

4: These expertly designed labels are traditionally determined using known marker genes

greater agreement between two sets of labels. Leiden clustering on Schema-transformed data better agrees with the ground truth annotations of cell types (ARI of 0.46) than the corresponding Leiden cluster labels using just RNA-seq or ATAC-seq datasets individually (ARIs of 0.40 and 0.04, respectively, Figure 5.1F). Here, Schema facilitated a biologically informative synthesis despite limitations of data quality or sparsity in the ATAC-seq secondary modality. We observed that using only ATAC-seq data to identify cell types leads to poor concordance with ground-truth labels (Figure G.24a), likely because of the sparsity of this modality (for example, only 0.28% of the peaks were reported to have non-zero counts, on average); this sparsity was also noted by the original study authors.

To further analyze why combining modalities improves cell type clustering, we obtained Leiden cluster labels using either the RNA-seq or the ATAC-seq modalities individually. We then evaluated these cluster assignments by iterating over subsets of the data, each set covering only a pair of ground-truth cell types and used the ARI score to quantify how well the cluster labels distinguished between the two cell types. While RNA-seq clusters have higher ARI scores overall, indicating a greater ability to differentiate cell types, ATAC-seq does display a relative strength in distinguishing proximal tubular (PT) cells from other cell types (Figure 5.1A).

PT cells are crucial to kidney function, with the specific PT cell sub-types playing distinct roles in, for instance, glucose reabsorption [111]. They are also the most numerous cells in this dataset and many of the misclassifications in the RNA-seq based clustering relate to these cells (Figure 5.1B-D). When the two modalities are synthesized with Schema, a significant number of these PT cells are correctly assigned to their ground truth cell types[5], leading to an overall improvement in clustering quality (Figure 5.1E).

Furthermore, upon analyzing Schema's feature-selection output we found that the genes it up-weighted in the primary RNA-seq modality were differentially expressed in PT cells (one-sided t-test, FDR $q < 0.01$ for each of the three PT cell types), thus emphasizing the RNA-seq subspace where support from the secondary modality signal was strongest. These genes[6] are enriched for regulation of macromolecule metabolic process[7] and regulation of nitrogen compound metabolic process[8].

5: significance calculated using the one-sided binomial test, $p = 6.7 \times 10^{-15}$

6: For the biologists in the room, the top hits are *Pnisr*, *Ankrd11*, and *Kmt2c*

7: GO:0060255, FDR $q = 0.0103$

8: GO:0051171, FDR $q = 0.0133$

5.2 Schema's Constrained Data Synthesis Outperforms Unconstrained Approaches

In general, synthesis of multimodal data can also be done by statistical techniques my like canonical correlation analysis (CCA) or deep learning architectures that represent multiple modalities in a shared latent space [77–83]. A key conceptual advance of Schema over these approaches is its emphasis on limiting the distortion[9] of the high-confidence reference modality, allowing it to extract signal from the lower-confidence secondary modalities without overfitting to their noise and artifacts. Intuitively, the synthesis of two modalities requires the identification of a subspace (or latent space) in each modality that aligns well with the other. Due to noise and artifacts, an unconstrained approach may overfit by identifying a pair of subspaces that align well but are biologically uninformative[10]. In contrast, Schema's constrained optimization formulation, combined with the use of a high-confidence modality as the primary, ensures that any possible alignment will use only a biologically-informative subspace of the primary modality and thus guides the quadratic programming optimizer towards correspondingly informative subspaces in the other modalities. To demonstrate the importance of this constrained approach, we evaluated the performance of CCA and totalVI [80] in integrating the RNA-seq and ATAC-seq modalities (Figure 5.1F). We applied CCA to synthesize the two modalities and performed Leiden clustering on the resulting dataset, finding its overlap with the ground truth labels (ARI of 0.31) to be lower than that from Schema's synthesis (0.46). Indeed, this is a lower ARI than is achievable just with RNA-seq data (0.40), indicating that the CCA-based synthesis may be overfitting to the sparse and noisy ATAC-seq data.

To evaluate an **autoencoder**-based synthesis of these modalities, we applied Lopez et al. [77] and totalVI to compute per-modality and dual-modality latent space representations, respectively. We performed Leiden clustering in the autoencoder latent spaces and evaluated the clustering's overlap with ground truth labels. We first verified that the single-modality latent space representations did lead to Leiden clusters of comparable quality[11] as had previously been observed from Leiden clustering on the raw data[12]. However, the dual-modality shared-space representation from totalVI produced a Leiden clustering (Figure G.23b) that had a low overlap with the ground truth[13]. We hypothesize that the sparsity and low signal-to-noise ratio in the ATAC-seq modality led totalVI to a latent space representation that corresponds to low biological-information

9: Recall that we mean keeping the ordering of pairwise distances the same

10: There is a general paradigm in biology that some patterns in a dataset are "biologically relevant" and others are artifacts; it is not obvious how to distinguish them except through expert validation!

autoencoder: briefly, an autoencoder tries to learn an efficient (usually low-dimensional) representation of an unlabeled dataset, with the goal that the data can be recovered from the encoding

11: This was to ensure that the single-modality latent space representations were reasonable in themselves

12: ARIs of 0.365 and 0.038 for scVI-generated representations of RNA-seq and ATAC-seq data, respectively

13: ARI of 0.0043

subspaces of the two modalities, rather than their respective high information subspaces. We note that we were able to achieve better performance with totalVI when applying the same procedure to a synthetic, less-noisy secondary modality consisting of partially-randomized RNA-seq observations.

While these CCA and autoencoder results were likely due to overfitting, the Schema-based synthesis constrains the ATAC-seq modality's influence, enabling us to extract additional signal provided by ATAC-seq while preserving the rich information provided by the transcriptomic modality. We believe that this regularization offered by Schema's constrained optimization formulation is a key advantage that will be crucial in multimodal single-cell data synthesis. We also note that Schema offers additional advantages: unlike CCA, it can incorporate more than two modalities simultaneously and, unlike totalVI, its synthesis is interpretable[14], revealing a more accurate characterization of PT cells.

14: This is often a buzzword in machine learning; here, we mean that Schema tells us exactly which features are used in the synthesis

5.3 Schema Highlights Secondary Patterns While Preserving Primary Structure

Another powerful use of Schema is to infuse information from other modalities into RNA-seq data while limiting the data's distortion so that it remains amenable to a range of standard RNA-seq analyses. Since widely used visualization methods such as UMAP [39] do not allow a researcher to specify aspects of the underlying data that they wish to highlight in the visualization[15], we sought to apply Schema to improve the informativity of single-cell visualizations. We leveraged Schema to highlight the age-related structure in an RNA-seq dataset of *Drosophila melanogaster* neurons [56] profiled across a full lifespan, while still preserving most of the original transcriptomic structure. We chose RNA-seq as the primary modality and temporal metadata (cell age) as the secondary modality, configuring Schema to maximize the correlation between distances in the two while constraining the distortions induced by the transformation. We then visualized the transformed result in two dimensions with UMAP.

15: This is by design, as UMAP and its ilk are unsupervised and *only* use the features to calculate their embeddings

While some age-related structure does exist in the original data, Schema-based transformation of the data more clearly displays a cellular trajectory consistent with biological age (Figure 5.2). Importantly, revealing this age-related structure required only a limited distortion of the data, corresponding to relatively high values[16] of the minimum correlation constraint (Figure 5.2c)[17].

16: ≥ 0.99

17: Essentially, this means that there is plenty of "wiggle room" in the embedding to maintain orderings

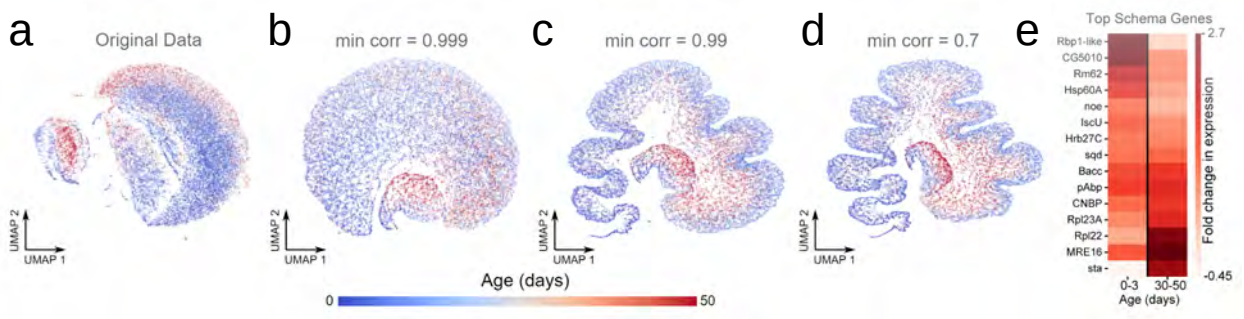

Figure 5.2: Incorporating temporal metadata into UMAP visualizations of aging neurons captures developmental changes. UMAP visualization of RNA-seq profiles of *D. melanogaster* neurons at 0, 1, 3, 6, 9, 15, 30, and 50 days after birth, representing the full range of a typical *D. melanogaster* lifespan. The transcriptomic data (primary modality) was transformed to a limited extent using Schema by correlating it with the temporal metadata (secondary modality) associated with each cell. **a.** UMAP visualization of the original transcriptomic data. **b, c, d.** Visualizations of transformed data with varying levels of distortion. As the value of the minimum correlation constraint s approaches 1, the distortion of the original data is progressively limited. Decreasing s results in a UMAP structure that increasingly reflects an age-related trajectory. **e.** Feature-selection interpretation of Schema's transformation. In synthesizing the two modalities, Schema up-weights genes (top 15 shown here) that are differentially active at the start or end of the time-course. For clarity, the set of genes has been reordered by the difference in their early and late-stage expression.

Analysis of Schema's feature selection indicated an up-weighting of genes differentially expressed at the start or end of the aging process (Figure 5.2E), with genes implicated in cell organization/biogenesis[18] [112] active at the start while ribosomal genes[19] were active at the end. We also confirmed that there was a significant overlap between Schema's highest-ranked genes and those found by a standard differential expression test between time-points[20]. To additionally verify that Schema was infusing additional age-related structure into RNA-seq data, we performed a diffusion pseudotime analysis of the original and transformed datasets and found that the Spearman rank correlation between this **pseudotime** estimate and the ground-truth cell age increased from 0.365 in the original data to 0.405 and 0.436 in the transformations corresponding to minimum correlation constraints of 0.999 and 0.99, respectively[21].

We note that the constrained optimization of Schema was again important to retaining biological signal during the synthesis: in comparison, an unconstrained synthesis by CCA led to a lower pseudotime correlation[22] than seen in the original RNA-seq dataset; the corresponding CCA-based UMAP visualization was also less clear in conveying the cellular trajectory (Figure G.26). Schema thus enables visualizations that synthesize biological metadata, while preserving much of the distance-related correlation structure of the original primary dataset[23]. With Schema, researchers can therefore investigate single-cell datasets that exhibit strong latent structure[24], infusing this secondary information into the primary RNA-seq modality. We recommend specifying a high minimum-correlation

18: for example, *Rm62*, *CG5010* and *IscU*

19: for example, *Rpl22* and *Rpl23A*

20: one-sided binomial test, FDR $q < 10^{-21}$ for the 1, 30 and 50-minute subsets

pseudotime: a computational estimate of the age of a cell given it scRNA-seq vector

21: In other words, the Spearman correlation indicates that the estimated pseudotime in the *new* embedding corresponds better to actual cell age

22: $R^2 = 0.059$

23: There is of course, the reasonable question about whether imbuing the visualization with additional side-information *reduces* its quality (since you will see what you are expecting); this depends on how much you trust the additional information to be biologically meaningful; see Chari, Banerjee, and Pachter [51] for a discussion on these semi-supervised embeddings

24: This could, for example, be due to metadata like age or spatial location

constraint (e.g., 0.99) during the synthesis, having observed that only a small transformation of the RNA-seq data is (generally) needed to make the latent structure visible[25].

25: One reason why the constraint can be so high is in some sense the converse of the JL-limits from Theorem 2.6.1 — in high dimensions, there is much more flexibility for points to move around while not changing distances between them

5.4 Spatial Density-informed Differential Expression Among Cerebellar Granule Cells

In addition to cell type inference, another important single-cell analysis task that stands to benefit from multimodal synthesis is the identification of differentially expressed marker genes[26]. To perform differential expression analysis with Schema, RNA-seq data should be used as the primary modality, while the distance metrics of the secondary modalities specify how cells should be differentiated from each other. We applied Schema to spatial transcriptomics data, another increasingly important multimodal scenario, here encompassing gene expression, cell-type labels, and spatial location.

26: See Subsection 2.2.2 for the discussion about the difference between these problems

We obtained Slide-seq data containing 62,468 transcriptomes that are spatially located in the mouse cerebellum. In the original study, these transcriptomes were assigned to putative cell types[27], and thus cell types are located throughout the tissue [62, 113]. Interestingly, we observed spatial density[28] variation for certain cell types; specifically, transcriptomes corresponding to granule cell types are observed in regions of *both* high and low spatial density (Figure 5.3B).

27: Noting that these transcriptomes are not guaranteed to be single-cell

28: It's all about density!

Schema's feature-selection capabilities could thus identify genes that are differentially expressed in granule cells in high density areas versus granule cells in low density areas. Schema is well suited to the constrained optimization setting of this problem: we optimize for genes expressed specifically in granule cells and in dense regions, but not all granule cells are in dense regions and not all cells in dense regions are granule cells. We specified RNA-seq data as the primary modality and spatial location and cell-type labels as the secondary modalities[29]. In the spatial location modality, the distance metric was defined such that two cells are similar if their spatial neighborhoods have similar density.

29: "Distance" in the categorical modality is merely the "one-hot" distance — two cells have a distance of 1 if they are different cell-types and 0 if they are the same cell-types

The densely-packed granule cell genes identified by Schema are strongly enriched for GO terms and REACTOME pathways [114] related to signal transmission[30]. This finding suggests potentially greater neurotransmission activity within these cells (Figures G.29 and G.30, Appendix F).

30: for example, ion-channel transport (REACTOME FDR $q = 1.82 \times 10^{-3}$), ion transport (GO:0022853, FDR $q = 1.8 \times 10^{-17}$), and electron transfer (GO:009055, FDR $q = 2.87 \times 10^{-11}$)

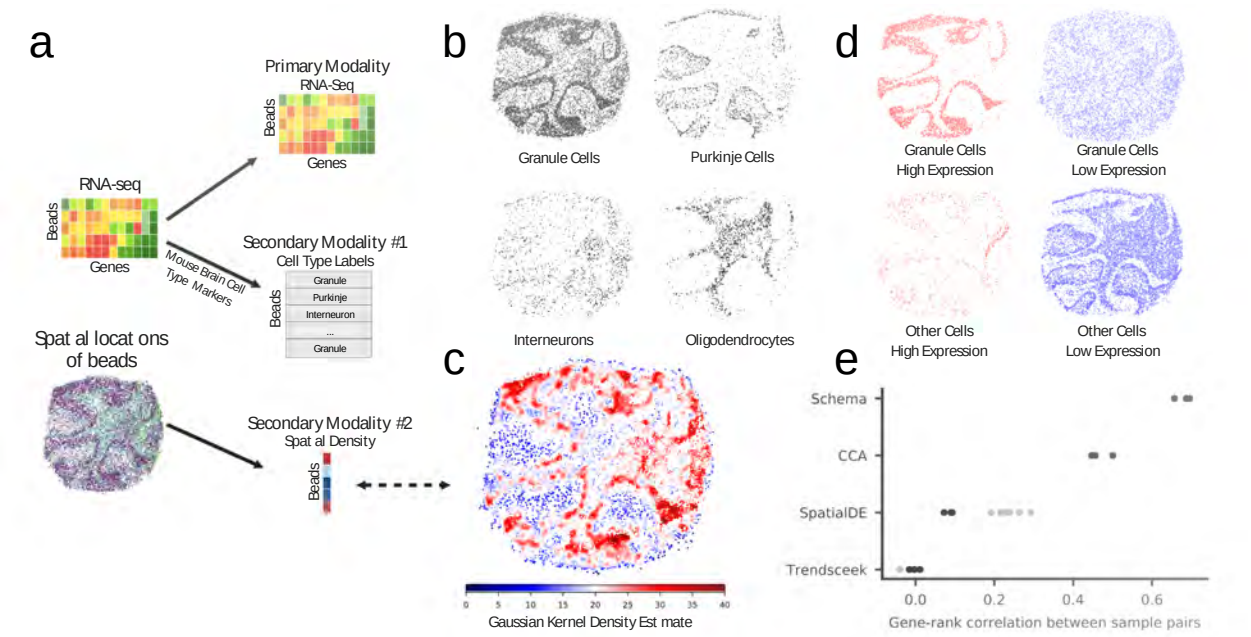

Figure 5.3: Schema identifies a gene set in granule neurons whose expression covaries with spatial cellular density. a. Rodriques et al. [62] simultaneously assayed spatial and transcriptomic modalities in mouse cerebellum tissue (data from puck 180430_1 is shown here). In addition, they labeled beads (each corresponding to a transcriptome) with a putative cell-type by comparing gene expression profiles with known cell-type markers. **b.** Spatial distribution of the most common cell types in the tissue: granule cells, Purkinje cells, interneurons, and oligodendrocytes. Note the variation in spatial density for granule cells. **c.** We quantified this spatial density variation by computing a two-dimensional Gaussian-kernel density estimate, with cells in dense regions assigned a higher score. **d.** Schema is able to identify a set of genes that are highly expressed only in densely-packed granule cells. The four figures here show mutually disjoint sets of cells: granule cells with high expression of the gene set, granule cells with low expression of the gene set, other cells with high expression, and other cells with low expression. Here, a cell is said to have high expression of the gene set if the cell's loading on this gene set ranks in the top quartile. **e.** Schema's results are robust across biological replicates. Across three replicates, we evaluated the consistency of gene rankings computed by Schema, canonical correlation analysis (CCA), SpatialDE and Trendsceek. The black points indicate the Spearman rank correlation of gene scores across pairs of replicates. We needed to adapt SpatialDE and Trendsceek for this task by first applying them separately on granule and non-granule cells and then combining the results; here, the black and grey points indicate the cross-replicate correlations of the final and intermediate gene-rankings, respectively.

5.5 Schema Outperforms Alternative Methods for Spatial Transcriptomic Analysis

We sought to benchmark our method by comparing the robustness of Schema's results with those based on canonical correlation analysis (CCA) and with two methods specifically intended for spatial transcriptomics[31], namely SpatialDE [68] and Trendsceek [69].

An important point is that CCA, SpatialDE, and Trendsceek are less general[32] than Schema and therefore require non-trivial modifications to approximately match Schema's capabilities. CCA is limited in that it can correlate only two datasets at a time, whereas here we seek to synthesize three modalities: gene expression, cell-type labels, and spatial density. We adapted CCA by correlating two modalities at a time and combining the sub-results. In the case of SpatialDE and Trendsceek, their unsupervised formulation does not allow the researcher to specify the spatial features to pick out[33]. To adapt these, we collated their results from separate runs on granule and non-granule cells. Notably, the *ad hoc* modifications required to extend existing methods beyond two modalities underscore the benefit of Schema's general analytic formulation that can be *naturally* extended[34] to incorporate any number of additional data modalities.

Reasoning that a robust computational approach should return consistent results across biological replicates[35], we evaluated the stability and quality of each spatial transcriptomic technique by comparing its results on three replicate samples of mouse cerebellum tissue[36]. While both Schema and CCA identify a gene set that ostensibly corresponds to granule cells in dense regions (Figure 5.3D; Figure G.24), the gene rankings computed by Schema are more consistently preserved *between* pairs of replicates than those computed by CCA, with the median Spearman rank correlation between sample pairs being 0.68 (Schema) versus 0.46 (CCA). Likewise, with Schema, 69.1% of enriched GO biological-process terms are observed in all three samples and 78% are in at least two samples. The corresponding numbers for CCA were 35.7% and 59.5%, respectively [37]. We thus find that Schema's results are substantially more robust across the three replicates. Compared to CCA's unconstrained synthesis, Schema's constrained formulation avoids overfitting to sample-specific noise, enhancing its robustness (Figure 5.3e; Figure G.25).

When performing the same gene list robustness analysis with SpatialDE and Trendsceek, while also looking at the stability of their gene rankings specific to the precursor cell type[38], we found that Spa-

31: We note that Schema is *not* explicitly designed for this type of analysis

32: This is meant in the sense that these spatial methods *need* one of the modalities to be spatial

33: We focus on spatial density variation

34: We do not want to overstate the case here: even for Schema, one must cleverly design the weights and distances, but the formulation is the same

35: This is an example of the difference between *biological* variation and external variation — consistency across replicates means that the biological signal is picked up

36: for example, coronal sections prepared on the same day [62]; pucks 180430_1, 180430_5, 180430_6

37: FDR $q < 0.001$ in all cases

38: the gray points in Figure 5.3e

tialDE produced slightly more stable gene rankings than Trendsceek, with median sample-pair correlations of 0.089 and -0.002, respectively, but these were still much lower than those for Schema. We also observed that SpatialDE and Trendsceek had substantially longer running times and we performed our analysis of the two methods on subsets of the overall dataset (see Section 5.8 for precise runtime and memory usage). These results demonstrate the robustness and efficiency of Schema's supervised approach.

5.6 Beyond Gene Expression: Schema Reveals CDR3 Segments Crucial to T-cell Receptor Binding Specificity

To further demonstrate the generality of Schema, we applied it to synthesize data modalities beyond gene expression. We integrated single-cell multimodal proteomic and functional data with Schema to better understand how sequence diversity in the hypervariable CDR3 segments of T-cell receptors (TCRs) relates to antigen binding specificities [115]. *De novo* design of TCRs for an antigen of interest remains a pressing biological and therapeutic goal [116, 117], making it valuable to identify the key sequence locations and amino acids that govern the binding characteristics of a CDR3 segment. Towards this end, we analyzed a single-cell dataset that recorded clonotype data for 62,858 T-cells and their binding specificities against a panel of forty four ligands [15] and used Schema's feature-selection capabilities to estimate the sequence locations and residues in the CDR3 segments of α and β chains important to binding specificity[39].

39: The underlying assumption is that residues present in the T-cells that bind to *many* residues are important for binding specificity

To estimate location-specific selection pressure, we ran Schema with the CDR3 peptide sequence data as the primary modality and the binding specificity information as the secondary modality, performing separate runs for α and β chains. In the primary modality, each feature corresponds to a CDR3 sequence location and we used the **Hamming distance** metric between observations. In the secondary modality, for T-cell i, coordinate j for the feature vector x_i indicates the binding strength between cell i and ligand j. Aligning the modalities well would emphasize the residues that are preserved *across* the T-cells that bind strongly to similar ligands. Schema assigned relatively low feature weights to the location segments 3-9 (in α chain CDR3) and 5-12 (in β chain CDR3), suggesting those regions can tolerate greater sequence variability while preserving binding specificity.

Hamming distance: between two sequences, this is the number of positions in which they differ

To evaluate these results, we compared them to estimates based on CDR3 sequence motifs sourced from VDJdb [118], a curated database of TCRs with known antigen specificities. In VDJdb, TCR motifs are scored using an adaptation of the relative-entropy algorithm by Murugan et al. [119] that assigns a score for each location and amino acid in the motif. We aggregated these scores into a per-location score, allowing a comparison with Schema's feature weights (Figure 5.4). While the comparison at locations 11-20 is somewhat complicated by VDJdb having fewer long sequences, there is agreement between Schema and VDJdb estimates on locations 1-10 where both datasets have good coverage[40]. We note that weight estimation using Schema required only a single multimodal dataset; in contrast, extensive data collection, curation, and algorithmic efforts underlie the VDJdb annotations[41]. The latter covers multiple experimental datasets, including the 10x Genomics dataset [15] we investigated here; we saw similar results when comparing against an older version of VDJdb without this dataset.

40: Spearman rank correlations of 0.38 and 0.92 for the α and β chains, respectively; Figure 5.4c–d

41: This is why VDJdb is used as ground truth for validation

Next, we used Schema to investigate the selection pressure on amino acids present in the variability-prone locations identified above. We first selected a sequence location[42] and constructed a primary modality where each cell was represented by a **one-hot encoding** of the amino acid at the location[43]. The secondary modality was binding specificity information, as before. We performed separate Schema runs for each such location of interest on the two chains, computing the final score for each amino acid as the average score across these runs. These scores are in good agreement with the corresponding amino acid scores aggregated from the VDJdb database[44]. The residue and location preferences estimated here can directly be used in any algorithm for computational design of epitope-specific CDR3 sequences to bias its search towards more functionally plausible candidate sequences.

42: for example, location 4 in α chain CDR3

one-hot encoding: a method of representing a sequence as a vector; given an alphabet with characters $\{\gamma_1, \ldots, \gamma_r\}$, each element v_i of the sequence is a Boolean vector of size r, with $v_i^{(j)} = 1$ if the ith character is γ_j; all other values are zero

43: so in this case, a 20-dimensional Boolean vector

44: Spearman rank correlation $\rho = 0.74$, two-sided t-test $p = 2 \times 10^{-4}$

Schema's ability to efficiently synthesize arbitrarily many modalities, with their relative importance at the researcher's discretion, allows information that might otherwise be set aside[45] to be effectively incorporated, enhancing the robustness and accuracy of an analysis. We exemplify this use-case on the TCR dataset by incorporating measurements of cell-surface markers as an additional secondary modality, hypothesizing that cell-surface protein levels should be unrelated to V(D)J recombination variability.

45: for example, metadata like batch information, cell line, or donor information

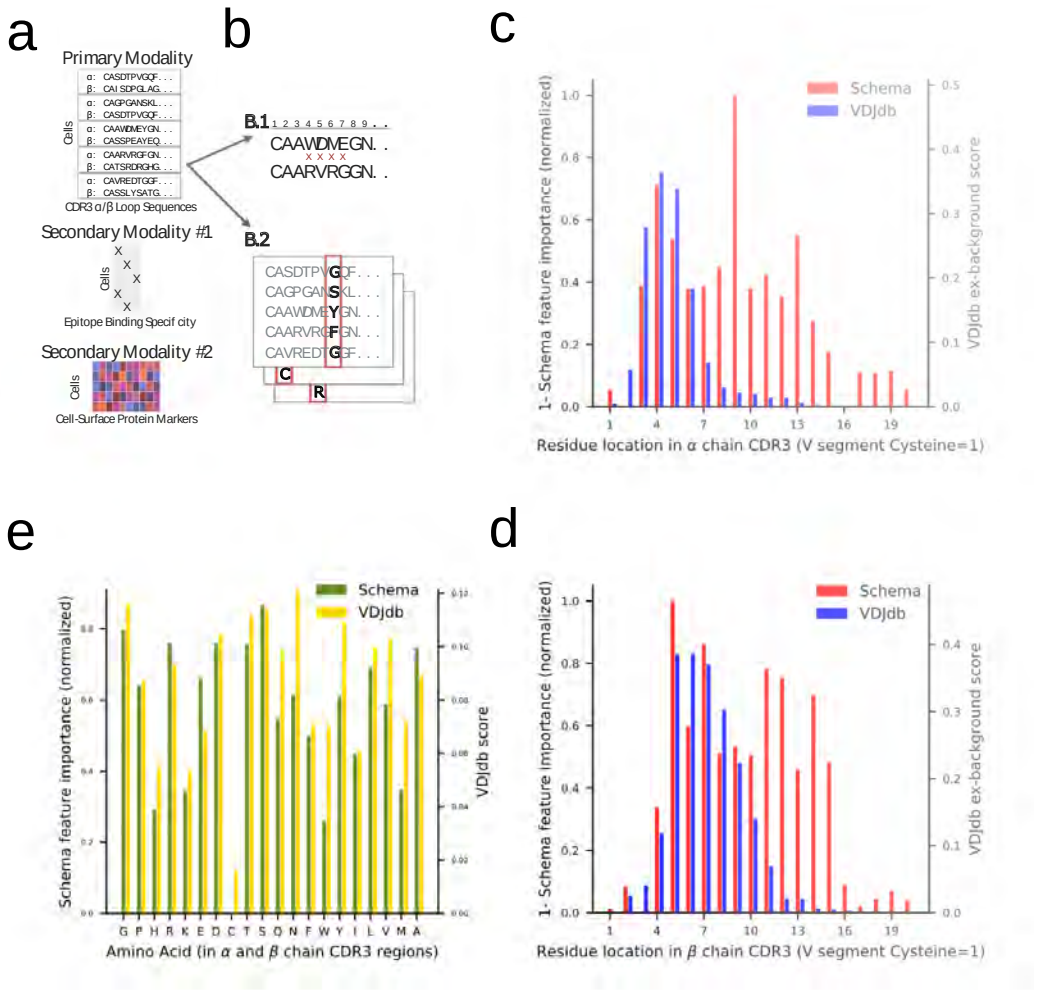

Figure 5.4: Schema reveals the locations and amino acids important in preserving binding specificity of T-Cell receptor CDR3 regions. a. We analyzed a multimodal dataset from 10x Genomics [15] to understand how a T-cell receptor's binding specificity relates to the sequence variability in the CDR3 regions of its α and β chains. The primary modality consisted of CDR3 peptide sequence data which we correlated with the secondary modality, the binding specificity of the cell against a panel of 44 epitopes. We optionally synthesized an additional modality, proteomic measurements of 12 cell-surface marker proteins, as a use-case of incorporating additional information. **b.** We performed two Schema analyses: (**b.1**) to infer location-wise selection pressure, each feature of the primary modality corresponded to a location in CDR3 sequence and (**b.2**) to infer amino-acid selection pressure, the primary modality was the Boolean vector of residues observed at a specific sequence location; we aggregated over an ensemble of Schema runs across various locations. **c, d.** Schema identifies sequence locations 3–9 (α chain) and 5–12 (β chain) as regions where sequences can vary with a comparatively modest impact on binding specificity. We compared Schema's scores to statistics computed from motifs in VDJdb. Here, we have inverted the orientation of Schema's weights to align them with the direction of VDJdb weights. **e.** Schema and VDJdb agree on the relative importance of amino acids in preserving binding specificity (Spearman rank correlation of 0.74, two-sided t-test $p = 2 \times 10^{-4}$). The low weight assigned to cysteine is likely due to its infrequent occurrence in the data.

5.7 Additional Demonstrations

Applying Schema on a mouse gastrulation dataset [120] consisting of 16,152 epiblast cells split over three developmental timepoints and with two replicates at each timepoint, we performed differential expression analysis[46] while simultaneously accounting for batch effects and developmental age, and evaluated its results alongside those from MOFA+, a recently introduced single-cell multimodal analysis technique [72] (Figure G.21; Appendix D). We also used Schema to study cell differentiation by synthesizing spliced and unspliced mRNA counts in a dataset of 2,930 mouse dentate gyrus cells[47] [120]. As in standard RNA velocity analyses, correlating spliced and unspliced counts in a cell picks up on the time derivative of a cell's expression state and thus illuminates the cell differentiation process. Schema's results agree with those from the dedicated RNA velocity tool scVelo [58], and we also demonstrate how Schema can be used to infuse velocity information into a t-SNE visualization (Figure G.22; Appendix E).

46: Recalling Chapter 3 Schema's feature selection can be interpreted as differential expression analysis

47: This is a type of analysis called *RNA velocity analysis* — since unpsliced mRNAs *become* spliced RNAs (see Subsection 2.1.4), the ratio of the two indicates how quickly the RNA is being processed

5.8 Schema Can Scale to Massive Single-cell Datasets

We have designed Schema to process large single-cell datasets efficiently, with modest memory requirements. On average, Schema processes data from a Slide-seq replicate[48] in 34 minutes, requiring less than 5GB of RAM in the process (Table H.1). The runtime includes the entire set of Schema sub-runs performed over an ensemble of parameters, as well as the time taken for the pre-processing transformation.

Schema's efficiency stems from our novel mathematical formulation. Deviating from standard metric learning approaches, we formulate the synthesis problem as a quadratic-program optimization, which can be solved much faster than the semi-definite program formulations typically seen in these approaches (Section 3.5). Additionally, while the full Schema algorithm has scales quadratically[49] in the number of cells, our formulation allows us to obtain good approximations with provably bounded error using only a logarithmic subsample of the dataset (Appendix A), enabling *sublinear* scalability in the number of cells that will be crucial as multimodal datasets increase in size. These subsampling techniques can also leverage diversity-preserving data sketching techniques [44, 46] that may

48: three modalities, 20,823 transcriptomes $\times$17,607 genes

49: The computer scientists will grimace at this!

empirically lead to greater representation of rare cell types in the Schema analysis.

6 Results for Density-preserving Visualization*

The utility and importance of methods focused on visualization is not entirely obvious, despite their ubiquity. After all, one might expect that a "good" scientist would *not* draw conclusions based on just a scatterplot reading of their complex, high-dimensional data. In this chapter, we focus on the *applications* of the densMAP and den-SNE algorithms introduced previously, highlighting both the importance of good visualization methods and the shortcomings of existing methods. We focus on a diverse array of published datasets, from lung cancer patients [17], human peripheral blood cells [121] and embryonic roundworm *Caenorhabditis elegans* [16], as well as the UK Biobank human genotype profiles [122] and the canonical MNIST hand-written digit images. These methods not only capture additional information beyond existing visualization techniques but also biological insights others miss, including immune cell transcriptomic variability in tumors; specialization of monocytes and dendritic cells; and temporally modulated transcriptomic variability across developmental lineages of *C. elegans*.

6.1 Visualizing the Heterogeneity of Immune Cells in Tumor

To illustrate the value of density-preserving visualization for biological studies, we first applied our methods to a scRNA-seq dataset of 41,861 immune cells in matched tumor and peripheral blood samples from seven non-small-cell lung cancer (NSCLC) patients [17]. The original study identified distinct transcriptomic states spanned

* The text in this section is also from the publication 'Assessing single-cell transcriptomic variability through density-preserving data visualization' by Narayan, Berger, and Cho [91]; it focuses on the **Results** sections of that work

by tumor-infiltrating myeloid cells that were reproducibly observed across different individuals, suggesting their potential relevance for cancer immunotherapies[1]. We asked whether our methods could more accurately capture the transcriptomic landscape of tumor-infiltrating immune cells than existing tools.

1: In other words, these immune cells are worth focusing on precisely because they are active in cancerous conditions

Comparison of den-SNE and t-SNE embeddings revealed several immune cell types with noticeable differences between the visualizations (Figure 6.1): tumor-infiltrating neutrophils and plasma cells occupy considerably more space in the den-SNE visualization than their t-SNE counterparts, while tumor-infiltrating T cells are relatively smaller in den-SNE. These discrepancies arise because visual size of a cluster in t-SNE corresponds more closely to the number of cells in the cluster than to underlying variability[2]. Thus, in t-SNE, tumor-infiltrating neutrophils occupy much less space than circulating neutrophils[3] despite den-SNE indicating they have comparable variability. The rich transcriptomic diversity of tumor-infiltrating plasma cells is also lost in t-SNE. Conversely, T cells, the most populous cell type in tumors[4] are visually overrepresented in t-SNE relative to their actual variability.

2: This is a function of the fact that t-SNE assumes uniform density of all clusters

3: $n = 2861$ for tumor-infiltrating and $n = 9217$ for circulating

4: $n = 10701$

To quantify the improvement in density preservation that our algorithms offer, we calculate two complementary measures of local density in the visualization — (i) local radius and (ii) neighborhood count (as defined in Subsection 4.4.1) — and assess their correlation with the local radii in the original data space, which represent underlying variability in the dataset. Both measures quantify our perception of density in the visualizations[5]; intuitively, the local radius captures the size of a neighborhood that contains a fixed number of nearest neighbors, and the neighborhood count captures the number of points within a fixed radius around each point. The former is consistent with how our algorithms model density for efficient optimization, while the latter is arguably a more direct notion of density previously used in the literature on visual perception [99].

5: Note that it is inversely-related for local radius

The accuracy of den-SNE's visualization of local density is confirmed by the high correlation based on both measures[6] compared to t-SNE [7] (Figure 6.1c; Figure G.1). Results with densMAP [8] and UMAP [9] are analogous (Figures G.1 and G.2). Different parameter choices for UMAP and t-SNE did not improve their density-preservation performance[10], as is expected based on our theoretical analysis in Section 4.5. We also observed that even on previously proposed metrics of visualization quality based on clustering accuracy and pairwise distance preservation [52], our density-preserving tools largely preserve or improve upon the performance of the original

6: $R^2 = 0.650$ for local radius; average $R^2 = 0.657$ for neighborhood count across different length-scales

7: $R^2 = 0.004$; $R^2 = 0.023$

8: $R^2 = 0.590$; $R^2 = 0.632$

9: $R^2 = 0.045$; $R^2 = 0.008$

10: $R^2 < 0.05$ in all cases; Figure G.3

methods (Appendix B; Figures G.4 to G.8). Traditional dimensionality reduction approaches, including principal component analysis [123] (PCA), multidimensional scaling [124] (MDS), and Isomap [125], were ineffective *both* at preserving density and at visualizing clustering structure[11] (Figure G.9). Our improved visualizations of simulated datasets in Figure 4.2 are similarly supported by our quantitative measures (Figure G.10).

Our visualizations motivate *transcriptomic variability* as a key distinguishing factor among cell types and biological conditions. To illustrate, we examined tumor-infiltrating lymphocytes (TILs) compared to those in blood. While essential in the anti-tumor immune response [126], these cells' molecular mechanisms in cancer remain poorly understood. Density-preserving visualization newly highlighted the increased transcriptomic variability of T and B cells compared to their counterparts in blood (Figure 6.1d). Despite an apparent size-difference between the tumor and blood TILs in t-SNE, lack of density-preservation means this pattern could only imply a difference in cell counts, not in variability of expression[12].

Ranking genes by their contribution to the increase in transcriptomic variability[13] in tumor implicated several biological processes as potential driving factors of TIL diversity (Subsection 6.1.1; Tables H.2 to H.11. Top genes for CD8 T cells and CD4 memory T cells were significantly enriched in negative regulation of IL2 production, transcription, and metabolic processes, suggesting that T cells in tumor are subjected to variable degrees of proliferation control, likely in response to biochemical signals in the tumor microenvironments (Tables H.7 and H.8). Notably, RGS1 and DUSP4 showed the largest difference in variability for both T cell types[14]. We validated[15] the variability difference of these two genes in CD8 T cells between tumor and blood based on another scRNA-seq dataset of TILs from NSCLC patients [129], along with seven other genes in our list of genes ranked by contribution to variability[16]. On the other hand, top genes for naïve CD4 T cells are enriched in proteins targeting membranes and in those that ensure the decay of mis-transcribed mRNA (Table H.9). For B cells, key biological processes underlying the variability difference included leukocyte activation and protein complex assembly for memory B cells, and response to cyclic AMP[17] and biotic stimulus for naïve B cells, along with transcriptional and metabolic regulation processes similar to those implicated for T cells (Tables H.10 and H.11).

While many genes implicated here are lowly-expressed in blood and activated in tumor, we also found a substantial portion[18] that show statistically significant **overdispersion** in tumor, whereby

11: This in itself is quite an interesting occurrence, considering that the algorithms mentioned are both well-motivated and well-used; we contend it is because they are not designed for the *drastic* reduction down to two dimensions that we require for visualization

12: In other words, we can now make conclusions about variability based on examining the *visualizations alone* now, whereas this was impossible with the original algorithms

13: We discuss this in more detail, but essentially we are interested in the *variance* of these genes within a cell-type

14: RGS1 encodes a regulator for the G-protein signaling pathway known to be involved in chemokine-induced lymphocyte migration [127], and DUSP4 encodes a phosphatase that modulates a T cell receptor signaling pathway with known association with immunological disorders [128]

15: This is a loaded word, but essentially, we were able to replicate our conclusions about genes that drive variability

16: Nine out of nineteen genes were found to have significant increase in variance in tumor in the validation dataset; Table H.12)

17: This is a known modulator of cell proliferation

18: Specifically, 42% among top twenty genes across all cell types

overdispersion: a random variable is overdispersed, if its variance increases super-linearly as its mean; this is motivated by the fact that the *Poisson distribution*, common for count variables, has its variance equal to its mean

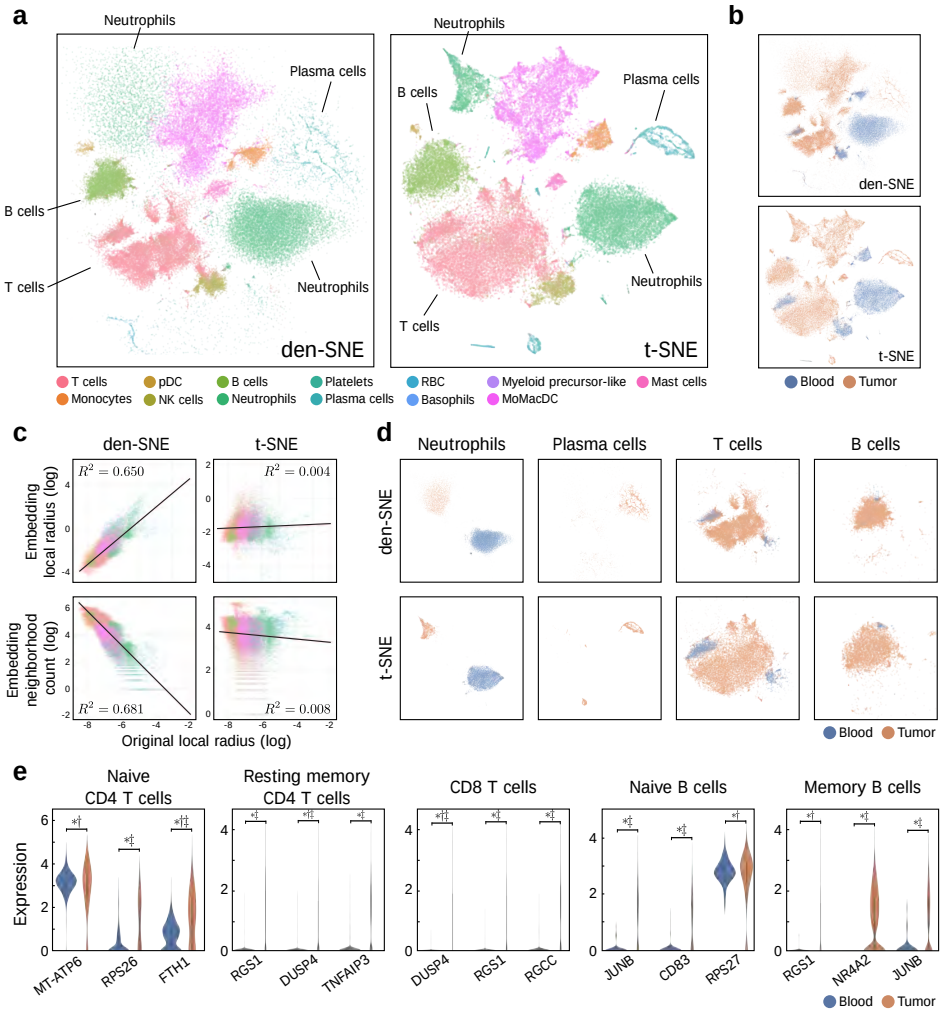

Figure 6.1: Density-preserving visualization reveals heterogeneity in transcriptomic variability of immune cells in blood and tumor. Dataset of tumor and blood immune cells from lung cancer patients [17] using den-SNE and t-SNE, colored by (**a**) cell type and (**b**) tissue type (tumor or blood); den-SNE exposes striking density differences between immune cell types and between blood and tumor, not seen in t-SNE. Note that the relative heterogeneity of neutrophils, plasma cells, and T cells sizes are misleading in t-SNE. **c.** Scatter plots comparing the local radii in the original space to local radius and neighborhood count in the den-SNE embedding. Higher correlations of den-SNE show that den-SNE more accurately conveys the density landscape. **d.** The same visualizations for den-SNE (top) and t-SNE (bottom), restricted to each of four notable cell types (neutrophils, plasma cells, T cells, and B cells) and colored by tissue type (tumor or blood). Neutrophils and plasma cells in tumor considerably expand in size in den-SNE, reflecting transcriptomic variability. T and B cells show a large increase in heterogeneity in tumor compared to blood in den-SNE. Although t-SNE shows a similar pattern, its lack of a density-preservation property precludes reasoning about differences in heterogeneity. **e.** Violin plots showing the distributions of gene expression in tumor and blood for the top three genes with the highest increase in variance in tumor for each subtype of T and B cell (see Appendix H for the full list of genes). These genes indicate potential biological mechanisms underlying the increased heterogeneity of T and B cells in tumor. ($*$, †, and ‡ denote statistically-significant difference in variance, dispersion, and mean, respectively, between blood and tumor, Bonferroni-corrected $p < 0.01$).

the increase in variance cannot be explained by an increase in mean expression (Subsection 6.1.1, Tables H.2 to H.6). In fact, some genes[19], show a significant increase in variance *without* a significant change in mean. These genes are especially common in the top genes for naïve CD4 T cells. Their stability in mean expression implies that these key distinguishing genes *cannot* be identified by conventional differential expression analysis[20]. Moreover, since standard visualization algorithms separate clusters largely based on difference in mean expression, the effects of these genes are lost in their visualizations. Our findings demonstrate that the transcriptomic variability landscape uncovered by our visualizations helps open new analytic directions for the study of anti-tumor immune response.

19: for example, RPS27 in naïve B cells, which encodes the MPS-1 protein that modulates the activity of tumor-suppressor p53 [130]

20: because these analyses focus on differences in mean expression *only*

6.1.1 Differential analysis of gene expression variability in the lung cancer data.

For each cell type with visible expansion of transcriptomic variability in tumor in our visualizations — CD8 T cells[21], CD4 memory resting T cells[22], CD4 naïve T cells[23], memory B cells[24], and naïve B cells[25] — we identified the twenty genes with the largest increase in variance in tumor compared to blood[26] for further analysis. For each gene and cell type, we calculated the differences in the mean and variance of expression between tumor and blood. The statistical significance of the observed differences is assessed using a **permutation test**.

21: 1,621 cells in blood, 443 cells in tumor

22: 1,036 cells in blood, 9,019 cells in tumor

23: 437 cells in blood, 61 cells in tumor

24: 67 cells in blood, 4,811 cells in tumor

25: 83 cells in blood, 396 cells in tumor

26: To be precise, this is the variance of that gene's expression within cells of the given cell-type

Permutation testing

Permutation testing allows you to perform statistical significance testing even when you do not know the distribution your data come from, by randomly permuting the connection between features and label to generate the null distribution.

In this case, the assignment of cells to tumor or blood is randomly permuted, and the statistic computed on the permuted dataset is viewed as samples from the *null* distribution where there is no difference between tumor and blood.

- **Test for variance:** For comparing the variance, we centered the expression levels for each group (tumor or blood) before the permutation procedure to control for the shift in mean. The p-value is calculated as the fraction of permutations that result in a statistic whose magnitude is larger than the statistic computed on the original dataset. We used 100k permutations to estimate the p-values and applied

Bonferroni correction within each cell type to account for multiple hypothesis testing[27].

When considering changes in the variance of gene expression, it is important to note that an increase in variance can often be explained by an increase in mean[28]. Thus, we additionally calculated the difference in **dispersion index** (DI) to assess the extent to which the change in variance is unexplained by a corresponding change in mean.

- **Test for dispersion:** We assessed the statistical significance of the difference in DI also using a permutation test. For the null distribution, we assume that in the absence of excess difference in dispersion, the variance of expression has a linear dependence[29] on the mean (as suggested by the dispersion index). A permutation scheme that correctly reflects this null distribution is one where the expression levels within each group (tumor or blood) are transformed as $x \mapsto \mu^{-1/2}(x - \mu) + 1$ before the permutation, where μ is the sample mean of the group. This transformation maps both groups to the same mean ($\mu = 1$) while preserving the DI[30], so that permuting the labels leads to a valid sample from the null distribution. Similar to the mean and variance tests, we used 100k permutations to estimate the p-values and applied Bonferroni correction.

27: Briefly, when k hypotheses are considered, the probability that any one incorrectly appears significant based on a given p-value is around kp, so Bonferroni correction uses p/k as the significance level for each individual test

dispersion index: given by σ^2/μ, where μ and σ^2 are the mean and variance of expression

28: For example, under the Poisson process model of underlying count distributions, variance of the observed counts naturally scales with the mean [106]

29: So $\sigma^2 = \alpha\mu + \beta$ for some fixed α and β

30: That is, the sample variance is equal to the dispersion index

6.2 Visualizing Immune Cell Specialization and Diversification in Peripheral Blood

While the above illustrates changing patterns of variability that come about due to disease, we show here that variability of expression *within* cellular subtypes also reveals interesting underlying biology[31]. We used densMAP to visualize a benchmark scRNA-seq experiment that profiled 68,551 peripheral blood mononuclear cells (PBMC) from 10X Genomics [15, 121]. While both UMAP and densMAP separate the various clusters corresponding to different cell types, the densMAP embedding considerably expands the sizes of natural killer (NK) cells, cytotoxic T cells, CD14+ monocytes and dendritic cells (DCs), and shrinks naïve cytotoxic T cells (Figure 6.2a). Similar to the cancer dataset, the sizes of these clusters in UMAP correspond to the number of cells belonging to them and do not accurately reflect their variability of expression. By quantifying the agreement between the local radius in the original dataset and the local density measures in each visualization, we confirmed that densMAP more

31: In fact, this is *precisely* what raises questions about the utility of discrete cellular subtypes

accurately preserves density[32], compared to UMAP[33] (Figure 6.2c; Figure G.11). The same pattern is observed when comparing den-SNE to t-SNE, with the density correlations in den-SNE much higher[34] than in t-SNE[35] (Figures G.11 and G.12).

We focus here on the monocyte and DC clusters, which are strikingly different between the two visualizations (Figure 6.2b). While both reveal two subtypes[36] of monocytes, densMAP separates them by density, with a dense subcluster adjacent to a much sparser one. Clustering these cells in the original gene expression space indeed identifies the two subtypes as separate clusters[37] (Figure G.13). These cells begin life as *classical* monocytes, characterized by expression of the gene CD14 and a lack of CD16 (also called FCGR3A); these can then differentiate into CD16 monocytes, macrophages, or dendritic cells (DCs) [132] (Figure 6.2d). Marker gene expression associated[38] the sparse cluster with classical monocytes and the dense cluster with CD16 monocytes (Figure 6.2f), suggesting that classical monocytes exhibit a high level of variability[39] before developing into more homogeneous CD16 monocytes. This trajectory has intriguing biological significance. Recent work has revealed that monocytes are an extremely heterogeneous cell type with complex intermediate states [133] and high transcriptional diversity [134]. However, non-classical monocytes are more specialized: they are thought to emerge from a small population of intermediate (CD14+CD16+) monocytes and spike rapidly during infections [133]; since their progenitor cell is rare and accounts for a small portion of transcriptional diversity represented by CD14 monocytes (Figure 6.2f), this supports the notion of a bottleneck in the development of non-classical monocytes[40].

We validated this difference in variability between classical and non-classical monocytes in two other scRNA-seq datasets of immune cells, one that profiled 1,078 monocytes, DCs and their subtypes [131] (PBMC2) and the other that profiled 13k PBMCs from two healthy donors [135] (PBMC3). In both, classical monocytes were sparser than non-classical ones: classical monocytes had larger local radii in the gene expression space than non-classical monocytes[41].

A similar analysis can be performed on the DC subset: this cell type shows (i) a dense cluster of cells adjacent to the CD14 monocytes, (ii) a dense cluster overlapping the CD16 monocytes, and (iii) a sparser cluster near the CD14 monocytes (Figure 6.2b). While the classification of dendritic cells[42] is still actively researched [136], the colocalization of the DCs (i) and (ii) and the monocytes in the densMAP visualization suggests[43] that these DCs originate from monocytes. By analyzing the expression of the marker genes of DC subtypes identified by the the PBMC2 study [131] in these DC subsets,

32: $R^2 = 0.712$ for local radius; average $R^2 = 0.727$ for neighborhood count

33: $R^2 = 0.000$; $R^2 = 0.000$

34: $R^2 = 0.704$; $R^2 = 0.696$

35: $R^2 = 0.052$; $R^2 = 0.037$

36: We do not attempt some rigorous definition for when subtypes exist, rather just appealing to visual judgment, in the tradition of Justice Potter Stewart

37: While clustering algorithms are notoriously dependent on parameter selection, these clusterings were generally robust

38: In other words, CD14 was found in the sparse cluster

39: This is again operationalizing the notion that density corresponds to variability

40: We note that we merely *propose* this as a mechanism that needs to be evaluated experimentally; but this is indeed the *intended* use of visualization — to motivate interesting analyses

41: one-sided Mann-Whitney U test, $p = 6.61 \times 10^{-7}$ for the PBMC2 dataset, $p = 2.89 \times 10^{-4}$ for the PBMC3 dataset, see Subsection 6.2.1 and Figure G.14

42: Indeed, this is a cell-type that again motivates the difficulty of classification

43: again, in the exploratory manner of visualization

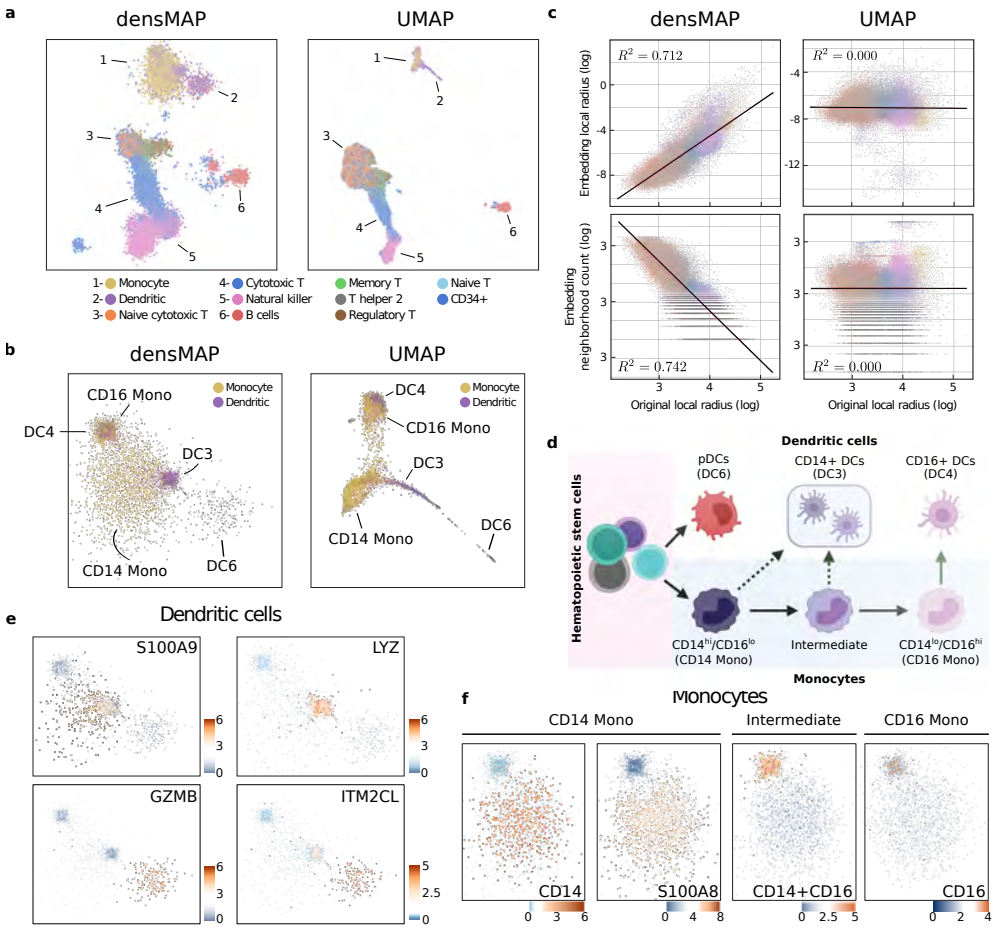

Figure 6.2: Density-preserving visualization of peripheral blood mononuclear cells reveals monocyte and dendritic cell subsets that differ in transcriptomic variability. a. We visualized the PBMC dataset [121] using densMAP (left) and UMAP (right), colored by cell type. **b.** The same visualizations restricted to the monocyte-DC subset revealed distinct subtypes of monocytes (CD16 Mono and CD14 Mono) and DCs (DC3, DC4, and DC6) with clear density differences in densMAP (using the classification from PBMC2 [131]). Density difference between the subtypes is lost in UMAP **c.** Scatter plots comparing the local radii in the original space and local radius and neighborhood count in the visualization (embedding) for densMAP and UMAP, colored by cell-type. Higher correlations in densMAP support the validity of the observed density differences.. **d.** Graphical illustration showing the biological relationships among the five monocyte and DC subtypes we found in the monocyte-DC subset. Under inflammatory conditions, CD14 Mono (classical monocytes) differentiate into CD16 Mono (non-classical monocytes) for immune response. Both CD14 Mono and CD16 Mono can differentiate into DCs. DC6 represents plasmacytoid DCs (pDCs), which come from a different differentiation trajectory than the rest. densMAP visualization suggests that the differentiation paths from CD14 Mono to CD16 Mono and DC3 both represent specialization with considerable decrease in transcriptomic variability. **e.** Gene expression heatmaps of DC marker genes from PBMC2 [131] for DC3 (top) and DC6 (bottom) in the densMAP visualization restricted to DCs. These support our assignment of DC clusters to DC3 and DC6 (see Figure G.15 for a comprehensive set of heatmaps). **f.** Gene expression heatmaps of monocyte marker genes CD14, S100A8, and CD16 in the densMAP visualization restricted to monocytes. CD14+CD16 indicate joint expression of the two genes, which is set to their mean if both are expressed, and zero otherwise.

we hypothesize that (i) corresponds to classical monocyte-derived DCs[44] (cDCs); (ii) corresponds to the poorly understood CD141– CD1C– DCs[45]; and (iii) corresponds to plasmacytoid DCs[46] (pDCs); (see Figure G.15). Despite the apparent closeness of the DC6 and DC4 clusters, we did not find any evidence that either subtype is derived from the other.

44: referred to as DC3 in PBMC2

45: referred to as DC4 in PBMC2

46: referred to as DC6 in PBMC2

Our visualizations reveal that the DC3 cluster is far denser than the CD14 monocytes colocated with it, hinting that, as with CD16 monocytes, these cells specialize as they develop from CD14 monocytes. Similarly, in PBMC2, the DC3 cluster is significantly denser than the classical monocyte cluster[47]; (Subsection 6.2.1 and Figure G.14). In addition, the pDC cluster expands drastically in the density-preserving visualization compared to the standard visualization, revealing previously hidden variability (Figure 6.2b). The PBMC3 dataset was omitted from this analysis as it contained too few DCs to draw conclusions about subtypes.

47: one-sided Mann-Whitney U test, $p = 5.43 \times 10^{-14}$

We also note the DCs dispersed throughout the CD14 monocytes (Figure 6.2b). When we classify the DC3 subset into dense and sparse categories based on their original local radius[48], we find that the sparse subset has *intermediate* expression of the marker genes of DC3 and those of CD14 monocytes (Figure G.15). While this could be due to misclassification[49], it could also indicate a bridging state between the two cell types, offering insights into the dynamics of cell state transition. These results suggest that there are key differences in transcriptomic variability among immune cell subtypes that are obscured by existing visualization tools.

48: with a log-scale threshold (determined *ad hoc*) of 3.9

49: The original study assigned cell types based on similarity to purified samples

6.2.1 Assessing significance of density differences in monocytes and dendritic cells

To verify our claims that classical (CD14+) monocytes have more variability of expression than both CD16+ monocytes and DC3 dendritic cells (as characterized by the PBMC2 dataset), we compared the distribution of the log local radius in the original data for each of these cell types in the PBMC2 and PBMC3 datasets. To assess significance, we used the one-sided Mann-Whitney U (MWU) test [137], which tests the hypothesis that values drawn from one distribution are larger than those drawn from another. We calculated the MWU test statistic for: CD14+ monocytes and CD16+ monocytes in the PBMC2 and PBMC3 datasets; and for CD14+ monocytes and DC3 dendritic cells in PBMC2. In PBMC2, there are 163 CD14+ monocytes, 122 CD16+ monocytes, and 107 DC3 cells; in PBMC3,

there are 1,264 CD14+ monocytes, 398 CD16+ monocytes, and 142 DCs.

6.3 Visualizing Time-dependent Transcriptomic Variability in *C. elegans* Development

To explore embryo development at high-resolution, Packer et al. [16] performed scRNA-seq profiling of *C. elegans* to create an atlas of gene expression at almost every cell division of the embryo. We asked whether density-preserving visualization could better capture the diversification[50] of different developmental lineages, complementing investigations into time-dependent patterns of gene expression in organism development and cellular differentiation [138–140].

50: or lack thereof

lineage distance: the number of generations to the nearest common ancestor

For most of the cell types profiled, the **lineage distance** between cells correlates strongly with transcriptomic dissimilarity, and many cells from the same progenitor diverge after gastrulation [16]. Thus, an accurate visualization should show that the density of cells for most cell types decreases over time[51], as the cells adopt their terminal fates. While both densMAP and UMAP show a central "progenitor" region that branches into the different major tissues, densMAP more clearly highlights the increase in variability in the outer branches of the lineages (Figure 6.3a and b). Evaluating the agreement between the local radius in the original dataset and both measures of local density in the visualization show that densMAP[52] more accurately preserves density than UMAP[53] (Figure 6.3c and Figure G.16). Results are analogous when comparing den-SNE[54] to t-SNE [55] (Figures G.16 and G.17).

51: since this would reflect the increasing diversity

52: $R^2 = 0.590$ for local radius; average $R^2 = 0.585$ for neighborhood count

53: $R^2 = 0.045$; $R^2 = 0.052$

54: $R^2 = 0.619$; $R^2 = 0.596$

55: $R^2 = 0.000$; $R^2 = 0.063$

While transcriptomic variability generally increases over the course of differentiation, notable exceptions are also made apparent by densMAP. Specifically, of the cell types well-represented[56], the intestinal, body-wall muscle (BWM), and hypodermis cells show relative homogeneity in density[57] throughout embryo development when compared to other cell types, e.g. both non-amphid and amphid neurons and seam cells; densMAP more accurately preserves these temporal changes in local density than UMAP (Figure 6.3d and e).

56: We define this *ad hoc* as those with greater than 1000 cells

57: This is measured by the average local radius in the original dataset across time

clades: in population genetics, a clade is a population that has the same common ancestor

The underlying biology supports these visual patterns since intestinal, BWM, and hypodermis cells are so-called *semi-clonal lineage* **clades** [16]. A semi-clonal lineage model is intermediate between *clonal* development, which closely adheres to the lineage structure whereby branching patterns in cell proliferation leads to increasingly more divergent cells, and *non-clonal* development, where daughter cells are

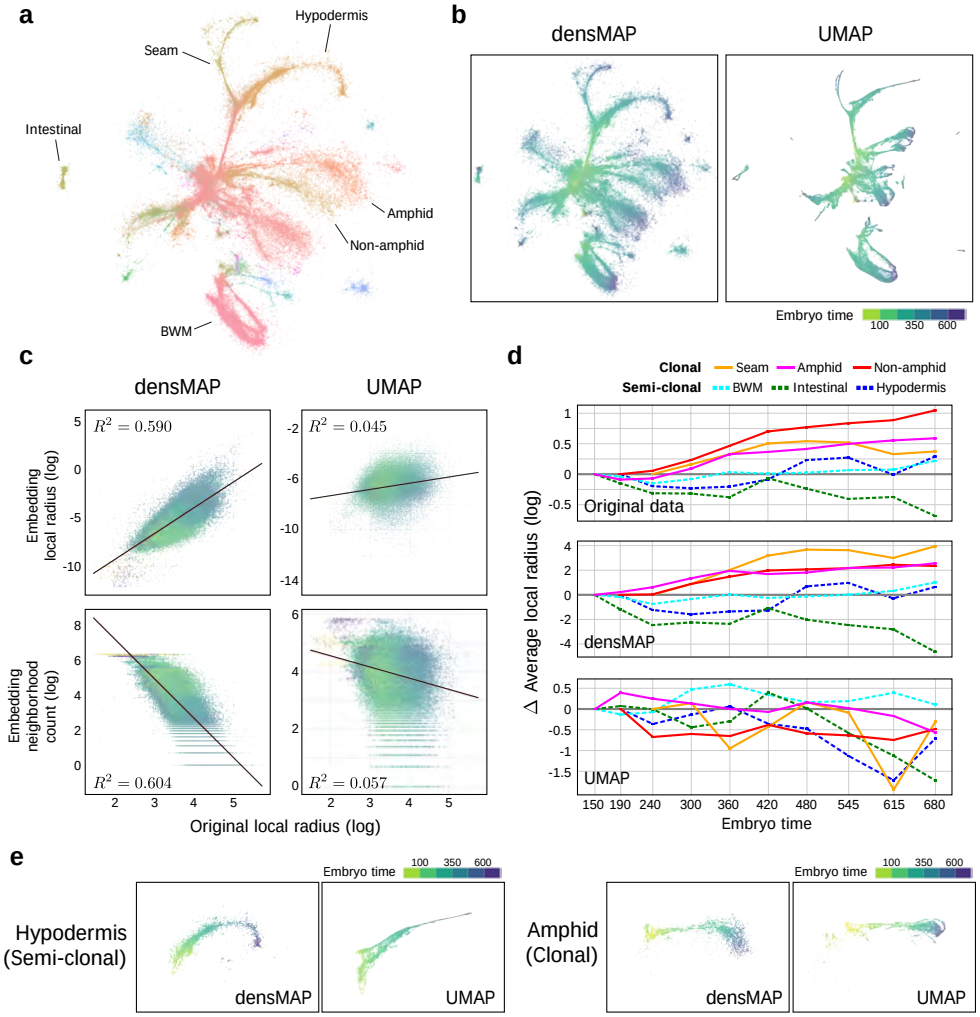

Figure 6.3: Density-preserving visualization of *C. elegans* development reveals temporal dynamics of transcriptomic variability in different developmental lineages. Visualizing the *C. elegans* dataset [16] using densMAP and UMAP, colored by (**a**) cell type (major cell types labeled) and (**b**) embryo time. densMAP clearly conveys an overall increase in transcriptomic variability as the organism develops. **c.** Scatter plots comparing the local radii in the original space to local radius and neighborhood count in the den-SNE embedding. Higher correlations of den-SNE show that den-SNE more accurately conveys the density landscape. **d.** To assess lineage-specific patterns of transcriptomic variability, we summarized the average local radius of each cell type (marked by different line style) within each embryo time interval for the original data (top), densMAP (middle), and UMAP (bottom). The plot for original data represents the temporal changes in the underlying transcriptomic variability of each cell type, and the plots for densMAP and UMAP represent apparent changes in variability based on the respective visualizations. The y-axis shows the change in average local radius compared to the earliest time interval in log scale. densMAP closely follows the temporal patterns of each cell type unlike UMAP These patterns uniquely captured by densMAP highlight the relatively constant variability of semi-clonal lineages (BWM, intestinal, and hypodermis) in contrast to the increasing variability of clonal lineages (seam, amphid and non-amphid neurons). **e.** densMAP and UMAP visualizations restricted to hypodermis and amphid cells for comparison, colored by embryo time. UMAP vastly under-represents the variability of the terminal cell state for hypodermis. Similarly, for amphid cells, densMAP accurately portrays expanding variability, a pattern that is lost in UMAP. BWM: body-wall muscle.

only loosely associated with their progenitors and different lineage branches share commonalities through horizontal transitions [141]. Semi-clonal cell types are thus expected to remain more compact in expression space than clonal lineages. Indeed, when we compare the average change in density over embryo time[58] for semi-clonal cells, this change is considerably lower than the average change for the other cell types (Figure G.17). The difference in density between these semi-clonal cell types and the rest is made clear in our density-preserving visualization but completely hidden by UMAP. In fact, the UMAP plots tend to show a *decrease* in density in many lineages because fewer cells were profiled at the late time-points (Figure G.17). Our methods can thus accurately portray continuous changes in transcriptomic variability in developmental trajectories, which are not captured by existing visualization tools.

58: This is just a measure of which level of cell division the given cell emerges

6.4 General Applicability of Density-preserving Data Visualization

Visualizing high-dimensional data is broadly useful both within and outside biology. Like t-SNE and UMAP, our density-preserving methods require only a distance metric defined between data points. To illustrate the performance of our methods on other data domains, we analyzed a genotype dataset from the UK Biobank and the MNIST image dataset widely used by the machine learning community.

The UK Biobank [122] (UKBB) project collects extensive genotypic and phenotypic data from British individuals for use in health-related research. Due to the skew in ethnicity of the British population, most of the individuals in the dataset self-identify as white[59]. This lack of diversity has raised important concerns about biases in downstream scientific analyses [142]. When visualizing the individuals in the dataset based on their genotype profiles, an analytic approach that is increasingly being explored[60] [143], t-SNE and UMAP show the cluster corresponding to white individuals disproportionately large, while the clusters corresponding to Asian and Black people can scarcely be seen[61] (Figure 6.4). Visualizing this data using den-SNE and densMAP results in a more balanced representation of ethnicities, considerably expanding the people-of-color clusters and shrinking the white cluster (Figure 6.4). Existing visualization tools thus grossly under-represent the genetic diversity of minority populations due to their limited sample sizes. Even among the white population, density-preserving visualizations obtain a more balanced representation of subpopulations[62]. In the UMAP and t-SNE visualizations, only the

59: 94% of the 534k individuals at the time of this study

60: I note here that the traditional method of using the Hamming distance between the SNP vectors of different individuals as the input to these algorithms has had its utility questioned — I too am skeptical but the upshot here is that *if* one uses these methods, one should at least use them accurately!

61: The obvious and problematic interpretation of these visualizations is that because the cluster of white people is so large, they *must* have more genetic diversity

62: These subpopulations are computationally identified using off-the-shelf clustering

two most populous subgroups take up significant space, whereas densMAP and den-SNE show five subgroups with comparable diversity.

A complementary situation occurs in the MNIST dataset, a dataset of handwritten digit images. Here, t-SNE and UMAP generate ten evenly sized clusters; den-SNE and densMAP visualizations, however, reveal that the cluster corresponding to the digit **1** is strikingly less variable than the other digits (Figure 6.5). This is as expected, since **1** is drawn with considerably limited degrees of freedom[63]. Analyzing the local radii in the original data reveals that, indeed, **1** has a higher density than the other digits (Figure 6.5). The improved accuracy of our visualizations for UKBB and MNIST datasets are supported by both density-preservation metrics based on local radius and neighborhood count (Figures 6.4, 6.5, G.18 and G.19). Taken together, these results show that density-preserving visualization reveals important insights about the data not captured by the existing methods on diverse types of datasets[64].

63: This is a fancy way of saying that **1** is merely a line, whereas other digits can be drawn more "creatively"

64: We note briefly here that densMAP has been used to represent DNA methylation data [144], political polarization [145] (also in unpublished work from Data for Progress), and heavy metal spectroscopy [146], among others!

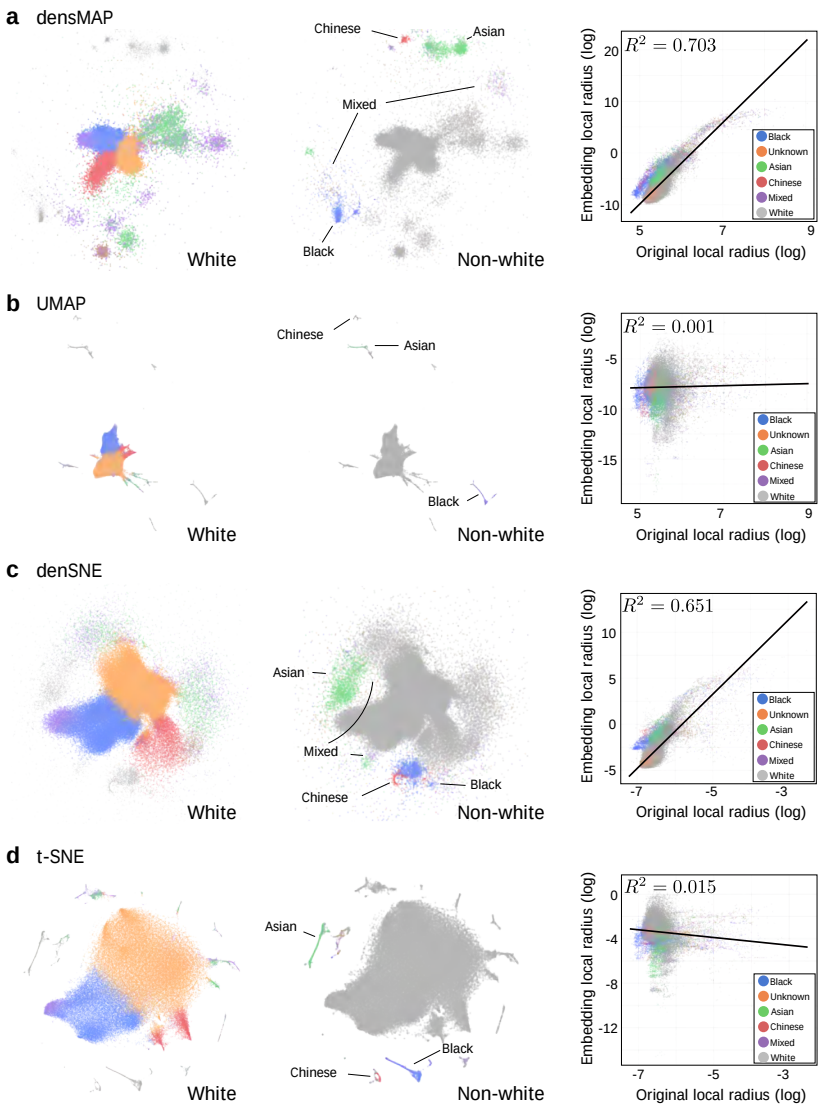

Figure 6.4: Density-preserving methods more accurately visualize diversity of small subpopulations in UKBB data. We visualize the genotype profiles of 97,676 UKBB participants (a 20% subsample of the dataset) using (**a**) densMAP, (**b**) UMAP, (**c**) den-SNE and (**d**) t-SNE. For each, in the left plot, points corresponding to white people are colored by five computationally-identified subpopulations; in the middle plot, non-white people are colored according to their ethnicity; right shows correlation of local radius between the original dataset and the embedding, with points colored by ethnicity and R^2 reported. We show the analogous scatter plots using neighborhood count to measure local density in the visualization in Figure G.18. As 94% of the the people in the UKB dataset self-identified as white, the UMAP and t-SNE plots give overwhelming visual space to this group, hiding the genetic variability of the other ethnic groups. The density-preserving plots, however, clearly expand the clusters of non-white people as well as certain white subpopulations, more accurately conveying their genetic diversity.

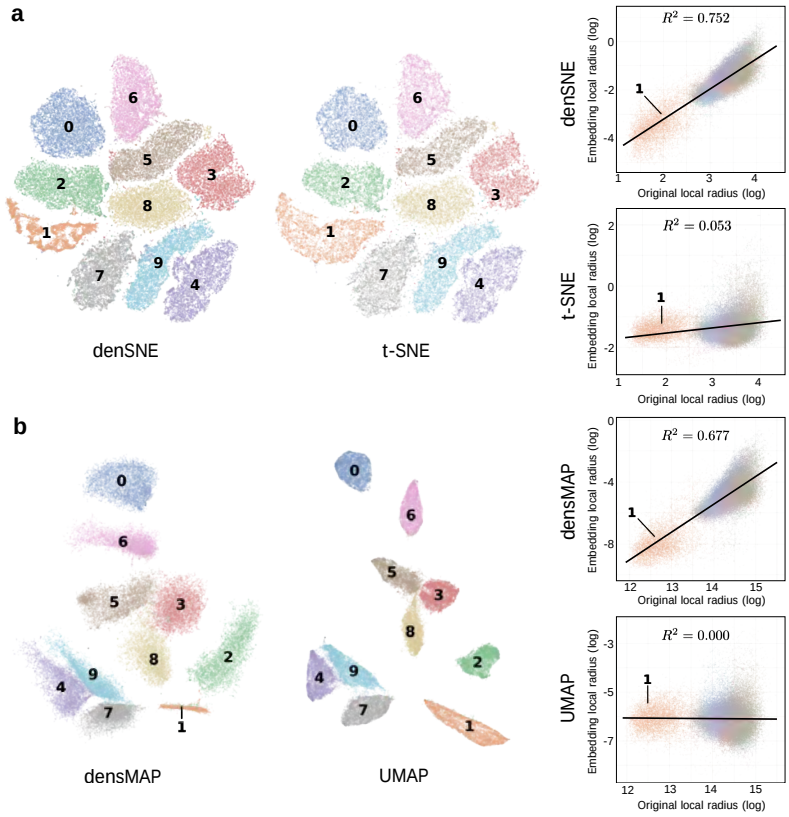

Figure 6.5: Density-preserving visualization of MNIST handwritten digit image dataset reveals the relative homogeneity of the digit *1*. We visualize the MNIST handwritten digits with (**a**) den-SNE and t-SNE and (**b**) densMAP and UMAP, with points colored by digit. Note that the size of the cluster corresponding to the digit **1** shrinks under both density-preserving algorithms. Plots on the right show the correlation of the local radii between the original dataset and the embedding in each algorithm, with points colored by digit and the R^2 score reported. The higher R^2 for the density-preserving methods illustrates that the digit **1** indeed has higher density than the other digits. We show the analogous scatter plots using neighborhood count to measure local density in Figure G.19.

6.5 Practical Considerations

6.5.1 Density-preserving visualization is almost as efficient as existing approaches

65: Datasets have reached orders of millions of cells

As experimental methods continue to generate larger datasets[65], computational tools to analyze them need to scale as well. By leveraging computations already done by t-SNE and UMAP, our density-preserving methods incur only $O(n)$ additional computation[66] and achieve the same asymptotic scaling as those methods[67]. Although density preservation increases the overall runtime of den-SNE and densMAP[68], we believe that this additional cost is not onerous, when weighed against additional information conveyed by accurately depicting density[69]. While t-SNE, even without density preservation, has limited scalability to datasets approaching many hundreds of thousands of cells, recent computational improvements to t-SNE for massive datasets [102, 147] could be augmented with our density-preservation technique. The memory requirements of den-SNE and densMAP are nearly identical to those of t-SNE and UMAP, respectively (Figure 6.6).

66: where n is dataset size

67: Both existing methods scale super-linearly, estimated at $O(n \log n)$ with approximations made

68: an overhead of about 30% for den-SNE and 20% for densMAP on our largest dataset with 250k points

69: Of course, this is a subjective claim — one can modulate the overhead by reducing the number of density-preserving computations

6.5.2 Data preprocessing

We obtained three publicly available scRNA-seq datasets for the main analyses: a dataset of immune cells in lung cancer and blood [17], a dataset of peripheral blood mononuclear cells (PBMCs) in healthy individuals [121], and a dataset that profiled the developmental trajectory of *C. elegans* [16]. We used three additional scRNA-seq datasets for validation experiments, including another lung cancer dataset [129] and two blood immune cell datasets [131, 135]. For each dataset, we applied the same cell and gene filtering schemes used by the original publications[70], then normalized the data so that each cell has the same total number of counts[71]. Following the standard in scRNA-seq analysis, we then log-transformed the normalized counts[72], i.e. $x \to \log(1 + x)$. Principal component analysis (PCA) was then used to produce lower-dimensional representations of individual cells, which are provided as input to the visualization algorithms. We used the number of principal components (PCs) prescribed by the original publications if present, or used 50 dimensions otherwise[73]. The resulting datasets for the main experiments included 48,969 cells and 306 PCs (representing 34.7% of total variance) for lung cancer, 68,551 cells and 50 PCs (9.7%) for PBMCs, and

70: While each data set has its bespoke techniques, these generally filter out low-quality cells and focus on "important" genes

71: The "units" of the value at cell i for gene j are thus "transcripts of gene j in cell i per 10k transcripts of cell i

72: While this is mostly an empirically motivated step, the justification is to "flatten" the data, so genes with extremely high counts do not dominate the analysis

73: This parameter is generally chosen in an *ad hoc* manner — it is an active area of research to choose the "right" number of PCs

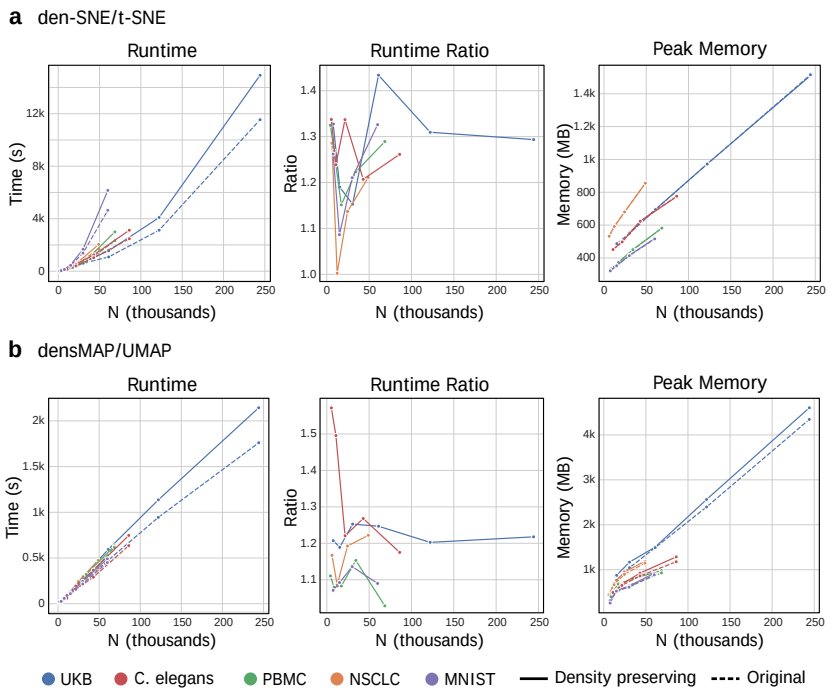

Figure 6.6: den-SNE and densMAP are nearly as efficient as t-SNE and UMAP in runtime and memory. We compare (**a**) den-SNE and t-SNE and (**b**) densMAP and UMAP with respect to runtime and peak memory usage on all of the datasets analyzed in this study. For these tests, we exclude the time taken to compute the local radii of the final embedding, which is used only for evaluation and does not affect the embedding. Left plots running time in seconds at different data sizes (achieved by subsampling the datasets); middle shows the ratio of the density-preserving algorithm's runtime to that of the original method; right shows peak memory usage over different data sizes. Although density-preserving methods take longer, the overhead is small (around 30% additional runtime for den-SNE and 20% additional runtime for densMAP both for our largest dataset). Both densMAP and UMAP obtain fast runtimes for large datasets, taking less than ~30 minutes for all our datasets. Peak memory usage is the same between t-SNE and den-SNE, and differs by a small constant between UMAP and densMAP.

86,024 cells and 100 PCs (25.2%) for *C. elegans*. We used the cell type labels provided by the original datasets for visualization.

For the UK Biobank dataset [122], we used the 40 PC loadings provided as part of the genetic data for visualization. We analyzed a 20% subsample of the dataset including 97,676 individuals, for computational efficiency. Ethnicity labels for the individuals were obtained from Data Field 21000, which was collected from the participants via a touchscreen questionnaire. To visualize subpopulation structure within the white British individuals, we performed spectral clustering using the 40 PCs as input to identify five subclusters.

For the MNIST dataset, we flattened each of the 60,000 28× 28 pixel images to a 784-dimensional vector and used the top 50 PCs (82.4% of total variance) as our input to the visualization algorithms. Labels classifying the handwritten digits were provided in the dataset.

6.5.3 Runtime and memory benchmarking

To evaluate runtime and memory usage of our density-preserving visualization methods, we used each of the five datasets[74] along with logarithmically downsampled subsets of each[75]. The dataset from Packer et al. [16] with 86,024 cells is the largest scRNA-seq dataset used in this paper. In addition to the full dataset, we subsampled it into smaller datasets, including 43,012 cells, 21,506 cells, 10,753 cells, and 5,376 cells. We measured the runtimes of den-SNE, densMAP, t-SNE, and UMAP on each of the datasets with the default parameter settings and profiled memory usage using the `psrecord` package[76]. All experiments were run on an Intel Xeon Gold 6130 (2.30 GHz) processor and used a single core.

74: the three scRNA-seq datasets, UK Biobank, and MNIST

75: that is, subsamples of size $N/2$, $N/4$, down to 1,000 datapoints for a dataset of size N

76: https://github.com/astrofrog/psrecord

6.5.4 Data availability

The lung cancer [17] and *C. elegans* [16] datasets are available from the Gene Expression Omnibus (GEO) database with accession numbers GSE127465 and GSE126954, respectively. The PBMC dataset [121] is available from 10x Genomics[77]. For our validation datasets, the secondary lung cancer dataset [129] is available from GEO (GSE99254), and the PBMC2 [131] and PBMC3 [135] datasets can be accessed through the Broad Institute's Single Cell Portal [78] with dataset IDs SCP43 and SCP345, respectively. Data access applications for the UK Biobank data can be submitted online [79]. The MNIST dataset is available online[80]. We also provide our preprocessed data for the main datasets (lung cancer, PBMC, and *C. elegans*)[81].

77: https://support.10xgenomics.com/single-cell-gene-expression/datasets

78: https://singlecell.broadinstitute.org/

79: https://www.ukbiobank.ac.uk/

80: http://yann.lecun.com/exdb/mnist/

81: http://densvis.csail.mit.edu/datasets

7 Local k-nearest Neighbors Graphs

7.1 Introduction

Recent methods for the analysis of high-dimensional datasets have taken advantage of the latent structure that these data might have — namely, that the data lie upon some low-dimensional manifold. One of the key consequences of this hypothesis is that computing similarity between data points *cannot* merely be done using Euclidean distance; rather, the geodesics along a manifold have to be learned using the data.

Methods to learn these geodesics rely on the fact that manifolds are locally Euclidean, and thus piece together "global" distances by making several short "jumps" between the points. Thus, in many ways the foundational data structure for manifold learning is the k-nearest neighbors graph, where each data points is connected to its k nearest neighbors in the dataset[1].

Despite their key importance, the process of constructing k-NN graphs has received comparatively little attention. Most methods tend to assume that, as a baseline, the k nearest points in terms of Euclidean distances accurately reflect the manifold neighborhood and covariance structure[2] at that point. In this work, we interrogate this assumption and present a method for calculating *local* distances informed by the local covariance structure of the manifold[3].

The presence of noise is what motivates this question. As we discuss in Section 7.3, our generative model assumes a sparse covariance structure at each point, but potentially significant *uncorrelated* noise across the features[4]. In high-dimensional conditions, the uncorrelated noise can drown out the signal of the *actual* neighbors in the manifold.

1: This is essentially the diffusion distances discussed in Subsection 2.5.3

2: Waving our hands, by "covariance structure", we generally just mean how one coordinate varies with another coordinate around a given point

3: Essentially, our contention is that the scale of the nearest neighbors is *not* small enough to be considered naïvely Euclidean

4: This causes points to "fall off" the manifold in the manner of noisy Swiss roll in Figure 2.9

We thus present topological stitching, which breaks the dataset up into coherent locales, and learns the relevant covariance structure *within* each locale; then, the distance information induced by each locale is stitched back together to produce a global k-NN graph.

Our main motivation is high-dimensional biological data, where, empirically, the scenario of significant uncorrelated noise and nontrivial manifold structure, is ubiquitous[5]. We now motivate the biological case where our method is most applied.

5: This is mostly an empirical claim, as our results will show

7.1.1 Single-cell RNA-sequencing challenges

Recent advances in experimental methods have given biologists an unprecedentedly high-resolution view into the transcriptomic profiles of individual cells[6]. Single-cell RNA sequencing (scRNA-seq) is a sequencing technology that, for each cell in a sample, counts the amount of each particular gene transcript that is expressed in that cell.

6: Much of this material is covered in Section 2.2, but we include the section here for completeness

The resulting data gives us a so-called *expression profile* for each cell. Under the hypothesis that cellular function is determined by the proteins present in the cell, the differences in expression profiles help us understand the underlying mechanisms for cellular differences.

While new methods for scRNA-seq are actively being developed, they follow a similar framework: RNA fragments that are complementary to the transcripts of interest are first prepared. Then, cells in the sample are individually[7] combined with sets of these complementary transcripts, and each set is identified with a unique "barcode". In this way, the sequenced transcripts can be traced back to the cells that they came from.

7: For example, in *droplet based* sequencing methods, cells fed through a stream of oil droplets, so each oil droplet should contain a single cell

Single-cell RNA-sequencing technologies, by allowing us to assess the expression profiles of individual cells, have greatly expanded our understanding of the cells that modulate immune responses to tumors [148, 149]. However, a major hurdle is recovering rare cell types, which comprise just a small portion of the sample[8] and are often distinguished from other cell types by just a handful of marker genes, but may play important roles in the tumor microenvironment. For example, gamma-delta and MAIT cells have shown promise in early trials despite accounting for 5% or less of immune cells present [150, 151]. Traditional clustering and dimensionality-reduction pipelines frequently fail to highlight these cells [44, 46].

8: by definition!

Mathematically, a scRNA-seq dataset with N cells and G genes can be thought of as a matrix $X \in \mathbb{R}^{N \times G}$, where each row represents a

cell and each column a gene — so X_{ij} is the number of transcripts of gene j found in cell i[9] Analyzing a dataset thus comes down to understanding similarity between these vectors: finding cell-types by clustering the data, finding the genes whose expression drives cell-types, and inferring trajectories, are just some of the tasks for the computational biologist.

9: In order to deal with large counts, a *log-transform*: $x \mapsto \log(1+x)$ is common as well.

In computing distances between cells, most existing methods rely on some form of Euclidean distance, potentially after some global linear transform. For example, one extremely common pre-processing technique in the field[10] is to use principal component analysis (PCA) on the original data, taking the first M principal components[11], and then computing a k-nearest neighbors (k-NN) graph (see Subsection 7.2.1) based on those components. That k-NN graph is then the input to downstream algorithms such as clustering, visualization, and trajectory inference [38, 39, 110, 152]

10: It might even be considered the default

11: M is commonly less than 100

7.1.2 Problems with global decomposition

The global decomposition step has severe limitations. In general, there are mathematical limitations to how well high-dimensional distances in a dataset can be represented in low-dimensions. For example, in d dimensions, you can construct a set of at most $d+1$ points which are all pairwise equidistant, so low-dimensional representations *must* distort some relationships[12].

12: Even having *approximately* equidistant points is difficult, another consequence of Theorem 2.6.1 [51]

But perhaps more importantly, even *calculating* the distance between points in high-dimensional datasets is difficult. For example, because of the noisiness of scRNA-seq data, the expression counts of some genes can be uncorrelated noise in some region of the data, but highly informative in another region[13]. For example, in Figure 7.1, we see each gene shows up as uncorrelated noise for a dyadic subset of the cells, and then becomes a marker gene for the remaining cells. While we go into details in Subsection 7.4.1, we note that the classes would not be nearest-neighbors *because* of the noisy features. Thus, it is crucial to find exactly *which* features are noise and which are markers in a local neighborhood.

13: That is, a marker gene for one cell-type could just be noisy in other regions

Notably, we see in the above example that even when the Euclidean metric is used *only* to calculate nearest neighbors[14], the presence of noisy genes makes even the nearest-neighbors noisy. Downstream methods that use these nearest-neighbors data as an input have a fundamental assumption: that cells and their neighbors are similar — and the presence of these **cross-edges** belies this assumption.

14: as is common in many downstream methods

cross-edges: when different cell-types are each other's nearest neighbor

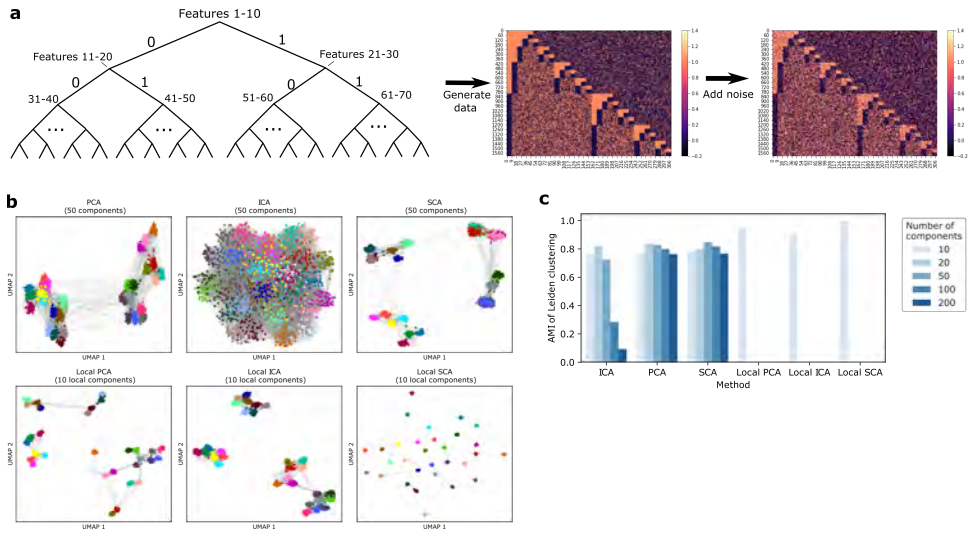

Figure 7.1: Results on synthetic, hierarchical data. a. Construction of the synthetic dataset. Clusters are related in a binary tree structure, with different sets of features defining each split. We then add noise to the resulting dataset to simulate dropouts and variable capture rate. **b.** UMAP plots with k-nearest-neighbor graphs computed downstream of global PCA, ICA, and SCA (top), and local version (bottom), colored according to cluster. topological stitching consistently produces better separation between clusters. **c.** Adjusted mutual information (AMI) of Leiden clusterings downstream of each method with the true cluster labels. topological stitching gives higher AMI than global dimensionality reductions, even with as many as 200 global components.

7.1.3 Introducing local dimensionality-reduction

In the synthetic example in Figure 7.1, the main confounder was that features that are *noise* in a given context are used to compute local distances. In an ideal world, for a datapoint i, the only features that would be used to compute its neighbors would be the set of features $\mathcal{F}_i$ that are *not* noise for that cell.

We present a novel algorithm which combines geometric, topological, and statistical properties of single-cell data to learn context-aware distance metrics based on localized feature selection. Combining these metrics to create global affinities enables detection of small and clinically relevant immune cell subpopulations at high resolution, with the potential to guide the development and targeting of immunotherapies.

The goal of topological stitching is to create more accurate nearest-neighbor graphs for representing scRNA-seq datasets by detecting informative genes *locally* and using that information to learn more accurate *local* distance functions.

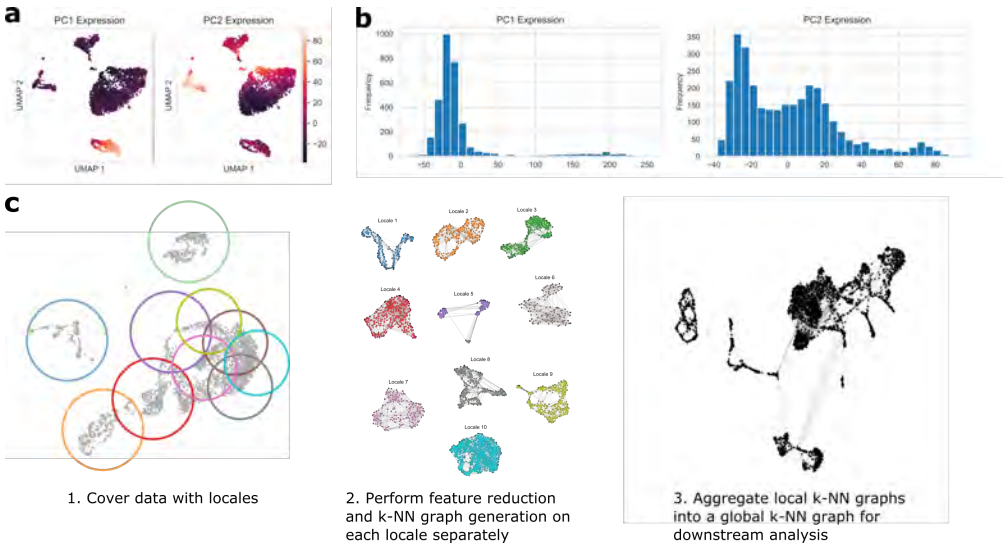

Figure 7.2: Overview of topological stitching. a: Expression of the leading principal components of the single-cell dataset from [153]. PC1 identifies HEK cells, and PC2 identifies B cells (see Figure 7.4a). **b**: Histogram of expression values for the leading principal components. **c**: Overview of topological stitching. We cover the data with overlapping locales, and generate k-nearest neighbor graphs in each locale separately. We then aggregate these local k-nearest neighbor graphs into a global one for downstream analysis, using mean random walk with restart probabilities (Methods).

7.2 Methods

Topological stitching essentially consists of three main pieces: (1) the dataset is covered by overlapping locales, where each cell is included in at least C locales; (2) dimensionality reduction is performed *within* each locale: the underlying assumption is that axes of variation within a locale better capture differences between the cells therein; (3) since each cell is in multiple locales, the final step is to integrate the locales back together into a global k-NN graph (see Figure 7.2). We discuss each briefly before going into detail below:

Outline of topological stitching

- **Generating locales:** The key consideration for the initial split of the dataset is to ensure that *every* cell is well-represented. In other words, every cell should be in some locales where it is relatively close to the center, and so its expression is close to the average expression within the locale. A particularly important consideration is rare cell types — a good method for locale construction would ensure that even rare cell types are in multiple locales. We detail our method for locale construction in Subsection 7.2.2.
- **Local feature reduction:** Within a locale, our goal is to find the genes that are most informative in constructing

distance metrics between the cells. While we are generally agnostic to the method used to do the local decomposition, we note that one of the major benefits of our focus on locales is that even straightforward linear decomposition methods (like PCA) perform well. We motivate PCA and the recently developed surprisal component analysis (SCA) [35] in Subsection 7.2.3; and in Subsection 7.3.1, we discuss a local factor model interpretation that explains *why* local linear decompositions can perform well.

- **Topological stitching:** After the local decomposition step, every locale has an induced distance metric. In order to get a global k-NN graph, the different induced distances need to be integrated. Specifically, consider points i and j which are *both* in locales L_1 and L_2. This means the "global" distance between has to reconcile between $d_{L_1}(i,j)$ and $d_{L_2}(i,j)$, the distance in L_1 and L_2 respectively.
  In Subsection 7.2.4, we detail our method for stitching: mean *random walk with restart* (RWR) distance. Essentially, we take the average distance across locales between the two points using the probability of getting one point to other through a random walk — this ensures that points that are well connected with each other have high similarity.

7.2.1 Mathematical setup

We are given an scRNA-seq dataset of N cells, where cell i is represented by a G-dimensional vector $x_i \in \mathbb{R}^G$. The dataset can thus be represented by an $N \times G$-dimensional matrix X.

Given a distance function d_X where $d_X(i,j)$ is the distance between cells i and j, the k-neighborhood of i, denoted $N_i^{(k)}$, is the set of k points in X that are closest to i[15]. The k-neighborhood structure of X, denoted $N_X^{(k)}$, is then the function:

15: We will often drop the superscript k when it is clear

$$i \mapsto N_i^{(k)}$$

From the k-neighborhood structure, we can generate the k-nearest-neighbors (k-NN) graph for X, $G_X^{(k)}$. Each point in the dataset is a vertex in the graph, and the edges from i are those to the points in $N_i^{(k)}$.

Our goal is thus to learn the *best* distance function d^* so that a global k-NN graph can be made for downstream analyses. As discussed in Section 7.2, this is a three-part problem: constructing

locales, decompositions within locales, and stitching the datasets back together. We discuss each section in turn.

7.2.2 Locale construction

The goal of dividing the entire dataset up into locales is that gene coexpression patterns will be more coherent at these smaller scales than across the entire dataset [42]. Clustering the dataset and using the clusters as locales is the obvious way to achieve this division.

However, a clustering does not represent all cells equally well — cells near the center of a cluster are probably more similar to each other than to those at the periphery[16]. And, since one of the motivating reasons for our topological stitching method is that errors in naïve distance metrics leads to poor clustering, we have to recognize the possibility that peripheral cells would fit better in a different cluster *or in an entirely different clustering*.

16: Of course, this is depending on the clustering method; if the peripheries are clustered separately, this might not be true

To that end, we perform a *multiclustering* of the data. That is, we make sure that each point is in *several* locales. We aim thus to ensure that each point is well suited[17] in at least some subset of the locales it is contained in.

17: Morally, our goal is that each point is not in the periphery in at least one of these clusterings, although we cannot guarantee that

The objective function used for generating locales also should take into account the goal that each point should be relatively close to *some* locale center. This is similar to the goal of **sketching** (i.e. subsampling) scRNA-seq data while ensuring that rare cell types are found in the subsample [44, 46] — here the points that are chosen for a subsample can be used as the centers for locales.

For our implementation, we use a modified version of the Hopper algorithm [46] adapted to graph-distances (which we call Graph-Hopper). First, the M cluster centers are chosen according to the Hopper objective of greedily minimizing the *Hausdorf* distance between the centers and the entire dataset: for a given subset $\mathcal{M} \subseteq X$ with M points, the Hausdorf distance is given by:

$$\mathcal{H}(\mathcal{M}) = \max_{x \in X} \left\{ \min_{m \in \mathcal{M}} d(x, m) \right\} \tag{7.1}$$

$$\mathcal{M}^* = \arg\min_{\mathcal{M}} \mathcal{H}(\mathcal{M}), \tag{7.2}$$

where the subset given by (7.2) is the objective of Hopper.

To go from the subset to locales, each point in the dataset is assigned to its C nearest locales[18]. For each locale $m \in \mathcal{M}$, let R_m be the set of points from X assigned to that locale.

18: C is chosen by the user, with the tradeoff being coverage vs efficiency

7.2.3 Local decomposition

The locales R_m become the units of our analysis now — each represents one local piece of the dataset. We now want to separate the noisy and marker genes within these locales: intuitively, we are looking for the genes that show strong patterns of coexpression with other genes for cells within the same cell-type[19].

19: See Subsection 7.3.1 for a discussion on a theoretical description of the difference

The most canonical method for noise reduction is principal component analysis (PCA). The goal of PCA is to find a new orthogonal basis for the dataset, where each basis vector captures the direction of maximum variance in the dataset. Under the condition that the magnitude of noise is smaller than the magnitude of any real coexpression, the top principal components capture the signal in the data.

The issue with PCA, of course, is that it looks at *global* variance across the dataset — this is exactly what we aim to solve by breaking the dataset into locales. In the PCA-version of topological stitching, we thus run PCA on each of the locales, keeping the top p principal components — in other words, for locale m, we learn a linear transform $P_m : \mathbb{R}^G \to \mathbb{R}^p$, and, for points i and j in locale m, the locale-mediated distance becomes:

$$d_m(i, j) = ||P_m x_i - P_m x_j||.$$

We note, however, that when the locales or the noise are relatively large, PCA might not be the most appropriate decomposition method. The surprisal components analysis (SCA) [35] algorithm learns a transform that more explicitly tries to capture which genes are meaningful[20]. We refer the reader to DeMeo and Berger [35] for the details of the algorithm, but briefly, the algorithm learns the genes that are selectively expressed within neighborhoods in the datasets, and learns a linear transformation that emphasizes *those* genes. The default application of topological stitching uses SCA as its local decomposition method.

20: That is, which have significant covariance in the locale

7.2.4 Topological stitching

The crucial last step of topological stitching is the *stitching* back together of the local decompositions from above. Recall that the goal is to generate a *global* k-NN graph, which means that, for each point i in the dataset, a unified distance metric $d_i^*(\cdot) = d^*(i, \cdot)$ must be found, so that the nearest neighbors of i can be found *across* locales.

One fundamental assumption of topological stitching is that the nearest neighbors of i *must* be found within the locales that i is a part of[21]. Thus, the k-nearest neighbors of i can be taken as the k closest points to i across the locales it is part of.

21: This is reasonable, since the locales are chosen with the aim of fully covering the neighborhood of i

For a given pair (i, j), if the set of locales that contain them both is empty, the above discussion indicates that j will *not* be considered as a neighbor of i. A similarly straightforward case is when the pair are both only in *one* locale together. In this case, the distance $d_i(j)$ is the distance induced in *that* locale.

The main nontrivial case is when the pair are shared by *multiple* locales[22]. In this case, each locale gives some information about how to get from i to j, depending on what subspace it is traversing. For our application, we thus just take the *average*[23] distance between the two points across all the locales.

22: In fact, this is the *ideal* case, as it provides multiple pieces of evidence for "closeness" of the points

23: One could consider other aggregation methods; we also tried the maximum, which gave comparable results

Having the distances $d_i(j)$ calculated for all the j that i shares at least one locale with, the computation of the neighborhood N_i is straightforward — choose the smallest k distances.

Of course, the discussion here is heavily dependent on the choice of distance metric. While Euclidean distance can be used at this local scale[24], we take advantage of the local structure of the data and use a form of *random walk distance*, which was defined in Subsection 2.5.3:

24: and would be an improvement over global Euclidean distance

Random walk with restart

Recalling our exposition, the distance between two points is the probability that a random walk starting at one point ends up at the other, with each edge-wise probability represented by the exponential kernel. However, one complication of this formulation is that with enough time steps t, the probability of reaching a point from any other point becomes 1[25].

To get around that, we use a **random walk with restart**, which means that at each timestep, there is a non-zero probability of going *all the way* back to its starting point. This ensures that, even as $t \to \infty$, there is a non-degenerate limiting distribution on the edges between points.

25: Of course, this assumes the graph is connected, which we will indeed assume

7.3 Theory

As with our work in density-preserving visualization, the main validation of our method is in its empirical performance, which we detail below in Section 7.4[26]. However, as we did with density-

26: The utility of methods in bioinformatics is a fraught question that we briefly broached in Section 2.7 — one tenet is that methods that show interesting results that can be biologically validated are useful

preserving visualization, we attempt here to ground our method in some theoretical footing.

We note that the underlying assumption of topological stitching is that, as we range over an scRNA-seq dataset, different genes become marker genes. Our goal here is not to prove that assumption, but rather to show that *if* that assumption holds, then our method can pick up those marker gene variations. Specifically, we consider an extremely simplified model, where the actual expression vectors of the dataset are generated by a straightforward transformation when the marker genes are known. We detail this model below.

7.3.1 Generative model

We are given an input scRNA-seq dataset $X \in \mathbb{R}^{N \times G}$. Assume that the data are generated by some complicated probability distribution $\mathcal{P}$. We further assume that the $\mathcal{P}$ can be decomposed into a distribution $\mathbb{Q}$ which has support on some low dimensional manifold $\mathcal{M}$, and independent noise acting on each gene. Formally, the expression vector of cell i is the sum of random variables:

$$X_i = M_i + E_i$$

where

$$M_i \sim \mathbb{Q} \text{ where supp } Q = \mathcal{M}$$
$$E_i \sim \text{Normal}(\mathbf{0}, D)$$

where D is a diagonal variance matrix.

Learning $\mathcal{M}$ is the goal of *manifold learning*. Here, we restrict ourselves to learning the *neighborhood structure* of the points in our dataset based on the manifold distance.

Given a distance function d_X where $d_X(i, j)$ is the distance between cells i and j, define the k-neighborhood of i, denoted $N_i^{(k)}$, as the set of k points in X that are closest to i. (We will often drop the superscript k when it is clear). The k-neighborhood structure of X, denoted $N_X^{(k)}$, is then the function:

$$i \mapsto N_i^{(k)}$$

Letting $d_{\mathcal{M}}(\cdot, \cdot)$ represent the geodesic distance along $\mathcal{M}$, we define the dataset distance metric $d_X(i, j) = d_{\mathcal{M}}(M_i, M_j)$ — so for each

point, we must be able to separate its manifold component from its independent noise component.

The signal and the noise

The assumption that our data lie on a lower dimensional manifold constrains the shape of the data by restricting the ability of the different genes to be expressed independently. For example, in the *swiss roll* dataset that is ubiquitous in motivating manifold learning, the possible y-coordinate of a point is heavily reliant on its x-value.

This motivates understanding the *covariance* (or in the case of scRNA-seq data, the *coexpression*), between the features. At any point on the manifold, there will be a subset of genes have non-trivial coexpression, and a subset of genes whose expression is just independent noise. Learning the manifold distance between points can thus be reduced to learning the set of non-trivial genes at a particular point and the distance induced *by those genes*[27].

27: Note that we are implicitly taking advantage of the locally Euclidean property of our data manifold, by assuming that a linear covariance structure at each point can essentially determine local distances accurately

Factor models

We rely on the fact that manifolds are (a) locally Euclidean and (b) smoothly varying[28] to help learn the non-trivial non-trivial subset. In particular, we assume that each cell's expression vector is locally generated from a linear *factor model*.

28: This is actually probably a separate condition on the manifold, which we may at some point have to expand on

A factor model is a generative model for a random vector $X \in \mathbb{R}^d$. Let $C = (C_1, \ldots, C_k)^T$ be a vector of k random variables that are independent — these are the *factors* — and let $Q \in \mathbb{R}^{d \times k}$ be a fixed matrix of *weightings*.

Then, we write:

$$X = QC + E,$$

where E is a mean zero noise vector with diagonal covariance $\sigma_E^2 I$, totally independent[29] from C.

29: So the set of genes *not* affected by Q but affected by E make up the "noise genes" from above

It is straightforward to compute the mean and variance of X under this model:

$$\begin{aligned} \mathbb{E}X &= Q\,\mathbb{E}C \\ \text{var}X &= \sigma_C^2 Q^T Q + \sigma_E^2 I \end{aligned}$$

Under certain conditions for the distributions of C and E it can be seen that the MLE for the span of Q is given by looking at the principal components of the emprirical variance matrix. This insight

will be crucial in motivating our local decomposition and stitching methods.

Factor variation

The key idea behind our approach to local dimensionality reduction is that a single factor model cannot adequately describe the entire dataset. Therefore, we assume that the factor matrix Q *varies with each point*, so we treat it as a function:

$$Q : \mathbb{R}^d \to \mathbb{R}^{d \times k}$$

Crucially, we assume that the variation in Q is *smooth* because the underlying manifold $\mathcal{M}$ is smooth, which means $Q(x) \approx Q(y)$ if $x \approx y$. Thus, for a given cell i in X, we can use the expression vectors of its nearest neighbors to compute an empirical variance matrix and thus an estimation[30] for $Q(X_i)$:

$$\widehat{Q}(X_i) = F(\widehat{\text{var}} N_i^{(k)})$$

where F is an inference procedure that we detail below.

30: It should be noted that we cannot identify Q exactly but rather only its span. So when we write for example $\widehat{Q}$, we actually mean $\widehat{\text{span}\, Q}$, which we will not write out for the sake of notational brevity

7.3.2 Inferring the factors

Having defined our local factor generative process, we now turn to the task of inference. Our goal is to learn the underlying factor weight function Q at each of the points in X, and then use the learned factor weights to compute pairwise distances in the dataset.

The factor matrix is ideally suited for computing distances because it is a denoised version of the data. Under our modeling assumptions, the *actual* expression vector is generated through a random process mediated by the factors C and the white noise E, whereas Q is fixed. For example, the structure of the span of Q indicates the dependencies between the different genes in the dataset (and which genes are noise versus having a non-trivial covariance in a particular region).

PCA-version

As mentioned above, the principal components of the variance matrix of a dataset, under certain conditions, give the MLE of the weights matrix for the factors Q. Of course, computing an estimator requires multiple i.i.d. samples from the factor model in question, and our

assumption that Q varies over the dataset means that, of course, the rows of X are not i.i.d. samples. In fact, every element of the dataset is technically drawn from its own distribution[31].

31: There is a Bayesian interpretation where the function Q is drawn from its own prior distribution and the tuple (X_i, Q) are indeed i.i.d. — this is an interpretation we are keen to explore

However, because we assume that the weighting matrix varies *slowly* over the dataset, we make the assumption that the k-nearest neighbors of a point can be thought of as being sampled using the same weight matrix.

For each point i, we thus compute the principal components of the matrix restricted to the nearest neighbors of i, denoted X_{N_i}:

$$T_i = X_{N_i} W_i$$

where W_i are the eigenvectors of $X_{N_i}^T X_{N_i}$. Our estimator for $Q(X_i)$ is then, based on Bishop [154] T_i restricted to the top k principal components (k is the rank of $Q(x_i)$). So we write:

$$\widehat{Q}(X_i) = T_i[k]$$

We should note here that $\widehat{Q}$ as defined above is *not* the MLE of the factor model we have described, since we are not inferring it through i.i.d. samples. However, it is empirically reasonable that cells with very similar expression profiles (i.e. nearest neighbors) fulfill very similar biological functions and are therefore described by very similar biological program. In fact, many algorithms used for analyzing single-cell transcriptomics data actually assume that broad *clusters* of the data can be treated as members of the same population.

SCA-version

While PCA is popular and effective in denoising data and finding the most salient features, recent methods have focused more directly on *locality* (which is our ultimate goal here)[32] We use the recent surprisal components analysis (SCA) algorithm, which learns a (global) linear transformation that purports to project the data onto a space that learns locally important genes.

32: By locality here, we mean that the set of genes with non-trivial coexpression can vary across the dataset.

SCA is built upon the idea of *surprisal*, an idea from thermodynamics used to evaluate how well data fit a particular model. The surprisal of an observation is based on comparing the observed proportion of an observation to the probability of that observation under the explanatory model:

$$S = -\log \frac{p_0}{p}$$

where p_0 is the probability under the null model. For each cell in the dataset, SCA computes the surprisal of each gene, based on its prevalence in the neighborhood of that cell compared to the entire dataset. Computing the principal components *of the surprisal matrix* picks out patterns of high surprisal amongst the genes. SCA was found to effectively parse out rare cell types from within clusters.

We find that SCA is well-suited to our focus on locality, and we build further on this idea by learning *local* linear transformations through SCA instead of one global transform and then stitching those individual transformations together (see Subsection 7.2.4).

To figure out why an SCA-based version of topological stitching has good performance, we consider a relatively simple mode of variation in the factor model Q across the dataset. So, again, each point i has a factor matrix $Q(X_i)$. In the PCA version above, we estimated $Q(X_i)$ (which we write as Q_i below) by assuming Q is constant over a locale. However, we can also assume that, rather than the locale having constant factors, the *variation* of the factor *across* the locale is smooth or easy to describe. To make this concrete, assume that we can infer a factor model Q_{locale} across the locale, and assume there is some variation from the *actual* Q_i for an i in the locale:

$$\|Q_i - Q_{locale}\|^2 \geq \epsilon \|Q_{locale}\|^2 \tag{7.3}$$

singular value decomposition: decomposes a matrix M into $M = U\Sigma V^t$ where Σ is a diagonal matrix of *singular values* and U and V are orthogonal matrices

Considering a particularly simple model variation, we take the **singular value decomposition** (SVD) of Q_{locale}:

$$Q_{locale} = U\Sigma_{locale}V^t = \sum_j \sigma_j u_j v_j^t,$$

where σ_j are the diagonal elements of Σ and u_j and v_j are columns of U and V.

Now, crucially, we assume that Q_i varies significantly from Q_{locale} *only in one singular value*, say σ_ℓ:

$$Q_i = U\Sigma_i V^t = \sum_{j\neq\ell} \sigma_j u_j v_\ell^t + \sigma_\ell^* u_\ell v_\ell^t$$

33: This is not actually necessary, but it might be convenient to think of δ_i as positive

Assuming without loss of generality[33] that $\sigma_\ell^* > \sigma_\ell$, and setting $\delta_i = \sigma_\ell^* - \sigma_\ell$, we see that:

$$\Delta_i = Q_i - Q_{locale} = \delta_i u_\ell v_\ell^t,$$

and crucially,

$$\|\Delta_i\|^2 = \delta_i^2, \tag{7.4}$$

which by (7.3), means that

$$\delta_i^2 \geq \epsilon \tag{7.5}$$

Recalling that the gene expression counts *themselves* for cell i are given by $Q_i C_i$, our goal is thus to determine for which genes the expression of $Q_i C_i$ is markedly different from $Q_{locale} C_i$[34].

Under our model, this set of genes is entirely described[35] by the module $u_\ell v_\ell^t C_i$ — the difference in expected values being something like:

$$\delta_i u_\ell v_\ell^t \mathbb{E}C \tag{7.6}$$

If δ_i is large enough for the points X_i in the locale, then the large differences between expression vectors for a particular cell and its locale will be within the gene module given by the ℓth singular value. If we could thus identify ℓ, then we could precisely identify the marker genes within the locale[36].

In fact, this is *exactly* what the procedure detailed by SCA accomplishes! This is because the elements of the surprisal matrix that have large values are those which have variation across the locale, and so are those determined by the gene module in question. And taking the top principal components of the *surprisal matrix* as SCA does will then identify this gene module[37]. In fact, one could think of the SCA procedure as finding the principal components of *variation* across the locale[38].

34: This is because these are the coordinates that have significant surprisal scores in the SCA matrix

35: We note here that the rest of this analysis is *not* complete and airtight; rather, we are building a conjectural case that hopefully our future work can batten down

36: Again, by marker genes here, we mean genes that have nontrivial covariance across the locale

37: In the example above, only the *top* principal component should be significant because only one singular value changes

38: There is also a kernel-PCA flavor to this algorithm, where the surprisal matrix is the kernel, that PCA is done on, similar to how diffusion maps perform PCA on the diffusion distance matrix (Subsection 2.5.3)

7.4 Results

7.4.1 Topological stitching resolves classes in synthetic data

To demonstrate the limitations of global dimensionality reduction and highlight the advantages of local reduction, we generated a “worst-case scenario" synthetic dataset whose clusters are arranged in a hierarchical binary tree structure, with a different set of features defining each split of the tree. Features 1-10 separate the cells into two groups, which we label 1 or 2, taking the value 0 on group 1 and 1 on group 2. Features 11-20 evenly subdivide group 1 into two further

groups (1.1 and 1.2), and features 21-30 evenly subdivide group 2 into groups 2.1 and 2.2. We continue this for four levels, yielding 32 groups with 50 cells each, for a total of 1600 cells and 3100 features (Figure 7.1). To add noise, we randomly inverted 10% of the counts[39] in the resulting matrix[40], and added random Gaussian noise with variance 0.2. This data challenges global dimensionality reduction, since the features defining each division of the hierarchy add noise to further subdivisions. A heat map of the resulting data is shown in Figure 7.1.

39: Note the noise is uncorrelated, as desired

40: that is, replaced value x by $1 - x$

We performed topological stitching using twenty locales with coverage three computed using Graph-Hopper. Ten principal, independent, or surprisal components were computed within each locale, and the downstream 15-nearest neighbor graphs were aggregated using mean random walk distance with a restart probability of 0.1. We found that the 15-nearest neighbor graphs computed using topological stitching were far better at separating the clusters, even compared to global reductions with as many as 200 components (Figure 7.1). To quantify this, we performed Leiden clustering on each graph with resolution 1.0, and assessed the adjusted mutual information (AMI) with the true clusters. Global PCA achieved a maximum AMI score of 0.755. Global SCA performed slightly better, with a maximum AMI score of 0.816. In both cases, increasing the number of global components did not improve the AMI score. Local PCA and Local SCA performed far better, with AMI scores of 0.964 and 0.998 respectively.

7.4.2 Re-analyzing the Tabula Muris data

The Tabula Muris Consortium produced a compendium of mouse single-cell data comprising over 100, 000 cells over twenty different organs [155]. To identify cell types, the authors performed PCA followed by Louvain clustering on each tissue separately, manually refined the resulting clusterings, and inferred cell types based on differentially-expressed genes[41]. Since the dataset contains a wide range of cellular environments[42], we hypothesized that local dimensionality reduction on the full dataset would allow recovery of the identified populations in a single pass, and potentially reveal novel populations.

41: Note this is different than the traditional known-marker-gene method of cell-type determination

42: The wide diversity in this atlas makes it particularly ripe for locally motivated analyses

We obtained raw counts from GEO accession GSE109774. Following the original authors' analysis, we performed transcript-per-million normalization (TPM) followed by log transformation with a pseudocount of one transcript per million. In addition, to remove variation due to donor sex, we removed five sex-specific genes[43].

43: *XIST*,*TSIX*, *DDX3Y*, *EIF2S3Y*, and *UTY*

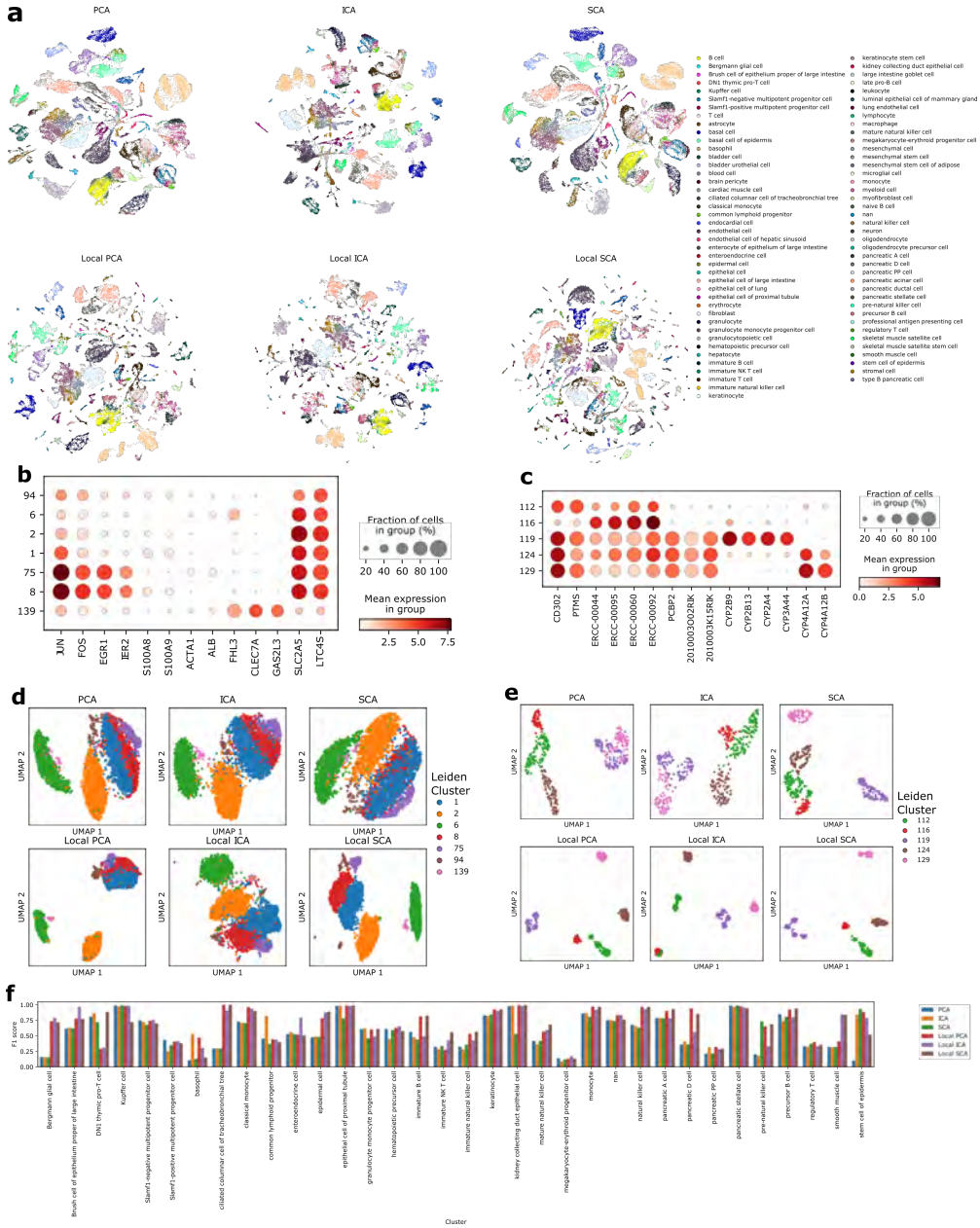

Figure 7.3: Performance of topological stitching on the Tabula Muris Consortium data. a: UMAP plots of the entire dataset downstream of PCA, ICA, SCA reductions (top), or locally-computed neighborhood graphs aggregated with topological stitching. **b**: Dotplot of differentially-expressed genes among the microglial Leiden clusters, where clusters are computed downstream of the topological stitching neighborhood graph on the whole dataset. **c**: Dotplot of differentially-expressed genes among the hepatocyte Leiden clusters. **d**: UMAP plots of microglia using neighborhoods computed globally (top), or using topological stitching (bottom), colored by Leiden clusters computed downstream of topological stitching. **e**: UMAP plots of hepatocytes using various neighborhood graphs, colored by Leiden cluster. **f**: F1 Scores for recovery of known cell types from Leiden clusters computed from different kNN-graph-generating strategies. Cell types for which all methods achieve F1 score greater than 0.8 are excluded

Using Graph-Hopper on a 50-dimensional PCA reduction, we constructed 50 locales such that each cell belonged to at least three locales. Within each locale, we computed 20-dimensional linear projections of the data using either principal component analysis (PCA) [123] or surprisal component analysis (SCA) [35], and generated 15-nearest neighbor graphs using Euclidean distance in the chosen reduction. We then aggregated these graphs using mean random walk with restart distance with restart probability 1%, producing a global 15-nearest neighbor graph. For comparison, we also computed global 15-nearest neighbor graphs using Euclidean distance in global PCA or SCA reductions with 50, 100, or 200 components. For each nearest-neighbor graph, we performed Leiden clustering with resolution 3.0, and UMAP embeddings with default parameters[44].

44: We note that the visualizations are robust across parameter choices

The UMAP embeddings suggest that topological stitching gives a more granular[45] view of the data, with stronger separation between the original authors' annotated cell types (Figure 7.3a). To quantify this, we assessed whether the cell types could be recovered from the Leiden clusters computed from each nearest-neighbors graph. For each Leiden clustering, we first split each cluster by tissue of origin to ensure that this knowledge is incorporated[46]. Then, for each cell type, we identified the set of clusters with highest overlap with that type, and computed the F1 score for detecting the cell type via the union of these clusters. This represents the accuracy with which the cell type can be detected via unsupervised clustering on the k-nearest neighbor graph, with knowledge of tissue origin. We found that topological stitching recovers many of the rarer cell types[47] where the global methods fail (Figure 7.3f).

45: That is, it is higher resolution

46: This shows, for example, that domain knowledge is important for these methods

47: for example, Bergmann glial cells, myofibroblasts, natural killer cells, and pancreatic D cells

In many cases, we recover finer-grained classifications.

Classification of TMC under topological stitching

We can summarize some of the main findings that topological stitching enables:

- Leiden cluster 139 corresponds to a small subset of microglia with specific expression of *CLEC7A*[][*-1][48]; and *GAS2L3*[49] [156, 157] (Figure 7.3b). This indicates that this cluster contains cells that are proliferating in response to an endogenous signal, perhaps to T-cells.
- Clusters 75 and 8 highly express *JUN*, *FOS*, *EGR1*, and *IER2*, indicating a role in neuronal early response [158]. Cluster 75 is distinguished from cluster 8 by the presence of actin and albumin[50] and the absence of pro-inflammatory activation

48: This is a co-stimulatory molecule which promotes activation and proliferation in response to a T-cell ligand

49: This promotes genomic stability during proliferation

50: *ACTA1* and *ALB*

markers *S100A8* and *S100A9*, suggesting these cells may be actively cycling. UMAP plots of the microglial subpopulation indicate that these clusters are poorly separated by global PCA, ICA, and SCA, but readily emerge using local SCA and topological stitching (Figure 7.3d).

- Topological stitching also more distinctly reveals hepatocyte populations (Figure 7.3e). Among the clusters newly separated are:
  - cluster 116, which lacks *CD302* and highly expresses many excision repair genes[51], suggesting damaged cells [159];
  - and Cluster 119, which specifically expresses Cytochrome P-genes[52] important to liver clearing [160];
  - and clusters 124 and 129 also express cytochrome genes, but an entirely different set than clusters 116 and 119 (Figure 7.3c).

51: *ERCC*s

52: *CYP2B9*, *CYP2B13*, and others

7.4.3 Topological stitching recovers rare immune subtypes

The human immune system contains a wide range of cell types with specialized roles in identifying and responding to various foreign antigens and other cellular abnormalities [148, 149]. Emerging therapies increasingly recruit, promote, or target the immune system to treat diseases [149, 161, 162]. Single-cell techniques have become important in understanding the immune landscape of complex diseases [163] , but subtle differences between immune subtypes challenge the technology[53], particularly in larger studies. We hypothesized that topological stitching could better detect these differences by adapting to local cellular environments.

53: Immune cells are among the most highly variable of all cell-types

We obtained quality-filtered transcript measurements of a benchmarking dataset with a variety of human cell types profiled using Smart-seq 3 from Hagemann-Jensen et al. [153]. The authors grouped the cells into broad classes using known marker genes. As with the Tabula Muris data, we performed PCA, ICA, and SCA reductions on the data with 10, 20, 50, 100, or 200 components, and downstream 15-nearest neighbor networks using the Euclidean distance. In addition, we used topological stitching to generate locally-tuned 15-nearest-neighbor networks. We used 30 locales with each cell belonging to three locales, generated local 15-nearest-neighbor networks using Euclidean distance in 10-dimensional PCA, ICA, or SCA space, and aggregated local networks using random walks with restart probability 0.1. UMAP plots downstream of each network-construction

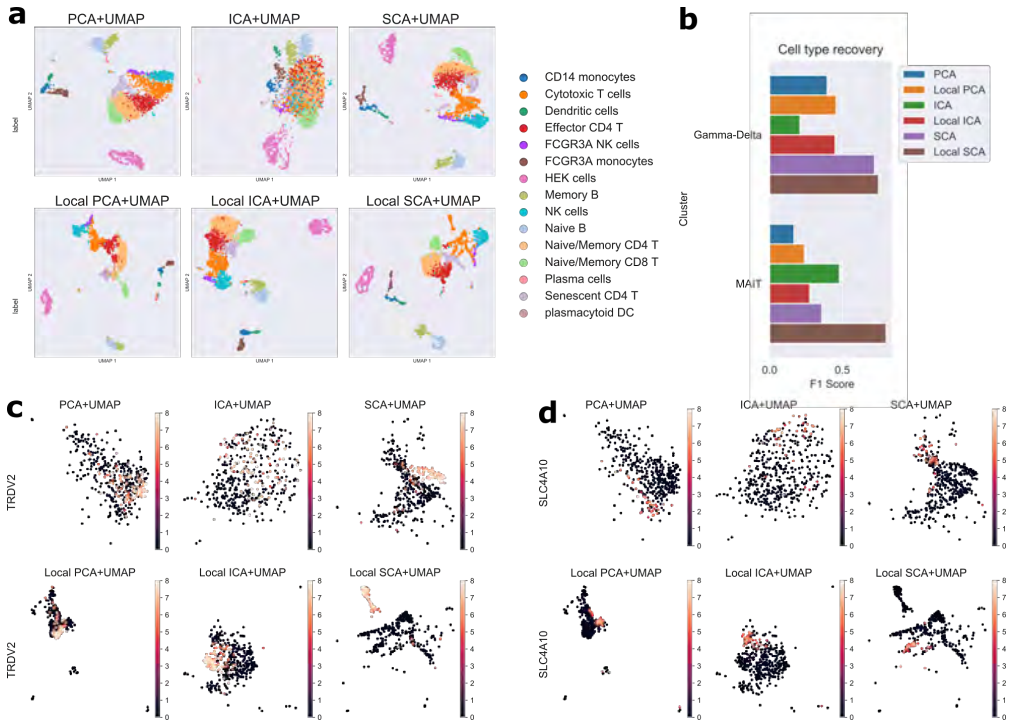

Figure 7.4: Topological stitching on cellular populations from [153]. **a**: UMAP plots downstream of PCA, ICA, SCA, and their local versions computed with topological stitching. **b**: F1 scores for recovery of the gamma-delta and MAIT populations using Leiden clusterings downstream of each method. **c**: UMAP plots of the cytotoxic T subset of the data, colored by TRDV2, a marker gene for gamma-delta T-cells. **d**: UMAP plots of the cytotoxic T subset colored by SLC4A10, a MAIT marker gene.

strategy are shown in Figure 7.4a. Here, global reductions are 100-dimensional[54].

Overall, we found that topological stitching improved visual separation[55] between annotated cell types compared to global reductions. In topological stitching using SCA, the cytotoxic T-cell population appeared to have finer substructures. Examining this population more closely, we found that these corresponded to known populations: gamma-delta T-cells[56]; and MAIT cells[57] (Figure 7.4c,d). To quantify this improvement, we performed Leiden clusterings from each k-NN network with resolution 2.0, assessed whether these cell types could be recovered as unions of Leiden clusters. We defined gamma-delta T-cells as those expressing at least two of *TRDV2*, *TRGV9*, and *TRDC*; and MAIT cells as those expressing at least one of *SLC4A10* and *LTK*. topological stitching with SCA followed by Leiden clustering recovered these two populations with high F1 score (Figure 7.4b).

54: But again, results are robust across these choices

55: in UMAP

56: marked by the delta T-receptor *TRDV2*

57: marked by *SLC4A10*

8 Discussion and Conclusions

In a famous story that is sadly apocryphal but nevertheless conveys the zeitgeist of the field, Lord Kelvin declared in 1900 that, "There is nothing new to be discovered in physics now". The success of Lagrangian and Hamiltonian mechanics, and the development of the statistical theory of thermodynamics had together seemed to ably describe all forms of dynamics in the natural world. Kelvin noted in passing the two "clouds on the horizon" — which, of course, became quantum mechanics and relativity, both of which absolutely revolutionized the study of physics in the twentieth century. In some ways, the world of single-cell RNA-sequencing, the primary focus of this dissertation, feels as though it has arrived at a similar crossroads[1].

1: I admit that the comparison is hyperbole — few people claim that scRNA-seq has been solved in any fundamental way

Attention is moving towards more and more complex data types — multimodal sequencing, which adds ever more types of data to the picture; different types of *-omics*, like transcription factor binding sites or chromatin accessibility; perturbation testing; and of course, larger and larger datasets. While the questions posed by these new and exciting methodologies are important in their own right, the goal of this dissertation has been to show that the very fundamental task of understanding what makes two cells biologically similar given their expression profiles is still open — and indeed remains both algorithmically and biologically interesting.

While the work presented here is not restricted to RNA, the theme of focusing on distance and distortions of that distance runs throughout. In this chapter, our aim is precisely to tie the research presented together and to draw some broad conclusions, not only about the algorithms but also about how one can think about metrics in biology more generally.

8.1 Summary of Work

As with any good progression of work, our first project focuses squarely on linear transformations[2]. In **Schema**, presented in Chapter 3, we stuck with Euclidean distance — modifying it with just a scaling transform to massage the original coordinate vectors to integrate other types of information. In other words, the goal was to work *within* the confines of Euclidean distance, taking advantage of the structure that that metric imbues on the space.

2: Indeed, the power of linear methods in high-dimensions is often underrated

The flexibility of Schema is a great advantage — in Chapter 5 we see how the simple framework can be cleverly utilized to answer facially quite complex questions. However, the limitations of Euclidean distance especially at large scales were already apparent, which motivated the push into understanding manifold learning algorithms, where the focus is on local distances only, at the expense of large-scale distances[3].

3: We note here that there are possible extensions to Schema, where we can use *kernel distances* and focus on particular bandwidths of distance. This is a thread we are eager to pursue

This is the assumption underlying the manifold hypothesis, which we discussed in detail in Section 2.5. Algorithms for visualization and dimensionality-reduction have shown remarkable success empirically in the scRNA-seq world by relying on this hypothesis. We also note that, while there often a tradeoff between runtime and complexity, the manifold hypothesis actually often gives a runtime advantage: rather than needing to pay attention to all $O(N^2)$ pairwise distances, we only need $O(kN)$ distances (where k is the number of neighbors that make up the local scale)[4].

4: We should note that *computing* the k-NN graph, for example, can be slow and naïvely quadratic; in fact, some of the main algorithmic developments in, for example, t-SNE and UMAP are in quickly computing the neighbors graphs

Our first effort to contend with the manifold hypothesis concerned the now ubiquitous visualization algorithms t-SNE and UMAP. Having analyzed the objective functions of these algorithms, we found that, in their attempt to focus on local distances and preserving neighborhood structures, they end up washing out information about the density of points in high-dimensional space. Our work in diagnosing and improving upon this shortcoming resulted in the **densVis** package — we showed that preserving density information leads to interesting insights about variability of expression. The idea of considering variance as an important feature of gene expression is not new, but our work shows it as a useful mode of analysis for a wide range of datasets — from immune cells in tumor to differentiation during embryo development.

While our presentation in Chapter 4 follows the traditional paradigm of exposition in mathematics, where problems and solutions descend from the heavens, it is instructive to pare back the process by which

developed our method a little, as it demonstrates the way that opaque, nonlinear methods can be interrogated.

It was while experimenting with various synthetic models where we *knew* what the underlying structure should be — in fact, we often used two-dimensional input data itself, so the input and output scatterplots could themselves be compared — that we discovered the lack of density-preservation in state-of-the-art algorithms.

While attempting to understand differentiating cells, which we modeled as a Gaussian point-cloud with increasing variance, we found that the representations did not look any different for differing variances. From there, digging into the objective function revealed that density is not preserved — almost by design. But by fiddling with the adaptive length-scale term, its importance became clear, motivating the consideration of *multiple characteristic length-scales* that seem present by *default* in these high-dimensional datasets.

Walking through this is not an exercise in exhausting the reader, but rather an example of a broader set of principles for analysis.

For one, it is crucial to think about exactly what metrics a dimensionality-reduction[5] algorithm can preserve — because *some* information must necessarily be lost. Our use of ground truth data allows us to better understand exactly what properties are lost under a transformation.

5: or, really, any type of transformation

But finding a metric on which an algorithm does poorly, while mathematically interesting, only becomes *biologically* interesting when the metric can be shown to have a biological analogue. Our ground truth modeling of differentiation, indeed, seemed to indicate that density is worth considering from a biological perspective, in that it reflects some notion of variability.

While our work with t-SNE and UMAP really centered the importance of k-NN graphs, in moving to **topological stitching** (Chapter 7), we wanted to interrogate even those local distances. Since we were already discounting the meaningfulness of non-local distances, the question of whether even those *nearest* distances are correct.

Our topological stitching algorithm thus asks exactly what makes even a local distance accurate. One of the key drivers of this question is whether local distance also can be corrupted by noise — can the presence of a noisy gene preclude a cell from finding its correct nearest neighbors, a scenario that is again motivated by a simulated dataset (see Figure 8.1) where a feature is a marker for one class but very noisy for another.

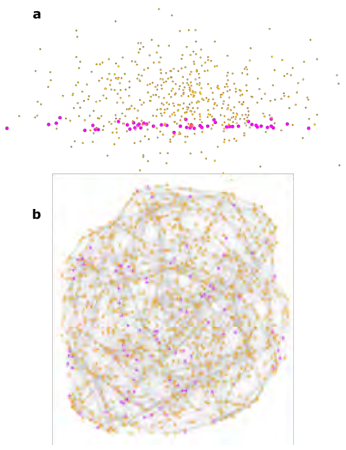

Figure 8.1: Noise confounds nearest neighbors. In a dataset where two clusters are separated by variance in the x-dimension (**a**), most of the neighbors of magenta points are orange (**b**).

Our work in using *local* feature selection also improves on a structure (the k-NN graph) that is so foundational for other tools, and so we hope that our consideration of distance metrics is impactful across the field.

8.2 Outlook

We contend that our work, while answering some questions of its own, more importantly opens up further questions in the craft of analyzing single-cell RNA-seq datasets. In fact, the notion of cell similarity, and in general that of generative models for scRNA-seq data might play the part of Lord Kelvin's "clouds" on the horizon.

6: while I proceed to indeed overwork the analogy

Not to overwork the analogy[6], but these "clouds" have only become clouds because of the massive increase in size and resolution that these scRNA-seq datasets have now achieved — just as classical physics was a fine description of the world until the resolution of instruments became strong enough that classical dynamics could be observably wrong.

Bibliography

[1] Hyunghoon Cho, Bonnie Berger, and Jian Peng. 'Neural Data Visualization for Scalable and Generalizable Single Cell Analysis'. In: *bioRxiv* (2018). DOI: 10.1101/289223 (cited on page 17).

[2] Florian Markowetz. 'All biology is computational biology'. In: *PLOS Biology* 15.3 (Mar. 2017), e2002050. DOI: 10.1371/JOURNAL.PBIO.2002050 (cited on page 18).

[3] Angela Saini. *Superior: the return of race science*. Beacon Press, 2019 (cited on page 18).

[4] François Jacob and Jacques Monod. 'On the Regulation of Gene Activity'. In: *Cold Spring Harbor Symposia on Quantitative Biology* 26.0 (Jan. 1961), pp. 193–211. DOI: 10.1101/SQB.1961.026.01.024 (cited on page 18).

[5] Paulien Hogeweg. 'The Roots of Bioinformatics in Theoretical Biology'. In: *PLOS Computational Biology* 7.3 (Mar. 2011), e1002021. DOI: 10.1371/JOURNAL.PCBI.1002021 (cited on page 18).

[6] Sandy L. Klemm, Zohar Shipony, and William J. Greenleaf. 'Chromatin accessibility and the regulatory epigenome'. In: *Nature reviews. Genetics* 20.4 (Apr. 2019), pp. 207–220. DOI: 10.1038/S41576-018-0089-8 (cited on page 24).

[7] Ian Dunham et al. 'An integrated encyclopedia of DNA elements in the human genome'. In: *Nature 2012 489:7414* 489.7414 (Sept. 2012), pp. 57–74. DOI: 10.1038/nature11247 (cited on page 24).

[8] E. A. Feingold et al. 'The ENCODE (ENCyclopedia Of DNA Elements) Project'. In: *Science* 306.5696 (Oct. 2004), pp. 636–640. DOI: 10.1126/science.1105136 (cited on page 25).

[9] Gregory M. Parkes and Mahesan Niranjan. 'Uncovering extensive post-translation regulation during human cell cycle progression by integrative multi-'omics analysis'. In: *BMC Bioinformatics* 20.1 (Oct. 2019), pp. 1–13. DOI: 10.1186/S12859-019-3150-5/FIGURES/5 (cited on pages 25, 26).

[10] Kui Wang, Canhua Huang, and Edouard Nice. 'Recent advances in proteomics: towards the human proteome'. In: *Biomed Chromatogr* 28.6 (2014), pp. 848–57. DOI: 10.1002/bmc.3157 (cited on page 25).

[11] Anne Claude Gingras, Ruedi Aebersold, and Brian Raught. 'Advances in protein complex analysis using mass spectrometry'. In: *The Journal of Physiology* 563.1 (Feb. 2005), pp. 11–21. DOI: 10.1113/JPHYSIOL.2004.080440 (cited on page 25).

[12] Taras Makhnevych and Walid A. Houry. 'The role of Hsp90 in protein complex assembly'. In: *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* 1823.3 (Mar. 2012), pp. 674–682. DOI: 10.1016/J.BBAMCR.2011.09.001 (cited on page 25).

[13] Siddhartha Roy and Tapas K. Kundu. 'An integrative view of chemical biology'. In: *Chemical Biology of the Genome*. 2021. DOI: 10.1016/b978-0-12-817644-3.00004-0 (cited on page 25).

[14] A. E. Bond, P. E. Row, and E. Dudley. 'Post-translation modification of proteins; methodologies and applications in plant sciences'. In: *Phytochemistry* 72.10 (July 2011), pp. 975–996. DOI: 10.1016/J.PHYTOCHEM.2011.01.029 (cited on page 25).

[15] 10x Genomics. *A New Way of Exploring Immunity - Linking Highly Multiplexed Antigen Recognition to Immune Repertoire and Phenotype*. Tech. rep. 2019 (cited on pages 26, 55, 56, 72, 114–116, 124).

[16] Jonathan S. Packer et al. 'A lineage-resolved molecular atlas of C. Elegans embryogenesis at single-cell resolution'. In: *Science* 365.6459 (Sept. 2019). DOI: 10.1126/SCIENCE.AAX1971/SUPPL{_}FILE/AAX1971{_}TABLES{_}S7{_}S8{_}S10{_}S11{_}S14.ZIP (cited on pages 26, 119, 128, 129, 134, 136).

[17] Rapolas Zilionis et al. 'Single-Cell Transcriptomics of Human and Mouse Lung Cancers Reveals Conserved Myeloid Populations across Individuals and Species'. In: *Immunity* 50.5 (May 2019), pp. 1317–1334. DOI: 10.1016/j.immuni.2019.03.009 (cited on pages 26, 119, 122, 134, 136).

[18] Elaine R. Mardis. 'ChIP-seq: welcome to the new frontier'. In: *Nature Methods 2007 4:8* 4.8 (Aug. 2007), pp. 613–614. DOI: 10.1038/nmeth0807-613 (cited on page 26).

[19] Brian Hie et al. 'Computational Methods for Single-Cell RNA Sequencing'. In: *https://doi.org/10.1146/annurev-biodatasci-012220-100601* 3.1 (July 2020), pp. 339–364. DOI: 10.1146/ANNUREV-BIODATASCI-012220-100601 (cited on pages 27, 55, 75, 76).

[20] Larry Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer, 2013 (cited on page 28).

[21] Geoffrey Schiebinger et al. 'Reconstruction of developmental landscapes by optimal-transport analysis of single-cell gene expression sheds light on cellular reprogramming'. In: (2017). DOI: 10.1101/191056 (cited on pages 28, 103).

[22] Laleh Haghverdi et al. 'Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors'. In: *Nature Biotechnology* 36.5 (2018). DOI: 10.1038/nbt.4091 (cited on pages 28, 50, 56).

[23] Brian Hie, Bryan Bryson, and Bonnie Berger. 'Efficient integration of heterogeneous single-cell transcriptomes using Scanorama'. In: *Nature Biotechnology 2019 37:6* 37.6 (May 2019), pp. 685–691. DOI: 10.1038/s41587-019-0113-3 (cited on pages 28, 50, 56, 103).

[24] Kamran Khan et al. 'DBSCAN: Past, present and future'. In: *5th International Conference on the Applications of Digital Information and Web Technologies, ICADIWT 2014* (2014), pp. 232–238. DOI: 10.1109/ICADIWT.2014.6814687 (cited on page 32).

[25] Khaled Alsabti, Sanjay Ranka, and Vineet Singh. 'An efficient k-means clustering algorithm'. In: *Electrical Engineering and Computer Science* (Jan. 1997) (cited on page 32).

[26] Xiaofei He et al. 'Laplacian regularized Gaussian mixture model for data clustering'. In: *IEEE Transactions on Knowledge and Data Engineering* 23.9 (2011), pp. 1406–1418. DOI: 10.1109/TKDE.2010.259 (cited on page 32).

[27] Xiao Zhang, Yun Liao, and Shizhong Liao. 'A survey on online kernel selection for online kernel learning'. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9.2 (Mar. 2019), e1295. DOI: 10.1002/WIDM.1295 (cited on page 37).

[28] Jacob Goldberger et al. 'Neighbourhood Components Analysis'. In: *Advances in Neural Information Processing Systems* 17 (2004) (cited on pages 37, 38).

[29] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. 'Distance Metric Learning for Large Margin Nearest Neighbor Classification'. In: *Advances in Neural Information Processing Systems* 18 (2005) (cited on page 38).

[30] Y. William Yu et al. 'Entropy-Scaling Search of Massive Biological Data'. In: *Cell Systems* 1.2 (Aug. 2015), pp. 130–140. DOI: 10.1016/J.CELS.2015.08.004 (cited on pages 39, 68).

[31] David Harel and Yehuda Koren. 'A fast multi-scale method for drawing large graphs'. In: *International Symposium on Graph Drawing*. Heidelberg: Springer, 2000, pp. 183–196 (cited on pages 40, 102).

[32] Ronald R. Coifman and Stéphane Lafon. 'Diffusion maps'. In: *Applied and Computational Harmonic Analysis* 21.1 (July 2006), pp. 5–30. DOI: 10.1016/J.ACHA.2006.04.006 (cited on pages 41, 47).

[33] William Johnson and Joram Lindenstrauss. 'Extension of Lipschitz maps into a Hilbert Space'. In: *Contemporary Mathematics* 26 (1986) (cited on page 43).

[34] Kasper Green Larsen and Jelani Nelson. 'Optimality of the johnson-lindenstrauss lemma'. In: *Annual Symposium on Foundations of Computer Science - Proceedings* 2017-October (Nov. 2017), pp. 633–638. DOI: 10.1109/FOCS.2017.64 (cited on page 44).

[35] Benjamin DeMeo and Bonnie Berger. 'Discovering Rare Cell Types through Information-based Dimensionality Reduction'. In: *bioRxiv* (Oct. 2021), p. 2021.01.19.427303. DOI: 10.1101/2021.01.19.427303 (cited on pages 45, 50, 142, 144, 154).

[36] Mikhail Belkin and Partha Niyogi. 'Laplacian eigenmaps for dimensionality reduction and data representation'. In: *Neural Computation* 15.6 (June 2003), pp. 1373–1396. DOI: 10.1162/089976603321780317 (cited on page 47).

[37] Jihun Ham et al. 'A kernel view of the dimensionality reduction of manifolds'. In: *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004* (2004), pp. 369–376. DOI: 10.1145/1015330.1015417 (cited on page 47).

[38] Laurens van der Maaten and Geoffrey Hinton. 'Visualizing Data using t-SNE'. In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605 (cited on pages 47, 75, 79, 80, 139, 187).

[39] Leland McInnes and John Healy. 'UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction'. In: *arXiv* (2018). DOI: 10.21105/joss.00861 (cited on pages 47, 59, 75, 79, 80, 109, 139, 188).

[40] Thalia E. Chan, Michael P.H. Stumpf, and Ann C. Babtie. 'Gene Regulatory Network Inference from Single-Cell Data Using Multivariate Information Measures'. In: *Cell Systems* 5.3 (Sept. 2017), pp. 251–267. DOI: 10.1016/J.CELS.2017.08.014 (cited on page 50).

[41] Florian Buettner et al. 'Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells'. In: *Nature Biotechnology* 33.2 (2015), pp. 155–160. DOI: 10.1038/NBT.3102 (cited on page 50).

[42] Alexander P. Wu et al. 'Bayesian information sharing enhances detection of regulatory associations in rare cell types'. In: *Bioinformatics* 37.Supplement_1 (July 2021), pp. i349–i357. DOI: 10.1093/BIOINFORMATICS/BTAB269 (cited on pages 50, 143).

[43] Rongxin Fang et al. 'Comprehensive analysis of single cell ATAC-seq data with SnapATAC'. In: *Nature Communications 2021 12:1* 12.1 (Feb. 2021), pp. 1–15. DOI: 10.1038/s41467-021-21583-9 (cited on page 50).

[44] Brian Hie et al. 'Geometric Sketching Compactly Summarizes the Single-Cell Transcriptomic Landscape'. In: *Cell Systems* 8.6 (June 2019), pp. 483–493. DOI: 10.1016/J.CELS.2019.05.003 (cited on pages 50, 117, 138, 143).

[45] Lan Jiang et al. 'GiniClust: Detecting rare cell types from single-cell gene expression data with Gini index'. In: *Genome Biology* 17.1 (July 2016), pp. 1–13. DOI: 10.1186/S13059-016-1010-4/FIGURES/6 (cited on page 50).

[46] Benjamin DeMeo and Bonnie Berger. 'Hopper: a mathematically optimal algorithm for sketching biological data'. In: *Bioinformatics* 36.Supplement_1 (July 2020), pp. i236–i241. DOI: 10.1093/BIOINFORMATICS/BTAA408 (cited on pages 50, 117, 138, 143).

[47] Satwik Rajaram et al. 'Sampling strategies to capture single-cell heterogeneity'. In: *Nature Methods 2017 14:10* 14.10 (Sept. 2017), pp. 967–970. DOI: 10.1038/nmeth.4427 (cited on page 50).

[48] Daniel B. Burkhardt et al. 'Quantifying the effect of experimental perturbations at single-cell resolution'. In: *Nature Biotechnology 2021 39:5* 39.5 (Feb. 2021), pp. 619–629. DOI: 10.1038/s41587-020-00803-5 (cited on page 50).

[49] Krzysztof Polański et al. 'BBKNN: fast batch alignment of single cell transcriptomes'. In: *Bioinformatics* 36.3 (Feb. 2020), pp. 964–965. DOI: 10.1093/BIOINFORMATICS/BTZ625 (cited on page 50).

[50] J. Javier Diaz-Mejia et al. 'Evaluation of methods to assign cell type labels to cell clusters from single-cell RNA-sequencing data'. In: *F1000Research* 8 (Oct. 2019), p. 296. DOI: 10.12688/F1000RESEARCH.18490.3 (cited on page 51).

[51] Tara Chari, Joeyta Banerjee, and Lior Pachter. 'The Specious Art of Single-Cell Genomics'. In: *bioRxiv* (Sept. 2021), p. 2021.08.25.457696. DOI: 10.1101/2021.08.25.457696 (cited on pages 51, 52, 110, 139).

[52] Etienne Becht et al. 'Dimensionality reduction for visualizing single-cell data using UMAP'. In: *Nature biotechnology* 37.1 (2019), p. 38 (cited on pages 51, 52, 75, 90, 91, 120, 183).

[53] Rohit Singh et al. 'SCHEMA: A general framework for integrating heterogeneous single-cell modalities'. In: *bioRxiv* (Nov. 2019), p. 834549. DOI: 10.1101/834549 (cited on page 55).

[54] Rohit Singh et al. 'Schema: metric learning enables interpretable synthesis of heterogeneous single-cell modalities'. In: *Genome Biology* 22.1 (Dec. 2021), pp. 1–24. DOI: 10.1186/S13059-021-02313-2/FIGURES/5 (cited on pages 55, 105).

[55] Bosiljka Tasic et al. 'Shared and distinct transcriptomic cell types across neocortical areas'. In: *Nature* 563.7729 (Nov. 2018), pp. 72–78. DOI: 10.1038/s41586-018-0654-5 (cited on page 55).

[56] Kristofer Davie et al. 'A Single-Cell Transcriptome Atlas of the Aging Drosophila Brain'. In: *Cell* 174.4 (2018). DOI: 10.1016/j.cell.2018.05.057 (cited on pages 55, 109).

[57] Xiao Dong et al. 'Accurate identification of single-nucleotide variants in whole-genome-amplified single cells'. In: *Nature Methods* 14 (2017), pp. 491–493. DOI: 10.1038/nmeth.4227 (cited on page 55).

[58] Junyue Cao et al. 'Joint profiling of chromatin accessibility and gene expression in thousands of single cells.' In: *Science* 361.6409 (2018), pp. 1380–1385. DOI: 10.1126/science.aau0730 (cited on pages 55, 56, 105, 106, 117).

[59] Ino D. Karemaker and Michiel Vermeulen. 'Single-Cell DNA Methylation Profiling: Technologies and Biological Applications'. In: *Trends in Biotechnology* 36.9 (2018), pp. 952–965. DOI: 10.1016/j.tibtech.2018.04.002 (cited on page 56).

[60] Assaf Rotem et al. 'Single-cell ChIP-seq reveals cell subpopulations defined by chromatin state'. In: *Nature Biotechnology* 33 (2015), pp. 1165–1172. DOI: 10.1038/nbt.3383 (cited on page 56).

[61] Marlon Stoeckius et al. 'Simultaneous epitope and transcriptome measurement in single cells'. In: *Nature Methods* 14 (2017), pp. 865–868. DOI: 10.1038/nmeth.4380 (cited on page 56).

[62] Samuel G Rodriques et al. 'Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution'. In: *Science* 363.6434 (2019), pp. 1463–1467 (cited on pages 56, 71, 111–113).

[63] Vladimir Yu Kiselev, Andrew Yiu, and Martin Hemberg. 'scmap: Projection of single-cell RNA-seq data across data sets'. In: *Nature Methods* 15 (2018), pp. 359–362. DOI: 10.1038/nmeth.4644 (cited on page 56).

[64] Nikolas Barkas et al. 'Joint analysis of heterogeneous single-cell RNA-seq dataset collections'. In: *Nature Methods* 16 (2019), pp. 695–698. DOI: 10.1038/s41592-019-0466-z (cited on page 56).

[65] Ilya Korsunsky et al. 'Fast, sensitive, and accurate integration of single cell data with Harmony'. In: *BioRxiv* (2018). DOI: 10.1101/461954 (cited on page 56).

[66] Tim Stuart and Rahul Satija. 'Integrative single-cell analysis'. In: *Nature Reviews Genetics* 20.5 (May 2019), pp. 257–272. DOI: 10.1038/s41576-019-0093-7 (cited on page 56).

[67] Joshua D. Welch et al. 'Single-Cell Multi-omic Integration Compares and Contrasts Features of Brain Cell Identity'. In: *Cell* 177.7 (June 2019), pp. 1873–1887. DOI: 10.1016/J.CELL.2019.05.006 (cited on pages 56, 72, 73).

[68] Valentine Svensson, Sarah A. Teichmann, and Oliver Stegle. 'SpatialDE: Identification of spatially variable genes'. In: *Nature Methods* 15.5 (2018). DOI: 10.1038/nmeth.4636 (cited on pages 56, 113).

[69] Daniel Edsgärd, Per Johnsson, and Rickard Sandberg. 'Identification of spatial expression trends in single-cell gene expression data'. In: *Nature Methods* 15.5 (2018). DOI: 10.1038/nmeth.4634 (cited on pages 56, 113).

[70] Shiquan Sun, Jiaqiang Zhu, and Xiang Zhou. 'Statistical analysis of spatial expression patterns for spatially resolved transcriptomic studies'. In: *Nature Methods* 17.2 (2020). DOI: 10.1038/s41592-019-0701-7 (cited on page 56).

[71] David DeTomaso and Nir Yosef. 'Hotspot identifies informative gene modules across modalities of single-cell genomics'. In: *Cell Systems* 12.5 (2021). DOI: 10.1016/j.cels.2021.04.005 (cited on page 56).

[72] Ricard Argelaguet et al. 'MOFA+: A statistical framework for comprehensive integration of multi-modal single-cell data'. In: *Genome Biology* 21.1 (2020). DOI: 10.1186/s13059-020-02015-1 (cited on pages 56, 117, 191–193).

[73] Jacob Goldberger et al. 'Neighbourhood Components Analysis'. In: *Advances in Neural Information Processing Systems*. 2005, pp. 513–520 (cited on pages 57, 70, 71).

[74] Jason V. Davis et al. 'Information-theoretic metric learning'. In: *Proceedings of the 24th international conference on Machine learning - ICML '07*. New York, New York, USA: ACM Press, 2007, pp. 209–216. DOI: 10.1145/1273496.1273523 (cited on pages 57, 70, 71).

[75] Kilian Q Weinberger and Lawrence K Saul. *Distance Metric Learning for Large Margin Nearest Neighbor Classification*. Tech. rep. 2009, pp. 207–244 (cited on pages 57, 70, 71).

[76] Eric P Xing et al. 'Distance Metric Learning, with Application to Clustering with Side-Information'. In: *Advances in Neural Information Processing Systems*. 2002 (cited on pages 57, 70).

[77] Romain Lopez et al. 'Deep generative modeling for single-cell transcriptomics'. In: *Nature Methods* 15.12 (2018). DOI: 10.1038/s41592-018-0229-2 (cited on pages 58, 59, 108).

[78] Gökcen Eraslan et al. 'Single-cell RNA-seq denoising using a deep count autoencoder'. In: *Nature Communications* 10.1 (2019). DOI: 10.1038/s41467-018-07931-2 (cited on pages 58, 59, 108).

[79] Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. 'scGen predicts single-cell perturbation responses'. In: *Nature Methods* 16.8 (2019). DOI: 10.1038/s41592-019-0494-8 (cited on pages 58, 59, 108).

[80] Adam Gayoso et al. 'Joint probabilistic modeling of paired transcriptome and proteome measurements in single cells'. In: *bioRxiv* (2020). DOI: 10.1101/2020.05.08.083337 (cited on pages 58, 59, 108).

[81] Mike Wu and Noah Goodman. 'Multimodal generative models for scalable weakly-supervised learning'. In: *Advances in Neural Information Processing Systems*. Vol. 2018-December. 2018 (cited on pages 59, 108).

[82] Yuge Shi et al. 'Variational mixture-of-experts autoencoders for multi-modal deep generative models'. In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019 (cited on pages 59, 108).

[83] Richard Kurle, Stephan Günnemann, and Patrick van der Smagt. 'Multi-source neural variational inference'. In: *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*. 2019. DOI: 10.1609/aaai.v33i01.33014114 (cited on pages 59, 108).

[84] Laurens van der Maaten. *Accelerating t-SNE using Tree-Based Algorithms*. Tech. rep. 2014, pp. 1–21 (cited on page 59).

[85] Debajyoti Sinha et al. 'Dropclust: Efficient clustering of ultra-large scRNA-seq data'. In: *Nucleic Acids Research* 46.6 (2018). DOI: 10.1093/nar/gky007 (cited on page 59).

[86] Tim Stuart et al. 'Comprehensive Integration of Single-Cell Data'. In: *Cell* 177.7 (June 2019), pp. 1888–1902. DOI: 10.1016/J.CELL.2019.05.031 (cited on pages 60, 72, 73).

[87] William De Vazelhes et al. 'Metric-learn: Metric Learning Algorithms in Python'. In: *Journal of Machine Learning Research* 21 (2020) (cited on page 71).

[88] Kilian Q. Weinberger and Gerald Tesauro. 'Metric learning for kernel regression'. In: *Journal of Machine Learning Research*. Vol. 2. 2007 (cited on page 71).

[89] Masashi Sugiyama. 'Local fisher discriminant analysis for supervised dimensionality reduction'. In: *ACM International Conference Proceeding Series*. Vol. 148. 2006. DOI: 10.1145/1143844.1143958 (cited on page 71).

[90] Ricard Argelaguet et al. 'Multi-omics profiling of mouse gastrulation at single-cell resolution'. In: *Nature* 576.7787 (2019). DOI: 10.1038/s41586-019-1825-8 (cited on pages 72, 191, 192).

[91] Ashwin Narayan, Bonnie Berger, and Hyunghoon Cho. 'Assessing single-cell transcriptomic variability through density-preserving data visualization'. In: *Nature Biotechnology 2021 39:6* 39.6 (Jan. 2021), pp. 765–774. DOI: 10.1038/s41587-020-00801-7 (cited on pages 75, 119).

[92] Geng Chen, Baitang Ning, and Tieliu Shi. 'Single-Cell RNA-Seq Technologies and Related Computational Data Analysis'. In: *Frontiers in Genetics* 10 (2019), p. 317. DOI: 10.3389/fgene.2019.00317 (cited on pages 75, 76).

[93] El-ad David Amir et al. 'viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia'. In: *Nature Biotechnology* 31.6 (2013), pp. 545–552. DOI: 10.1038/nbt.2594 (cited on page 75).

[94] Andrew Gelman et al. *Bayesian Data Analysis*. 3rd ed. Boca Raton: CRC press, 2013, p. 69 (cited on page 84).

[95] Harold Hotelling. 'Relations Between Two Sets of Variates'. In: *Biometrika* 28.3/4 (Apr. 1936), pp. 321–377. DOI: 10.2307/2333955 (cited on page 85).

[96] Galen Andrew et al. *Deep Canonical Correlation Analysis*. Tech. rep. 2013 (cited on page 85).

[97] Dmitry Kobak and Philipp Berens. 'The art of using t-SNE for single-cell transcriptomics'. In: *bioRxiv* (May 2019), p. 453449. DOI: 10.1101/453449 (cited on page 88).

[98] C G Healey and J T Enns. 'Building perceptual textures to visualize multidimensional datasets'. In: *Proceedings Visualization '98 (Cat. No.98CB36276)*. 1998, pp. 111–118. DOI: 10.1109/VISUAL.1998.745292 (cited on page 89).

[99] C G Healey and J T Enns. 'Large datasets at a glance: combining textures and colors in scientific visualization'. In: *IEEE Transactions on Visualization and Computer Graphics* 5.2 (1999), pp. 145–167. DOI: 10.1109/2945.773807 (cited on pages 89, 120).

[100] Max Vladymyrov and Migueí A Carreira-Perpiñán. *Entropic Affinities: Properties and Efficient Numerical Computation*. Tech. rep. 2013 (cited on pages 92, 100).

[101] Hyunghoon Cho, Bonnie Berger, and Jian Peng. 'Generalizable and Scalable Visualization of Single-Cell Data Using Neural Networks'. In: *Cell Systems* (2018). DOI: 10.1016/j.cels.2018.05.017 (cited on page 102).

[102] George C. Linderman et al. 'Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data'. In: *Nature Methods* 16.3 (Mar. 2019), pp. 243–245. DOI: 10.1038/s41592-018-0308-4 (cited on pages 102, 134).

[103] Peter Eades. 'A Heuristic for Graph Drawing'. In: *Congressus Numerantium* 42 (1984), pp. 149–160 (cited on page 102).

[104] Camden Jansen et al. 'Building gene regulatory networks from scATAC-seq and scRNA-seq using Linked Self Organizing Maps'. In: *PLOS Computational Biology* 15.11 (2019), e1006555. DOI: 10.1371/JOURNAL.PCBI.1006555 (cited on page 102).

[105] Hang Dai and Yongtao Guan. 'Nubeam-dedup: a fast and RAM-efficient tool to de-duplicate sequencing reads without mapping'. In: *Bioinformatics* 36.10 (May 2020), pp. 3254–3256. DOI: 10.1093/BIOINFORMATICS/BTAA112 (cited on page 102).

[106] Nils Eling et al. 'Correcting the Mean-Variance Dependency for Differential Variability Testing Using Single-Cell RNA Sequencing Data'. In: *Cell Systems* 7.3 (Sept. 2018), pp. 284–294. DOI: 10.1016/j.cels.2018.06.011 (cited on pages 102, 124).

[107] Graciela M. Castex. 'Frames of Reference: The Effects of Ethnocentric Map Projections on Professional Practice'. In: *Social Work* 38.6 (1993), pp. 685–93 (cited on page 103).

[108] Kenneth W. Haemer. 'Area Bias in Map Presentation'. In: *The American Statistician* 3.2 (1949), p. 19 (cited on page 103).

[109] Vladimir Yu Kiselev, Tallulah S. Andrews, and Martin Hemberg. 'Challenges in unsupervised clustering of single-cell RNA-seq data'. In: *Nature Reviews Genetics 2018 20:5* 20.5 (Jan. 2019), pp. 273–282. DOI: 10.1038/s41576-018-0088-9 (cited on page 103).

[110] V. A. Traag, L. Waltman, and N. J. van Eck. 'From Louvain to Leiden: guaranteeing well-connected communities'. In: *Scientific Reports* 9.1 (2019). DOI: 10.1038/s41598-019-41695-z (cited on pages 106, 139).

[111] Amanda Mather and Carol Pollock. *Glucose handling by the kidney*. 2011. DOI: 10.1038/ki.2010.509 (cited on page 107).

[112] W. M. Gelbart et al. *FlyBase: The Drosophila database*. 1996. DOI: 10.1093/nar/24.1.53 (cited on page 110).

[113] Arpiar Saunders et al. 'Molecular Diversity and Specializations among the Cells of the Adult Mouse Brain'. In: *Cell* 174.4 (2018). DOI: 10.1016/j.cell.2018.07.028 (cited on page 111).

[114] Antonio Fabregat et al. 'The Reactome Pathway Knowledgebase'. In: *Nucleic Acids Research* 46.D1 (2018). DOI: 10.1093/nar/gkx1132 (cited on pages 111, 197).

[115] Nishant K. Singh et al. 'Emerging Concepts in TCR Specificity: Rationalizing and (Maybe) Predicting Outcomes'. In: *The Journal of Immunology* 199.7 (2017), pp. 2203–2213. DOI: 10.4049/jimmunol.1700744 (cited on page 114).

[116] Pradyot Dash et al. 'Quantifiable predictive features define epitope-specific T cell receptor repertoires.' In: *Nature* 547.7661 (2017), pp. 89–93. DOI: 10.1038/nature22383 (cited on page 114).

[117] Neerja Thakkar and Chris Bailey-Kellogg. 'Balancing sensitivity and specificity in distinguishing TCR groups by CDR sequence similarity'. In: *BMC Bioinformatics* 20.1 (Dec. 2019), p. 241. DOI: 10.1186/s12859-019-2864-8 (cited on page 114).

[118] Mikhail Shugay et al. 'VDJdb: A curated database of T-cell receptor sequences with known antigen specificity'. In: *Nucleic Acids Research* 46.D1 (2018). DOI: 10.1093/nar/gkx760 (cited on page 115).

[119] Anand Murugan et al. 'Statistical inference of the generation probability of T-cell receptors from sequence repertoires'. In: *Proceedings of the National Academy of Sciences of the United States of America* 109.40 (2012). DOI: 10.1073/pnas.1212755109 (cited on page 115).

[120] Blanca Pijuan-Sala et al. 'A single-cell molecular map of mouse gastrulation and early organogenesis'. In: *Nature* 566.7745 (2019). DOI: 10.1038/s41586-019-0933-9 (cited on pages 117, 191, 192, 195).

[121] Grace X.Y. Zheng et al. 'Massively parallel digital transcriptional profiling of single cells'. In: *Nature Communications* 8 (Jan. 2017). DOI: 10.1038/ncomms14049 (cited on pages 119, 124, 126, 134, 136).

[122] Cathie Sudlow et al. 'UK Biobank: An Open Access Resource for Identifying the Causes of a Wide Range of Complex Diseases of Middle and Old Age'. In: *PLoS Medicine* 12.3 (Mar. 2015). DOI: 10.1371/journal.pmed.1001779 (cited on pages 119, 130, 135).

[123] Karl Pearson. 'LIII. On lines and planes of closest fit to systems of points in space'. In: *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 2.11 (1901), pp. 559–572 (cited on pages 121, 154).

[124] Michael A. A. Cox and Trevor F. Cox. 'Multidimensional Scaling'. In: *Handbook of Data Visualization* (2008), pp. 315–347. DOI: 10.1007/978-3-540-33037-0{_}14 (cited on page 121).

[125] J. B. Tenenbaum, V. De Silva, and J. C. Langford. 'A global geometric framework for nonlinear dimensionality reduction'. In: *Science* 290.5500 (Dec. 2000), pp. 2319–2323. DOI: 10.1126/SCIENCE.290.5500.2319/ASSET/3BDF9DAC-0D3D-4D72-B053-061BCF24AF03/ASSETS/GRAPHIC/SE5009080004.JPEG (cited on page 121).

[126] Theresa L Whiteside and Giorgio Parmiani. 'Tumor-infiltrating lymphocytes: their phenotype, functions and clinical use'. In: *Cancer Immunology, Immunotherapy* 39.1 (1994), pp. 15–21. DOI: 10.1007/BF01517175 (cited on page 121).

[127] Alexandre Bignon et al. 'DUSP4-mediated accelerated T-cell senescence in idiopathic CD4 lymphopenia'. In: *Blood, The Journal of the American Society of Hematology* 125.16 (2015), pp. 2507–2518 (cited on page 121).

[128] Fabien Agenes et al. 'Differential expression of regulator of G-protein signalling transcripts and in vivo migration of CD4+ naive and regulatory T cells'. In: *Immunology* 115.2 (2005), pp. 179–188 (cited on page 121).

[129] Xinyi Guo et al. 'Global characterization of T cells in non-small-cell lung cancer by single-cell sequencing'. In: *Nature medicine* 24.7 (2018), pp. 978–985 (cited on pages 121, 134, 136).

[130] Xiufang Xiong et al. 'Ribosomal protein S27-like and S27 interplay with p53-MDM2 axis as a target, a substrate and a regulator'. In: *Oncogene* 30.15 (2011), pp. 1798–1811 (cited on page 123).

[131] Alexandra-Chloé Villani et al. 'Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors'. In: *Science* 356.6335 (Apr. 2017), eaah4573. DOI: 10.1126/science.aah4573 (cited on pages 125, 126, 134, 136).

[132] Karolina A Palucka et al. 'Dendritic Cells as the Terminal Stage of Monocyte Differentiation'. In: *Journal of Immunology* 160 (1998), pp. 4587–4595 (cited on page 125).

[133] Brian K Stansfield and David A Ingram. 'Clinical significance of monocyte heterogeneity'. In: *Clinical and Translational Medicine* 4.1 (2015), p. 5. DOI: 10.1186/s40169-014-0040-3 (cited on page 125).

[134] Christine A Wells et al. 'Alternate transcription of the Toll-like receptor signaling cascade'. In: *Genome Biology* 7.2 (2006), R10. DOI: 10.1186/gb-2006-7-2-r10 (cited on page 125).

[135] Michal Slyper et al. *Study: ICA: Blood Mononuclear Cells (2 donors, 2 sites)*. URL: https://singlecell.broadinstitute.org/single_cell/study/SCP345/ica-blood-mononuclear-cells-2-donors-2-sites (cited on pages 125, 134, 136).

[136] Martin Guilliams et al. *Dendritic cells, monocytes and macrophages: A unified nomenclature based on ontogeny*. 2014. DOI: 10.1038/nri3712 (cited on page 125).

[137] Henry B Mann and Donald R Whitney. 'On a test of whether one of two random variables is stochastically larger than the other'. In: *The annals of mathematical statistics* (1947), pp. 50–60 (cited on page 127).

[138] Luke A D Hutchison, Bonnie Berger, and Isaac S Kohane. 'Meta-analysis of Caenorhabditis elegans single-cell developmental data reveals multi-frequency oscillation in gene activation'. In: *Bioinformatics* (Dec. 2019). DOI: 10.1093/bioinformatics/btz864 (cited on page 128).

[139] Virginie Freytag et al. 'Genome-wide temporal expression profiling in Caenorhabditis elegans identifies a core gene set related to long-term memory'. In: *Journal of Neuroscience* 37.28 (2017), pp. 6661–6672 (cited on page 128).

[140] Olga Minkina and Craig P Hunter. 'Intergenerational transmission of gene regulatory information in Caenorhabditis elegans'. In: *Trends in Genetics* 34.1 (2018), pp. 54–64 (cited on page 128).

[141] Martin C J Maiden. 'Multilocus Sequence Typing of Bacteria'. In: *Annual Review of Microbiology* 60.1 (Sept. 2006), pp. 561–588. DOI: 10.1146/annurev.micro.59.030804.121325 (cited on page 130).

[142] Turner Lee Nicol. 'Detecting racial bias in algorithms and machine learning'. In: *Journal of Information, Communication and Ethics in Society* 16.3 (Jan. 2018), pp. 252–260. DOI: 10.1108/JICES-06-2018-0056 (cited on page 130).

[143] Alex Diaz-Papkovich et al. 'UMAP reveals cryptic population structure and phenotype heterogeneity in large genomic cohorts'. In: *PLOS Genetics* 15.11 (2019), pp. 1–24. DOI: 10.1371/journal.pgen.1008432 (cited on page 130).

[144] Yasin Uzun et al. 'SINBAD: a flexible tool for single cell DNA methylation data'. In: *bioRxiv* (Oct. 2021), p. 2021.10.23.465577. DOI: 10.1101/2021.10.23.465577 (cited on page 131).

[145] Régis Ebeling et al. 'The effect of political polarization on social distance stances in the Brazilian COVID-19 scenario'. In: *Journal of Information and Data Management* 12.1 (2021). DOI: 10.5753/jidm.2021.1889 (cited on page 131).

[146] Seongyong Park et al. 'Machine Learning-Based Heavy Metal Ion Detection Using Surface-Enhanced Raman Spectroscopy'. In: *Sensors 2022, Vol. 22, Page 596* 22.2 (Jan. 2022), p. 596. DOI: 10.3390/S22020596 (cited on page 131).

[147] Hyunghoon Cho, Bonnie Berger, and Jian Peng. 'Generalizable and scalable visualization of single-cell data using neural networks'. In: *Cell systems* 7.2 (2018), pp. 185–191 (cited on page 134).

[148] Anthony R Cillo et al. 'Immune landscape of viral-and carcinogen-driven head and neck cancer'. In: *Immunity* 52.1 (2020), pp. 183–199 (cited on pages 138, 155).

[149] Xianwen Ren et al. 'Insights Gained from Single-Cell Analysis of Immune Cells in the Tumor Microenvironment'. In: *Annual Review of Immunology* 39 (2021), pp. 583–609 (cited on pages 138, 155).

[150] Zuzanna Lukasik, Dirk Elewaut, and Koen Venken. 'MAIT Cells Come to the Rescue in Cancer Immunotherapy?' In: *Cancers* 12.2 (2020), p. 413 (cited on page 138).

[151] Oliver Nussbaumer and Michael Koslowski. 'The emerging role of $\gamma\delta$ T cells in cancer immunotherapy'. en. In: *Immuno-Oncology Technology* 1 (July 2019), pp. 3–10. DOI: 10.1016/j.iotech.2019.06.002 (cited on page 138).

[152] Kevin R. Moon et al. 'Visualizing structure and transitions in high-dimensional biological data'. In: *Nature Biotechnology 2019 37:12* 37.12 (Dec. 2019), pp. 1482–1492. DOI: 10.1038/s41587-019-0336-3 (cited on page 139).

[153] Michael Hagemann-Jensen et al. 'Single-cell RNA counting at allele and isoform resolution using Smart-seq3'. In: *Nature Biotechnology* 38.6 (2020), pp. 708–714 (cited on pages 141, 155, 156).

[154] C.M. Bishop. 'Variational principal components'. In: *9th International Conference on Artificial Neural Networks: ICANN '99* 1999 (1999), pp. 509–514. DOI: 10.1049/CP:1999116010.1049/CP:19991160 (cited on page 149).

[155] Tabula Muris Consortium et al. 'Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris'. In: *Nature* 562.7727 (2018), pp. 367–372 (cited on page 152).

[156] Cristal Huysamen and Gordon D. Brown. 'The fungal pattern recognition receptor, Dectin-1, and the associated cluster of C-type lectin-like receptors'. In: *FEMS Microbiology Letters* 290.2 (Jan. 2009), pp. 121–128. DOI: 10.1111/J.1574-6968.2008.01418.X (cited on page 154).

[157] Patrick Wolter et al. 'GAS2L3, a target gene of the DREAM complex, is required for proper cytokinesis and genomic stability'. In: *Journal of Cell Science* 125.10 (May 2012), pp. 2393–2406. DOI: 10.1242/JCS.097253/258437/AM/GAS2L3-A-NOVEL-TARGET-GENE-OF-THE-DREAM-COMPLEX-IS (cited on page 154).

[158] Tatyana Veremeyko et al. 'The Role of Neuronal Factors in the Epigenetic Reprogramming of Microglia in the Normal and Diseased Central Nervous System'. In: *Frontiers in Cellular Neuroscience* 13 (Oct. 2019), p. 453. DOI: 10.3389/FNCEL.2019.00453/BIBTEX (cited on page 154).

[159] Jim McWhir et al. 'Mice with DNA repair gene (ERCC-1) deficiency have elevated levels of p53, liver nuclear abnormalities and die before weaning'. In: *Nature Genetics 1993 5:3* 5.3 (1993), pp. 217–224. DOI: 10.1038/ng1193-217 (cited on page 155).

[160] J-P Villeneuve and V Pichette. 'Cytochrome P450 and liver diseases'. In: *Current drug metabolism* 5.3 (2004), pp. 273–282 (cited on page 155).

[161] Elham Azizi et al. 'Single-Cell Map of Diverse Immune Phenotypes in the Breast Tumor Microenvironment'. In: *Cell* 174.5 (Aug. 2018), pp. 1293–1308. DOI: 10.1016/J.CELL.2018.05.060 (cited on page 155).

[162] Zhengtao Xiao, Ziwei Dai, and Jason W. Locasale. 'Metabolic landscape of the tumor microenvironment at single cell resolution'. In: *Nature Communications 2019 10:1* 10.1 (Aug. 2019), pp. 1–12. DOI: 10.1038/s41467-019-11738-0 (cited on page 155).

[163] S. Steven Potter. 'Single-cell RNA sequencing for the study of development, physiology and disease'. In: *Nature Reviews Nephrology 2018 14:8* 14.8 (May 2018), pp. 479–492. DOI: 10.1038/s41581-018-0021-7 (cited on page 155).

[164] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. 'SCANPY: large-scale single-cell gene expression data analysis'. In: *Genome Biology* 19.1 (2018), p. 15. DOI: 10.1186/s13059-017-1382-0 (cited on page 193).

[165] Volker Bergen et al. 'Generalizing RNA velocity to transient cell states through dynamical modeling'. In: *Nature Biotechnology* 38.12 (2020). DOI: 10.1038/s41587-020-0591-3 (cited on page 195).

[166] Eran Eden et al. 'GOrilla: A tool for discovery and visualization of enriched GO terms in ranked gene lists'. In: *BMC Bioinformatics* 10 (2009). DOI: 10.1186/1471-2105-10-48 (cited on page 197).

[167] Fran Supek et al. 'Revigo summarizes and visualizes long lists of gene ontology terms'. In: *PLoS ONE* 6.7 (2011). DOI: 10.1371/journal.pone.0021800 (cited on pages 197, 228).

[168] Yuzhi Chen, Rachael L. Neve, and Helena Liu. 'Neddylation dysfunction in Alzheimer's disease'. In: *Journal of Cellular and Molecular Medicine* 16.11 (2012). DOI: 10.1111/j.1582-4934.2012.01604.x (cited on page 197).

[169] Chantal M. Maghames et al. 'NEDDylation promotes nuclear protein aggregation and protects the Ubiquitin Proteasome System upon proteotoxic stress'. In: *Nature Communications* 9.1 (2018). DOI: 10.1038/s41467-018-06365-0 (cited on page 197).

A Concentration Bounds for Schema

Our approach is to show that, given a $\hat{\omega}$ that has been calculated based on a random sample, the correlation coefficient between *all* pairwise distances cannot be too different than the correlation coefficient computed on the sample. To do this, we use Chernoff bounds, which bound how far away a random variable can be from its expectation, on the covariance and variance terms of correlation coefficient given by (3.1). This gives us a bound on how far away the correlation coefficient on the whole population can be from the one calculated on the sample.

Let P be a random subset of all possible interactions. For now, we assume that interactions are chosen uniformly at random. Solving the optimization problem (3.2) with our sample P yields $\hat{\omega}$, an estimator for the true optimal transform ω. We show that $\hat{\omega}$ approximates ω well by showing that the pairwise distances of $\hat{\omega}(D)$ have high correlations with the secondary datasets as long as $\hat{\omega}$ has high correlations on the subsample.

Formally, we will guarantee, for any $\alpha, \delta > 0$ and sample size at least $|P| = O\left(\frac{\log(1/\alpha)}{\delta^2}\right)$,

$$\left|\text{Corr}(\hat{w}, \rho_j) - \widehat{\text{Corr}}(\hat{w}, \rho_j)\right| < \delta \text{ with probability at least } 1 - \alpha, \tag{A.1}$$

where $\widehat{\text{Corr}}(\,\cdot\,,\,\cdot\,)$ is the sample correlation coefficient.

This is a powerful result, made possible by our restriction to scaling transforms, which are easy to analyze. First of all, note that we only need a sample-size *logarithmic* in our desired confidence level in order to get strong concentration, allowing analysis of massive scRNA-seq datasets.

To begin our analysis, let $W \geq 0$ be a $k \times k$ psd matrix (in our specific case it will be diagonal, but this analysis will generalize to any psd matrix, which motivates the generalization to all psd matrices in Subsection 3.6.1). We also assume randomly draw pairwise distances δ uniformly from the set of pairs of points in our primary dataset. Here, we focus on the correlation between the transformed dataset and the primary dataset (i.e. the one that appears in the constraint in all of our examples). Analyses for correlations between the transformed data and the secondary datasets will be similar.

Consider the form of the (population) correlation:

$$\text{Corr}(W, \rho_1) = \frac{\overbrace{\mathbb{E}[\delta^T W \delta \delta^T \delta]}^{A} - \overbrace{\mathbb{E}[\delta^T W \delta]}^{B}\overbrace{\mathbb{E}[\delta^T \delta]}^{C}}{\underbrace{\text{Var}^{1/2}(W)}_{D}\underbrace{\text{Var}^{1/2}(\rho_1)}_{E}} \tag{A.2}$$

If, for our samples, we can determine confidence intervals of size 2ϵ for each of the terms A, B, C, D, E, then we can bound the distance away from the correlation on the *entire* set of pairwise distances. This distance is maximized when A is as small as possible, and B, C, D, and E is as large as possible. So:

$$\begin{aligned}
\widehat{\text{Corr}}(W, \rho_1) &\geq \frac{(A-\epsilon) - (B+\epsilon)(C+\epsilon)}{(D+\epsilon)(E+\epsilon)} \\
&\approx \frac{A - BC - (1+B+C)\epsilon}{DE(1+\epsilon/D)(1+\epsilon/E)} \\
&\approx \left(\frac{A-BC}{DE} - \frac{B+C+1}{DE}\epsilon\right)(1-\epsilon/D)(1-\epsilon/E) \\
&\approx \left(\frac{A-BC}{DE} - \frac{B+C+1}{DE}\epsilon\right)(1-\epsilon/D-\epsilon/E) \\
&\approx \left(\frac{A-BC}{DE}\right)\left(1 + \frac{D+E}{DE}\epsilon\right) - \frac{B+C+1}{DE}\epsilon \\
&= \text{Corr}(W, \rho_1)\left(1 - \frac{\text{Var}^{1/2}(W) + \text{Var}^{1/2}(\rho_1)}{\text{Var}^{1/2}(W)\text{Var}^{1/2}(\rho_1)}\epsilon\right) - \frac{1 + \mathbb{E}[\delta^T W \delta] + \mathbb{E}[\delta^T \delta]}{\text{Var}^{1/2}(W)\text{Var}^{1/2}(\rho_1)}\epsilon
\end{aligned}$$

Thus, for a desired overall confidence level η, the relationship between ϵ and η is given by:

$$\epsilon = \left(\frac{\text{Var}^{1/2}(W)\text{Var}^{1/2}(\rho_1)}{\max\left\{\text{Var}^{1/2}(W) + \text{Var}^{1/2}(\rho_1), 1 + \mathbb{E}[\delta^T W \delta] + \mathbb{E}[\delta^T \delta]\right\}}\right)\eta$$

To show that we can bound each of the terms A, B, C, D, E we use *Hoeffding's inequality* to limit how far away the terms can be from their expectations. Let $X_1, \ldots, X_n$ be i.i.d. random variables drawn from bounded range $[a, b]$, and set $s = b - a$, and let $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$. Then Hoeffding's inequality states:

$$\Pr\left[\bar{X} - \mathbb{E}X \geq t\right] \leq \exp\left(-\frac{nt^2}{s^2}\right)$$

This can be converted into giving a (one-sided) confidence interval of length t by substituting the probability on the left with a desired confidence level α, and solving for n, which gives a statement:

$$\mathbb{E}X \geq \bar{X} - t \text{ with confidence } 1-\alpha \text{ for } n \geq \frac{s^2 \log(1/\alpha)}{t^2} \tag{A.3}$$

We begin by applying the inequality on term $A = \mathbb{E}[\delta^T W \delta \delta^T \delta]$ by bounding $\delta^T W \delta \delta^T \delta$. It is clear that $|\delta^T W \delta \delta^T \delta| \leq |\delta^T W \delta||\delta^T \delta|$, so we can bound each individually. Note that we can assume without loss

of generality that W is diagonal here, because otherwise (since it is psd), we could write $W = UDU^T$, where D is diagonal and U is unitary; setting $y = U\delta$ yields $\left|\delta^T W \delta\right| = \left|y^T D y\right|$, and, by unitarity, $\|\delta\| = \|y\|$.

Then, by Cauchy-Schwarz:

$$\left|\delta^T W \delta\right| \leq \left|\sum \delta_i W_{ii} \delta_i\right| \leq \|W\|\|\delta\|^2 \tag{A.4}$$

where $\|W\|$ is the matrix-norm, i.e. $\|W\| = \sqrt{\mathrm{Tr}\left(W^T W\right)}$. So for a diagonal matrix, $\|W\|^2 = \sum W_{ii}^2$. We can bound $||\delta|| \leq \max_{x_i, x_j \in D}\{\left\|x_i - x_j\right\|\} \equiv \mathrm{diam}(D)$.

Thus, $\left|\delta^T W \delta \delta^T \delta\right| \leq \|W\|\mathrm{diam}^4(D)$.

To get a confidence interval of size ϵ, we plug into (A.3), so we require:

$$N \geq \frac{\|W\|\mathrm{diam}^8(D)\log(1/\alpha)}{\epsilon^2}$$

Note that the diameter is an *extremely* coarse bound for the above bound. Morally, one can replace "diameter" with "variance", and the user has control over $\|W\|$ by choice of hyperparameters. We also note that the sample complexity improves drastically if we focus only or *local* distances, as discussed in Subsection 3.6.1.

The same analysis can be used for terms B and C in (A.2), but the dependency on the diameter is not as bad for those terms, so term A is the worst case.

Now, we consider the variance terms D and E. For term E, note:

$$\mathrm{Var}(\delta^T \delta) = \mathbb{E}[(\delta^T \delta - \mathbb{E}[\delta^T \delta])^2]$$

Again, $\left|\delta^T \delta - \mathbb{E}[\delta^T \delta]\right|$ is bounded by the maximum squared distance in the dataset $\mathrm{diam}^2(D)$, so we can use the Hoeffding inequality from above in the same way.

And term D takes the same form as above, but with $\delta^T W \delta$ instead of $\delta^T \delta$. As noted in (A.4), this is a bounded random variable as well, but here with bound $\|W\|^2\mathrm{diam}^4(D)$.

Thus, in order to get a uniform confidence interval across all the terms, we require:

$$N \geq \frac{\|W\|^2\mathrm{diam}^8(D)\log(1/\alpha)}{\epsilon^2} \tag{A.5}$$

B Density-preservation on Traditional Metrics

To further assess the impact of our density-preservation objective on the properties of existing visualization tools, we evaluated our methods on three previously proposed metrics of visualization quality [52]: classification score (CS), mutual information score (MIS), and pairwise distance score (PDS) (Methods). Intuitively, CS and MIS measure the effectiveness of a visualization in conveying the clustering structure of the original dataset. To this end, CS evaluates the accuracy of classifiers that assign cells to known clusters based on the visualization coordinates, whereas MIS quantifies the agreement between clustering in the original space and in the visualization. Next, PDS measures the preservation of pairwise distances, considering long-range distances that our methods as well as UMAP and t-SNE do not aim to preserve, but might still convey useful information about the global organization of the dataset.

Across all five datasets we analyzed, den-SNE and densMAP consistently obtained CS and MIS close to those of t-SNE and UMAP, respectively, albeit with a slight reduction in performance; on average, CS was 2.3% lower for den-SNE and 1.8% lower for densMAP compared to their corresponding baselines, and MIS was 1.7% lower for den-SNE and 8.9% lower for densMAP (see Figures G.4 to G.8). These results are consistent with the observation that density-preserving visualizations, despite largely recapitulating the properties of the existing algorithms while additionally incorporating density information, occasionally show less clear cluster boundaries due to the sparsity of boundary regions. Overall, however, our algorithms still retain the substantial edge that nonlinear data visualization algorithms have in preserving clustering structure; e.g., a traditional approach to dimension reduction using PCA results in 35.7% (36.7%) lower performance on CS and 63.2% (64.1%) lower on MIS on average compared to t-SNE (UMAP). Moreover, the trade-off between preserving density and capturing the clustering structure can be modulated by the user by changing the weight of the density-preservation objective, and we confirmed that the den-SNE and densMAP scores converge to t-SNE and UMAP scores as that weight decreases (Figures G.4 to G.8).

With respect to PDS, den-SNE and densMAP generally outperformed t-SNE and UMAP, respectively, across a wide range of original distances. For instance, when computing the PDS over the shorter half of sampled pairwise distances, den-SNE obtained 19.9% higher PDS and densMAP obtained 47.2% higher PDS than their counterparts on average, agreeing with the intuition that preserving density is closely related to preserving the original distances between points. An exception is for the PDS on the full range of distances on the UKB and NSCLC datasets, where UMAP outperformed densMAP. We hypothesize that this behavior is due to outlier points from extremely sparse regions of the dataset,

which may distort the long-range distances in the visualization (e.g. between different clusters) to a greater extent than the existing tools. Note that the primary focus of nonlinear visualization tools like t-SNE and UMAP is to preserve the local structure of the dataset. Indeed, when the longest distance quantiles are added, a linear dimension reduction by PCA tends to obtain higher PDS than all of the nonlinear methods, despite the overall poor visual clarity of embeddings produced by PCA for complex transcriptomic landscapes (see Figure G.9). These results demonstrate that our methods achieve density preservation while maintaining competitive performance according to existing notions of visualization quality.

C Gradient Computations for Density-preserving Visualizations

The core of our density-preserving tools lies in the optimization of the Pearson correlation between the log local radius of points in the original dataset and in the embedding (see Methods). Here we compute the gradient of this correlation with respect to the embedding coordinates for optimization. Let $X = \{x_i\}_{i=1}^N$ be our original dataset and $y_i = s(x_i)$ be our embedding, where $s \in \{\text{den-SNE, densMAP}\}$ is our algorithm of choice.

Let $\{R_i^o\}_{i=1}^N$ and $\{R_i^e\}_{i=1}^N$ be measures of pointwise density in the original and embedded spaces respectively. We discuss specific density functions at the end, but allow full generality here. Let $r_i^e = \log R_i^e$. We center the original densities, so we let $r_i^o = \log R_i^o - N^{-1} \sum_{k=1}^N \log R_i^o$.

Since we want the densities in the embedded and original dataset to have a power-law relationship, (Subsection 4.3.5) we maximize the correlation between $r^o = \{r_i^o\}_{i=1}^N$ and $r^e = \{r_i^e\}_{i=1}^N$, denoted $\rho_{e,o}$. We write:

$$\rho_{e,o} = \frac{\text{Cov}(r^e, r^o)}{\sigma^o \sigma^e} = \frac{\sum_{k=1}^N (r_k^e - \mu^e) r_k^o}{s^o (N-1)^{\frac{1}{2}} \left[\sigma^2 + \sum_{k=1}^N (r_k - \mu^e)^2\right]^{\frac{1}{2}}} \tag{C.1}$$

where μ^e is the average of r^e; σ^o and σ^e are the sample standard deviations of r^o and r^e respectively, and σ^2 is a user-specified constant for regularization (this ensures that the standard deviation of the embedded local radii does not go to zero).

Now, we compute the gradient of the correlation with respect to pairwise squared distances of the embedded datapoints, $d_{ij}^2 = \left\|y_i - y_j\right\|^2$:

$$\frac{\partial \rho_{e,o}}{\partial d_{ij}^2} = (s^o)^{-1} (N-1)^{-\frac{1}{2}} \left[\widetilde{\text{Var}}(r^e)^{-\frac{1}{2}} \frac{\partial \widetilde{\text{Cov}}(r^e, r^o)}{\partial d_{ij}^2} - \frac{1}{2} \widetilde{\text{Cov}}(r^e, r^o) \widetilde{\text{Var}}(r^e)^{-\frac{3}{2}} \frac{\partial \widetilde{\text{Var}}(r^e)}{\partial d_{ij}^2} \right],$$

where $\widetilde{\text{Var}}(r^e) = (N-1)\left[\sigma^2/(N-1) + \text{Var}(r^e)\right]$ and $\widetilde{\text{Cov}}(r^e, r^o) = (N-1)\text{Cov}(r^e, r^o)$ (these are sample variances and covariances, hence the normalization by $N-1$ instead of N).

Now consider the component parts:

$$\begin{aligned}\frac{\partial \widetilde{\mathrm{Cov}}(r^e, r^o)}{\partial d_{ij}^2} &= \sum_{k=1}^{N} r_k^o \left(\frac{\partial r_k^e}{\partial d_{ij}^2} - \frac{\partial \mu^e}{\partial d_{ij}^2} \right) \\ &= \sum_{k=1}^{N} r_k^o \frac{\partial r_k^e}{\partial d_{ij}^2} - \frac{\partial \mu^e}{\partial d_{ij}^2} \sum_{k=1}^{N} r_k^o \\ &= \sum_{k=1}^{N} r_k^o \frac{\partial r_k^e}{\partial d_{ij}^2},\end{aligned} \tag{C.2}$$

where the second term in (C.2) is zero since r^o is centered. Similarly,

$$\begin{aligned}\frac{\partial \widetilde{\mathrm{Var}}(r^e)}{\partial d_{ij}^2} &= 2 \sum_{k=1}^{N} (r_k^e - \mu^e) \left(\frac{\partial r_k^e}{\partial d_{ij}^2} - \frac{\partial \mu^e}{\partial d_{ij}^2} \right) \\ &= 2 \sum_{k=1}^{N} (r_k^e - \mu^e) \frac{\partial r_k^e}{\partial d_{ij}^2} - 2 \frac{\partial \mu^e}{\partial d_{ij}^2} \sum_{k=1}^{N} (r_k^e - \mu^e) \\ &= 2 \sum_{k=1}^{N} (r_k^e - \mu^e) \frac{\partial r_k^e}{\partial d_{ij}^2},\end{aligned} \tag{C.3}$$

where, similar to before, the second sum in (C.3) is zero.

For many density functions (and certainly the one we use), r_i will only depend on $\{d_{ik}, d_{ki}\}_{k=1}^{N}$, so we can further simplify the above expressions to:

$$\begin{aligned}\frac{\partial \widetilde{\mathrm{Cov}}(r^e, r^o)}{\partial d_{ij}^2} &= r_i^o \frac{\partial r_i^e}{\partial d_{ij}^2} + r_j^o \frac{\partial r_j^e}{\partial d_{ij}^2} \\ \frac{\partial \widetilde{\mathrm{Var}}(r^e)}{\partial d_{ij}^2} &= 2 \left((r_i^e - \mu^e) \frac{\partial r_i^e}{\partial d_{ij}^2} + (r_j^e - \mu^e) \frac{\partial r_j^e}{\partial d_{ij}^2} \right).\end{aligned}$$

Under this scenario, putting this all together, we get:

$$\frac{\partial \rho_{e,o}}{\partial d_{ij}^2} = \frac{\widetilde{\mathrm{Var}}(r^e) \left(r_i^o \frac{\partial r_i^e}{\partial d_{ij}^2} + r_j^o \frac{\partial r_j^e}{\partial d_{ij}^2} \right) - \widetilde{\mathrm{Cov}}(r^e, r^o) \left((r_i^e - \mu^e) \frac{\partial r_i^e}{\partial d_{ij}^2} + (r_j^o - \mu^e) \frac{\partial r_j^e}{\partial d_{ij}^2} \right)}{s^o (N-1)^{\frac{1}{2}} \widetilde{\mathrm{Var}}(r^e)^{\frac{3}{2}}}. \tag{C.4}$$

The measure of local density for the embedded points we have used is the squared distance, weighted by the embedding distribution Q for either t-SNE or UMAP:

$$R_k^e = \left(\sum_{\ell=1}^{N} (1 + a d_{k\ell}^{2b})^{-1} \right)^{-1} \sum_{m=1}^{N} d_{km}^2 (1 + a d_{km}^{2b})^{-1} = \mathfrak{Z}_k^{-1} \sum_{m=1}^{N} d_{km}^2 (1 + a d_{km}^{2b})^{-1},$$

where $\mathfrak{Z}_k = \sum_{m=1}^{N} (1 + a d_{km}^{2b})^{-1}$. We also write $\widetilde{Q}_{k\ell} = \mathfrak{Z}_k^{-1} (1 + a d_{k\ell}^{2b})^{-1}$. Note that these are related to the

Q matrix and $\mathfrak{Z}$ partition functions of t-SNE and UMAP, equations (4.5) and (4.6) in Methods, by

$$\mathfrak{Z} = \sum \mathfrak{Z}_i$$
$$Q_{ij} = \widetilde{Q}_{ij} \frac{\mathfrak{Z}_i}{\mathfrak{Z}}.$$

To finish (C.4), we need to evaluate $\frac{\partial r_i^e}{\partial d_{ij}^2}$.

$$\begin{aligned}
\frac{\partial r_i^e}{\partial d_{ij}^2} &= \frac{\partial}{\partial d_{ij}^2} \log\left(\mathfrak{Z}_i R_i^e\right) - \frac{\partial}{\partial d_{ij}^2} \log \mathfrak{Z}_i \\
&= (R_i^e \mathfrak{Z}_i)^{-1} \frac{\partial (\mathfrak{Z}_i R_i^e)}{\partial d_{ij}^2} - \mathfrak{Z}_i^{-1} \frac{\partial \mathfrak{Z}_i}{\partial d_{ij}^2} \\
&= (R_i^e \mathfrak{Z}_i)^{-1} \left[(1 + a d_{ij}^{2b})^{-1} - abd_{ij}^{2b-2} d_{ij}^2 (1 + a d_{ij}^{2b})^{-2}) \right] + \mathfrak{Z}_i^{-1} abd_{ij}^{2b-2} (1 + a d_{ij}^{2b})^{-2} \\
&= \frac{\widetilde{Q}_{ij}}{R_i^e} (1 - abd_{ij}^{2b} (1 + a d_{ij}^{2b})^{-1}) + abd_{ij}^{2b-2} \widetilde{Q}_{ij}^2 \mathfrak{Z}_i \\
&= (1 + a d_{ij}^{2b})^{-1} \widetilde{Q}_{ij} \left[1 + a d_{ij}^{2b} - a d_{ij}^{2b} \right] + \widetilde{Q}_{ij}^2 \mathfrak{Z}_i \\
&= \widetilde{Q}_{ij}^2 \mathfrak{Z}_i \left[1 + \frac{1 + a d_{ij}^{2b} (1 - b)}{R_i^e} \right]
\end{aligned}$$

Note that when $a = b = 1$, as in t-SNE, this simplifies to:

$$\frac{\partial r_i^e}{\partial d_{ij}^2} = \widetilde{Q}_{ij}^2 \mathfrak{Z}_i \left[1 + \frac{1}{R_i^e} \right]. \tag{C.5}$$

As discussed in Methods, in both den-SNE and densMAP, for the sake of efficiency, we assume for the local radius computation that, for a point i with embedding y_i and original coordinates x_i, $d_{ij}^2 \neq 0$ only for j such that i and j are in the edge set E of the k-nearest neighbors graph produced by each algorithm. Since the objective functions of each algorithm prioritize preserving local structure, they should encourage i and j to be nearest neighbors in the embedding as well, and so we only need to consider density with respect to those points.

We can now write the full den-SNE gradient by combining the gradient of the correlation with the gradient of the original t-SNE objective function [38] (we discuss adaptations needed for the densMAP gradient below):

$$\nabla_{y_i} \mathcal{L}^{\text{den-SNE}} = \sum_{\{i,j\} \in E} P_{ij} Q_{ij} \mathfrak{Z} (y_i - y_j) - \sum_{j \neq i} Q_{ij}^2 \mathfrak{Z} (y_i - y_j) - \lambda \sum_{\{i,j\} \in E} \frac{\partial \rho_{e,o}}{\partial d_{ij}^2} (y_i - y_j),$$

where λ is a user-provided parameter that determines the weight of the density-preservation objective, and $\frac{\partial \rho_{e,o}}{\partial d_{ij}^2}$ is given in (C.4) with (C.5) plugged in.

C.1 Stochastic Gradient Descent for densMAP

Here we detail how densMAP adapts the stochastic gradient descent (SGD) formulation of UMAP. The cross-entropy loss function for UMAP and its gradient with respect to the squared distance d_{ij}^2 between points i, j in the embedding [39], given E, the set of edges in the nearest-neighbors graph, is:

$$\mathcal{L} = \sum_{\{i,j\}\in E} \underbrace{P_{ij}\log Q_{ij}}_{\text{attractive}} + \underbrace{(1-P_{ij})\log\big(1-Q_{ij}\big)}_{\text{repulsive}}$$

$$\frac{\partial \mathcal{L}}{\partial d_{ij}^2} = P_{ij}\frac{\partial}{\partial d_{ij}^2}[\log Q_{ij}] + (1-P_{ij})\frac{\partial}{\partial d_{ij}^2}[\log\big(1-Q_{ij}\big)],$$

where P and Q are the distributions on the original and embedded data respectively. Note that in UMAP, unlike t-SNE, the value Q_{ij} depends *only* on distance d_{ij}^2 and does not involve a normalization term over all edges.

To optimize the attractive term of the objective function, at each step, UMAP draws an edge $\{i, j\} \in E$ randomly according to the distribution P, and computes the gradient $\frac{\partial}{\partial d_{ij}^2}\left(\log Q_{ij}\right)(y_i - y_j)$. This means that, over the course of the optimization, edge $\{i, j\}$ will be chosen with proportion P_{ij}/Z, where $Z = \sum_{i\neq i} P_{ij}$. To estimate the repulsive term, a set of points $S = \{k_s\}_{s=1}^{n_s}$ is chosen *uniformly* at random and the algorithm computes the gradient $\frac{1}{|S|}\sum_{k\in S}\frac{\partial}{\partial d_{ik}^2}\left[\log(1-Q_{ik})\right](y_i - y_j)$. The size of S is a tunable parameter n_s.

Now, incorporating the density-preservation term into this objective function means taking the gradient of the correlation (C.1) and adding it to the UMAP gradient. The full gradient becomes:

$$\nabla_{y_i}\mathcal{L} = \sum_{\{i,j\}\in E}\left(P_{ij}\frac{\partial}{\partial d_{ij}^2}[\log Q_{ij}] + (1-P_{ij})\frac{\partial}{\partial d_{ij}^2}[\log\big(1-Q_{ij}\big)] + \lambda\frac{\partial \rho_{e,o}}{\partial d_{ij}^2}\right)(y_i - y_j).$$

Note that, for the correlation term of the optimization, each *edge* is given equal weight (i.e. the term is not weighted by P_{ij}). Since, in the stochastic descent algorithm of UMAP, an edge is actually chosen with proportion P_{ij}/Z, we *re-weight* the gradient estimate of the correlation term by multiplying by $Z/(NP_{ij})$ (we divide by the number of points N for numerical stability, since Z grows with N).

The densMAP gradient estimate for an edge $\{i, j\}$ at each iteration of the SGD is then:

$$\nabla_{y_i}\mathcal{L}|_{\{i,j\}} = \left(\frac{\partial}{\partial d_{ij}^2}[\log Q_{ij}] + \lambda\frac{Z}{NP_{ij}}\frac{\partial \rho_{e,o}}{\partial d_{ij}^2} - \frac{1}{|S|}\sum_{k\in S}\frac{\partial}{\partial d_{ik}^2}[\log(1-Q_{ik})]\right)(y_i - y_j),$$

where S is a set of edges adjacent to i chosen uniformly at random. This ensures that, over the course of the optimization, the edges are weighted equally when optimizing the correlation term.

Next, we consider the Q distribution. Since, unlike t-SNE, UMAP does *not* normalize Q_{ij} over all the edges, it treats the term as a Bernoulli random variable over each edge. For the calculation of the local radius, however, we need a probability distribution over the nearest neighbors of each point. In

other words, we need $Q_{ij}/\sum_{k:\{i,k\}\in E} Q_{ik} = Q_{ij}/\mathfrak{L}_i$. To achieve this, we compute $\mathfrak{L}_i$ at the start of each epoch and take it as fixed for all the edges that are updated in that epoch, which is akin to performing coordinate descent with the update for $\mathfrak{L}_i$ happening once per epoch. Similarly, we compute the local radius R_i^e, and global variance and covariance terms at the start of each epoch. These techniques allow us to use SGD to optimize densMAP in a similar manner as UMAP.

D Schema: Differential Expression and Batch Effects

Aside from cell type inference, another important single-cell analysis task that stands to benefit from multimodal synthesis is the identification of differentially expressed marker genes. To illustrate how, we explored a mouse gastrulation single-cell dataset [120], consisting of 16,152 epiblast cells split over three developmental timepoints ($E6.5$, $E7.0$, and $E7.25$) and with two replicates at each timepoint, resulting in six distinct batches (Figure G.21a). Applying Schema to this dataset, we sought to identify differentially expressed genes that are consistent with the developmental time course while being robust to batch effects between the replicate pairs. To perform differential expression analysis with Schema, RNA-seq data should be used as the primary modality, while the distance metrics of the secondary modalities specify how cells should be differentiated from each other. Here, we used batch and developmental-age information as secondary modalities, configuring Schema to maximize RNA-seq data's agreement with developmental age and minimize its agreement with batch information. We weighted these co-objectives equally; results were robust to ±25% variations in these weights (Figure G.28). We used RNA-seq data as the primary dataset, representing it by its top ten principal components. (Methods below).

We evaluated Schema alongside MOFA+, a recently introduced single-cell multimodal analysis technique [72, 90]. Schema and MOFA+ approach the data synthesis problem from complementary perspectives: while the emphasis in Schema is to identify important features of the primary dataset and its corresponding transformation that reflects a synthesis of the various modalities, MOFA+ focuses on *de novo* identification of features that explain the covariation across modalities. In MOFA+ analysis by Argelaguet et al. [72] of this dataset, the authors identified ten factors that capture similar information to the top principal components (Figure G.27). To identify differentially expressed genes with MOFA+, we selected the top genes from two factors (MOFA1 and MOFA4) reported by Argelaguet et al. [72] as capturing developmental variation.

In addition to accounting for batch effects, we could also configure Schema to reduce the weight of transient changes in expression, thus identifying genes with monotonically changing expression along the time course (Figure G.21b–d). To do so, we encoded developmental age as a distance metric by specifying zero distance between cells at the same timepoint, unit distance between directly adjacent timepoints, and an additive sum of the unit distances across more separated timepoints. As a control, we also tested a metric that did distinguish between the stages but did not increase in time, finding that the highest-weighted feature (PC5) in that case was indeed non-monotonic (Figure G.21b–c). To

encode batch effect as a distance metric, we specified zero distance between cells in the same replicate and unit distance otherwise.

We estimated the set of differentially expressed genes as the top-loading genes of the principal components up-weighted by Schema. Seeking to evaluate if the Schema or MOFA+ genes did show time-dependent monotonicity in expression, we linearly regressed each identified gene's normalized expression against an ordering of the three developmental stages (we expand on this below). We found that the Schema genes corresponded to regression coefficients significantly different from zero (Figure G.21d–e), consistent with time-dependent monotonicity (two-sided t-test, $p = 3.83 \times 10^{-6}$); this was not true of MOFA+ ($p = 0.77$).

Next, we evaluated the batch-effect robustness of Schema and MOFA+ gene sets. Our configuration of Schema balances batch-effect considerations against differential expression considerations. For instance, introducing the batch-effect objective in Schema reduces the weights associated with the first and second principal components (PC1 and PC2), which show substantial within-timepoint batch-effect variations without a compensating time-dependent monotonicity, by 11% and 17%, respectively. In comparison, explicitly up-weighting "good" variation or down-weighting "bad" variation is difficult when using MOFA+. To systematically evaluate the batch-effect robustness of Schema and MOFA+ gene sets, we constructed benchmark sets of differentially expressed genes by applying a standard statistical test, adjusting for batch effects by exploiting the combinatorial structure of this dataset. Specifically, we aggregated over computations that each considered only one replicate per timepoint (Methods below). We then measured the overlap of Schema and MOFA+ gene sets with these benchmarks (Figure G.21f) and found that, compared to MOFA+, the Schema gene set shows a markedly higher overlap with the benchmarks that is statistically significant (hypergeometric test with Bonferroni correction, $p = 5.9 \times 10^{-12}$ for the benchmark set of size 188). Schema allows us to express the intuition that variation attributable to batch effects should be ignored while variation attributable to developmental age should be highlighted.

D.1 Methods

This mouse gastrulation dataset was originally described by Pijuan-Sala et al. [120] and investigated by Argelaguet et al. [90] and [72] using the MOFA+ algorithm. We operated on the data as preprocessed and made available by them and, for the MOFA+ evaluation in this paper, also used their pretrained models.

We first reduced the RNA-seq data (primary modality) to its top 10 principal components (PCs), to be in line with the 10 MOFA+ factors from Argelaguet et al. [90]. The MOFA+ algorithm can be thought of as a generalization of PCA and we did indeed observe that the top PCs were very similar to the top MOFA+ components (Figure G.27), validating that Schema was able to access the same sources of variation as found by MOFA+ here.

We configured Schema to use batch information as a secondary modality with weight -1 and developmental age information as a secondary modality with weight $+1$; thus, correlation with the former was minimized and the latter was maximized. The minimum correlation threshold was set to 0.9; we found that the results were robust to variations in this setting (0.8 and 0.95).

Since Schema accepts arbitrary distance measures on secondary datasets, we could investigate the impact of treating developmental timepoints as categories rather than a time ordering. In the categorical distance metric, we defined two cells to be at distance 0 if they were at the same developmental timepoint and at distance 1 otherwise. In the time-ordering metric, we specified the first and third time-points to be at distance 2 apart and the middle time-point to be at distance 1 from either end. The two distance measures lead to different feature-selection results from Schema, reflecting the distinct underlying variations in expression profiles. For category-based distances, PC5 receives the highest weight while PCs 4 and 6 are given higher weights in the time-ordering case (Figure G.21b). This happens because the mean expression level of PC5 shows large variation across the three time-points but does not change monotonically along the time course; in contrast, the PCs 4 and 6 display expression profiles that change monotonically with developmental age (Figure G.21c). Schema's flexibility to incorporate a distance measure that highlights specific variability patterns can thus enable researchers to identify precisely targeted gene-sets.

To create a gene set from Schema's feature weighting, we selected the intersection of k top loadings (by absolute value) of PCs up-weighted by Schema (PCs 4, 6 and 9 for the time-ordering metric); we choose k so that the intersection contained 30 genes. For MOFA+, we chose genes that had the top loadings (by absolute value) in the factors MOFA1 or MOFA4. Here, we were following Argelaguet et al. [72] who, after an investigation of the various MOFA+ factors, had identified these two as the most relevant to understanding developmental age variability. Since the top loadings of the two MOFA+ factors do not overlap much, we chose the top 16 genes from each, with their union consisting of 31 genes (there was one overlap between the two subsets).

For each gene identified by Schema or MOFA+, we regressed its expression against developmental time, encoding stages $E6.25$, $E7.0$ and $E7.25$ as timepoints $1, 2$ and 3, respectively. The gene's expression profile (across all cells) was first normalized to zero mean and unit standard deviation.

We created batch-effect adjusted benchmark gene sets by using different combinations of replicates. One can create a subset of the original dataset by sampling cells from only one of the two replicates at each time-point. By iterating over all possible combinations of replicates, we created eight such subsets. These subsets differ in the batch information they contain but share the same developmental age information. Using the Wilcoxon rank sum test in `scanpy` [164], we identified genes differentially expressed between the first ($E6.5$) and last ($E7.25$) stage in each subset and defined the benchmark gene set to consist of genes that are differentially expressed across a majority of the subsets. The benchmark set is thus robust to batch effects, being comprised of genes whose differential expression stands out across different replicates (i.e., batches). By varying the thresholds of the test, we could create benchmark sets of varying sizes and measured the overlap of Schema and MOFA+ gene sets with these. The Schema gene set has a higher overlap and for benchmark sets of all sizes, its overlap with them was significant (hypergeometric test with Bonferroni correction, $p = 5.9 \times 10^{-12}$ for the benchmark set of size 188 and Bonferroni-corrected $p < 10^{-5}$ for benchmark sets of all sizes, Figure G.21e); this was not the case for MOFA+.

E Schema and RNA Velocity

We next leveraged the flexibility of Schema to study cell differentiation by synthesizing spliced and unspliced mRNA counts in a dataset of 2,930 mouse dentate gyrus cells [120]. Specifying spliced counts as the primary dataset and unspliced counts as the secondary dataset, we configured Schema to compute a transformation of the spliced data that maximizes the correlation of its Euclidean distances with those in the unspliced dataset while distorting the former only minimally (Figure G.22a-c).

Our intuition here is the same as that underlying RNA velocity techniques: correlating spliced and unspliced counts in a cell should pick up on the time derivative of a cell's expression state and thus illuminate the cell differentiation process. To validate this intuition, we computed a pseudotime measure from the difference between transformed and original RNA-seq data, finding it to be highly correlated with the latent-time estimate produced by Bergen et al. [165] and their RNA velocity tool scVelo (Spearman rank correlation 0.72, two-sided t-test $p < 10^{-128}$, Figure G.22d). Since Schema relies on the same underlying biological phenomena as specialized RNA velocity tools but analyzes the data differently, these results show the breadth of Schema's generality and may be used to help supplement and strengthen the findings from standard RNA velocity analyses.

Schema can complement methods like scVelo by facilitating additional analyses. As in our demonstrations of cell type inference and UMAP visualization, the transformed data produced here by Schema incorporates additional information (the time derivative of expression) but remains analyzable as an RNA-seq dataset. As an example, we visualized the transformed dataset with t-SNE, finding that the two-dimensional t-SNE plot of the Schema-transformed data places more closely together cell types at similar stages of differentiation (as quantified by scVelo latent-time, Figure G.22e–g). To confirm this visual observation, we computed the Spearman rank correlation of scVelo latent-time differences between pairs of cells and their corresponding Euclidean distances in the t-SNE embedding space, finding that it increases from 0.397 in the original dataset to 0.432 in the transformation corresponding to a minimum correlation constraint of 0.95 (Methods below). In contrast, an unconstrained synthesis using CCA produced a substantially lower correlation of 0.163; see Figure G.26 for the corresponding CCA-based t-SNE visualization. Schema can thus facilitate visualizations that reflect the deeper underlying differentiation processes.

E.1 Methods

We normalized the spliced and unspliced counts, log transformed them and reduced them to their top 100 principal components. These were specified to Schema as the primary and secondary modalities, respectively. To construct a pseudotime estimate from Schema's output, we first computed the mean per-cell difference between the transformed and original RNA-seq data. Interpreting this difference as the major axis of transcriptional change, we projected the original RNA-seq values on it. The magnitude of projection for each cell is a score that we interpreted as a pseudotime measure.

F Schema and Differential Expression in Granule Cells

The densely-packed granule cell genes identified by Schema are strongly enriched for signal transmission, potentially indicating greater neurotransmission activity within these cells. In particular, REACTOME [114] pathway enrichment analysis (top 1000 genes, mapped to human) include vesicle-mediated transport (FDR $q = 5.11 \times 10^{-4}$), ion-channel transport (FDR $q = 1.82 \times 10^{-3}$), and cellular responses to external stimuli (FDR $q = 6.44x10^{-15}$) (Table S2, Figure G.30). An enrichment analysis of this gene set against the Gene Ontology (GO) database, performed using the GOrilla web-tool [166] and visualized using REViGO [167] also identified terms consistent with such activity: ion transport (GO:0022853, FDR $q = 1.8 \times 10^{-17}$), electron transfer (GO:009055, FDR $q = 2.87 \times 10^{-11}$) and enzyme binding (GO:0019899, FDR $q = 2.72 \times 10^{-11}$). (Table S3, Figure G.29). Interestingly, we also observed enrichment in REACTOME pathways related to autophagy (FDR $q = 3.19 \times 10^{-4}$), ubiquitination (FDR $q = 1.94 \times 10^{-4}$) and protein metabolism (FDR $q = 3.3 \times 10^{-7}$). In particular, we observed enrichment for the process of Neddylation (FDR $q = 2.26 \times 10^{-3}$), shown to have a role in nuclear protein aggregation [168, 169].

G Supplementary Figures

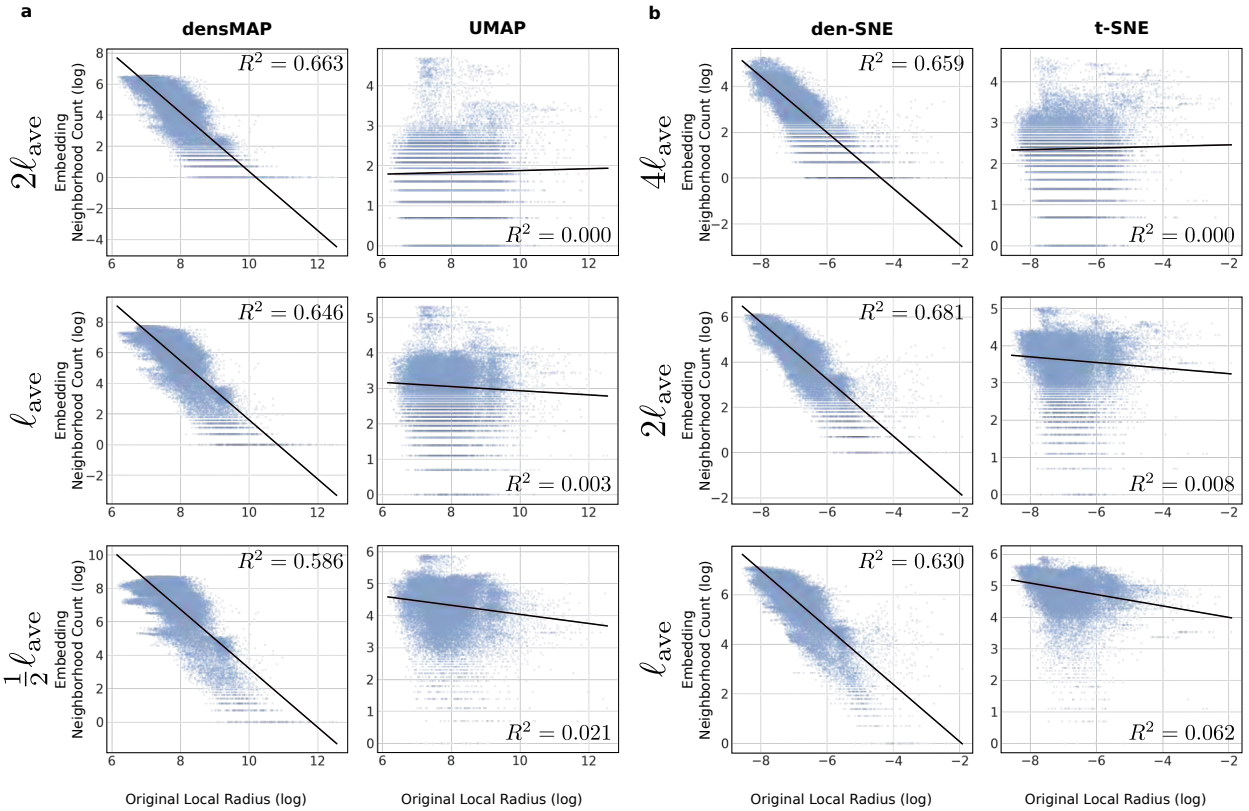

Figure G.1: Density-preserving methods preserve density robustly at different scales on lung cancer data based on neighborhood count. We compared the local radius of each point in the original lung cancer dataset to its neighborhood count in the visualizations (a measure of visual density; see Methods) for (**a**) densMAP and UMAP; and (**b**) den-SNE and t-SNE. We chose for each embedding, a length-scale ℓ_{ave} and multiples of that length-scale for which to compute the neighborhood counts (Methods): $\{\frac{1}{2}\ell_{\text{ave}}, \ell_{\text{ave}}, 2\ell_{\text{ave}}\}$ for densMAP and UMAP, and $\{\ell_{\text{ave}}, 2\ell_{\text{ave}}, 4\ell_{\text{ave}}\}$ for den-SNE and t-SNE. Since neighborhood count represents the density around a given point, a visualization that preserves density information will have higher neighborhood counts for points with smaller local radii in the original space. Note that this negative correlation is significantly stronger for our density-preserving tools than for t-SNE and UMAP, and this pattern holds across the different length-scales.

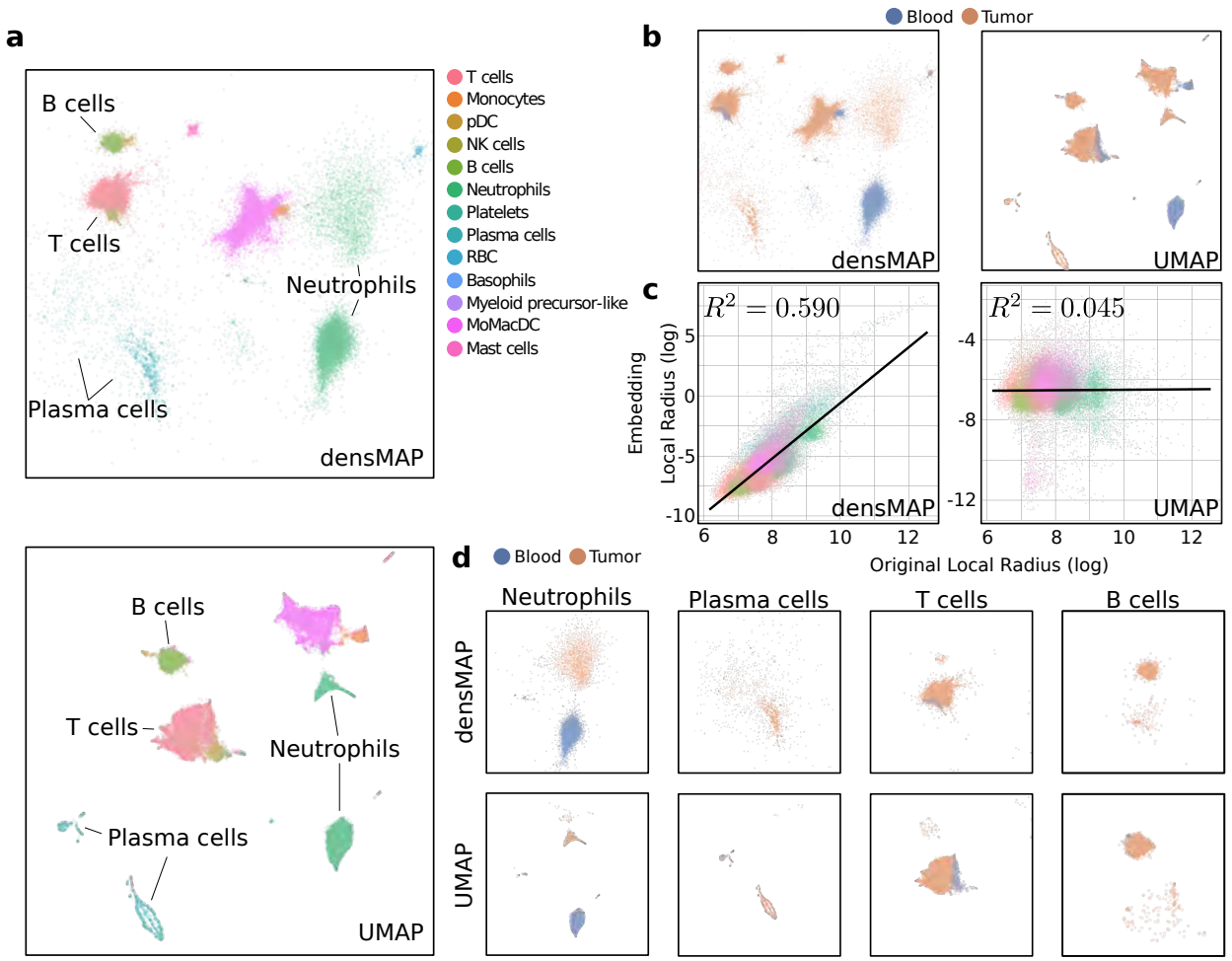

Figure G.2: Visualizing lung cancer using densMAP recapitulates den-SNE results. We repeat the analysis presented in Figure 6.1 using densMAP and UMAP. **a.** Top is a densMAP embedding and bottom is a UMAP embedding; points are colored by cell type. Note that the relative heterogeneity of neutrophils, plasma cells, and T cells are misleadingly portrayed in the UMAP visualization. **b.** densMAP (left) and UMAP (right) plots, now colored by tissue type (blood or tumor). **c.** Scatter plots comparing the local radii, our measure of local density (Methods), in the original space and in the visualization (embedding). Points are colored by cell type, and the R^2 value of the correlation is shown for each plot. Higher correlation of densMAP shows that densMAP more accurately conveys the density landscape of the original data than UMAP. Scatter plots based on neighborhood count (another measure of visual density; Methods) are included in Figure G.1. **d.** densMAP (top) and UMAP (bottom) plots restricted to each of four notable cell types (neutrophils, plasma cells, T cells, and B cells) and colored by tissue type (tumor or blood). Neutrophils and plasma cells in tumor considerably expand in size in densMAP, reflecting transcriptomic variability previously hidden in UMAP. T and B cells show a large increase in heterogeneity in tumor compared to blood in densMAP. Although UMAP shows a similar pattern, its lack of density-preservation property precludes reasoning about differences in heterogeneity.

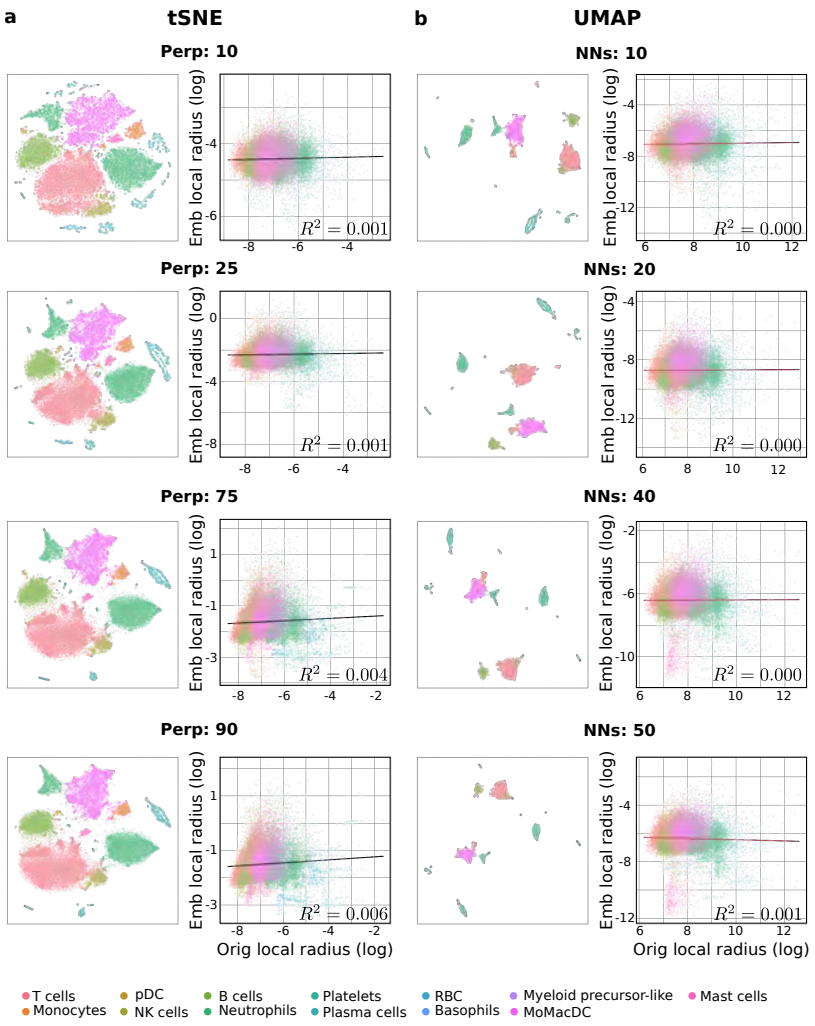

Figure G.3: Other choices of parameter do not yield density-preservation in tSNE and UMAP. Although changing the perplexity and n_neighbor parameters in t-SNE (**a**) and UMAP (**b**), respectively, can yield drastically different embeddings, this does not result in density preservation. We used the lung cancer dataset for this analysis. For t-SNE, we tried perplexity (Perp) values of 10, 25, 75, and 90, all of which resulted in near-zero correlation between the original and the embedded local radius. For UMAP, we chose n_neighbors (NNs) to be 10, 20, 40, and 50, and similarly, none of these choices led to density preservation. These results are consistent with our theoretical understanding of t-SNE and UMAP.

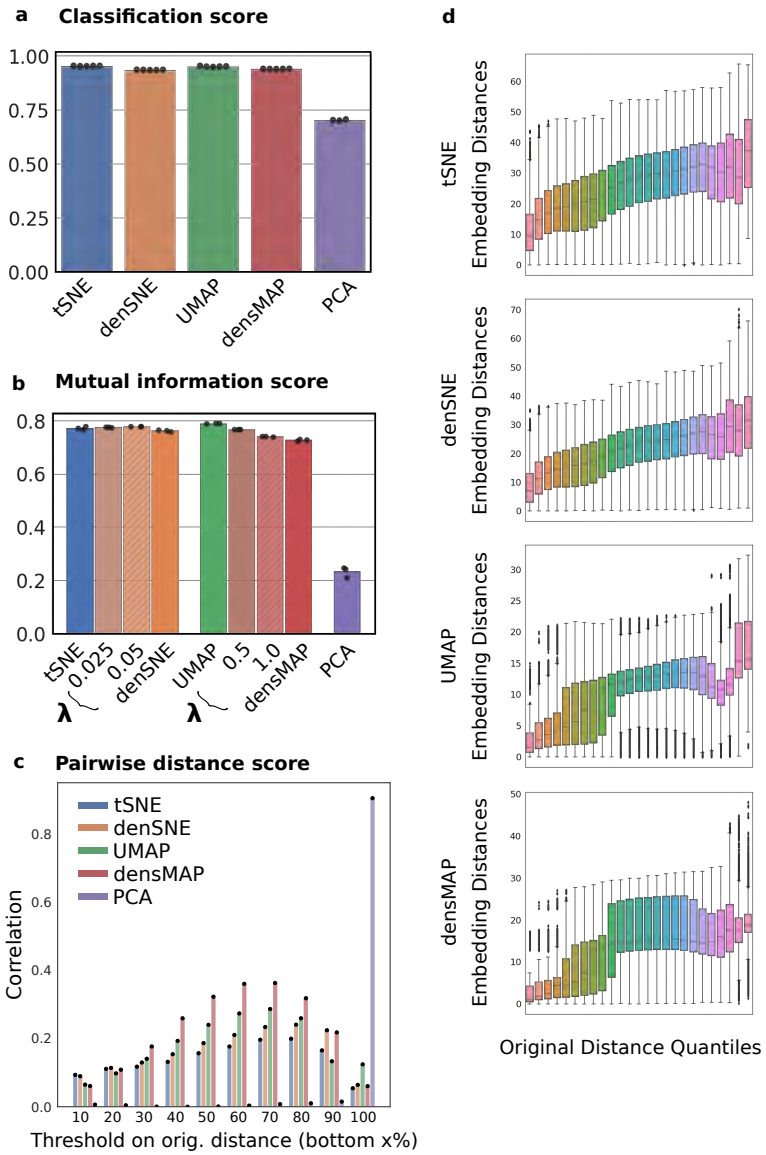

Figure G.4: Density-preserving methods achieve competitive performance on existing metrics for visualization quality on lung cancer data. We computed the classification score (**a**), the mutual information score (**b**), and the pairwise distance scores (**c** and **d**) as proposed in the literature (Methods) for t-SNE, den-SNE, UMAP, and densMAP, additionally including PCA as a baseline representing traditional dimensionality reduction approach. **a.** Both densMAP and den-SNE perform comparably to their counterpart on classifying cell types using embedding coordinates as features. Each bar is the mean of five subsample, with dots showing each individual measure. **b.** Both densMAP and den-SNE largely retain the superior performance of nonlinear data visualization methods on MIS compared to PCA, albeit with a slight reduction in performance of density-preserving methods. We believe this is due to the less clear cluster boundaries in our visualizations owing to their sparse nature. We also varied the weight λ of the density preserving methods from its default value to zero. The MIS increases as λ decreases, indicating a tunable tradeoff between clustering performance and density preservation (see Figure G.20). Each bar is the mean of three subsample, with dots showing each individual measure. **c.** We plot the pairwise distance score (Pearson correlation coefficient between the pairwise distances in the original and the embedding space) for the bottom x% of pairwise distances in the original space (indicated on the x-axis). Note that the density-preserving algorithms outperform their counterpart on all except the last decile. Each measure is performed on one subsample. **d.** We assign pairwise distances in the original space to 25 quantile bins and plot the corresponding distribution of distances in the embedding space. The boxes span the 25th to the 75th percentiles of the distribution, with the median marked. The whiskers extend to extrema (except outliers, individually marked outside of whiskers, which are defined as those points more than 1.5x the interquartile range (IQR) away from the box boundaries).

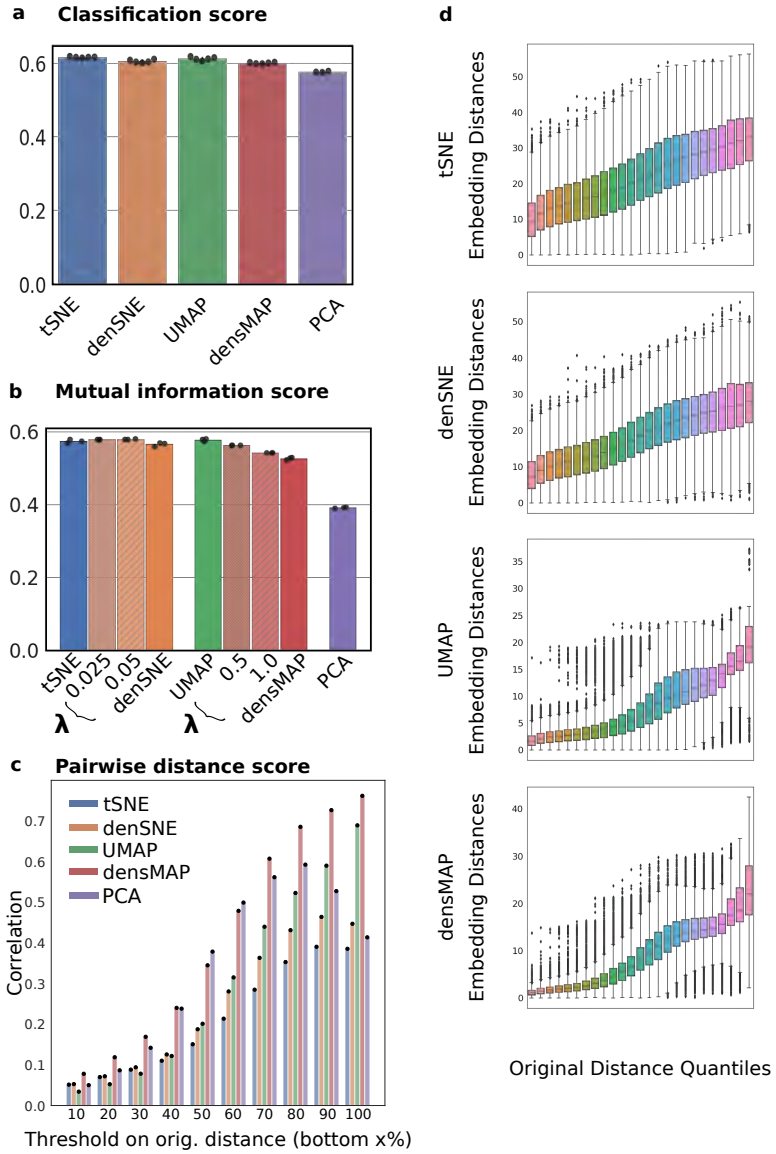

Figure G.5: Density-preserving methods achieve competitive performance on existing metrics for visualization quality on PBMC data. We computed the classification score (**a**), the mutual information score (**b**), and the pairwise distance scores (**c** and **d**) as proposed in the literature (Methods) for t-SNE, den-SNE, UMAP, and densMAP, additionally including PCA as a baseline representing traditional dimensionality reduction approach. **a.** Both densMAP and den-SNE perform comparably to their counterpart on classifying cell types using embedding coordinates as features. Each bar is the mean of five subsample, with dots showing each individual measure. **b.** Both densMAP and den-SNE largely retain the superior performance of nonlinear data visualization methods on MIS compared to PCA, albeit with a slight reduction in performance of density-preserving methods. We believe this is due to the less clear cluster boundaries in our visualizations owing to their sparse nature. We also varied the weight λ of the density preserving methods from its default value to zero. The MIS increases as λ decreases, indicating a tunable tradeoff between clustering performance and density preservation (see Figure G.20). Each bar is the mean of three subsample, with dots showing each individual measure. **c.** We plot the pairwise distance score (Pearson correlation coefficient between the pairwise distances in the original and the embedding space) for the bottom x% of pairwise distances in the original space (indicated on the x-axis). Note that the density-preserving algorithms outperform their counterpart on all deciles. Each measure is performed on one subsample. **d.** We assign pairwise distances in the original space to 25 quantile bins and plot the corresponding distribution of distances in the embedding space. The boxes span the 25th to the 75th percentiles of the distribution, with the median marked. The whiskers extend to extrema (except outliers, individually marked outside of whiskers, which are defined as those points more than 1.5x the interquartile range (IQR) away from the box boundaries).

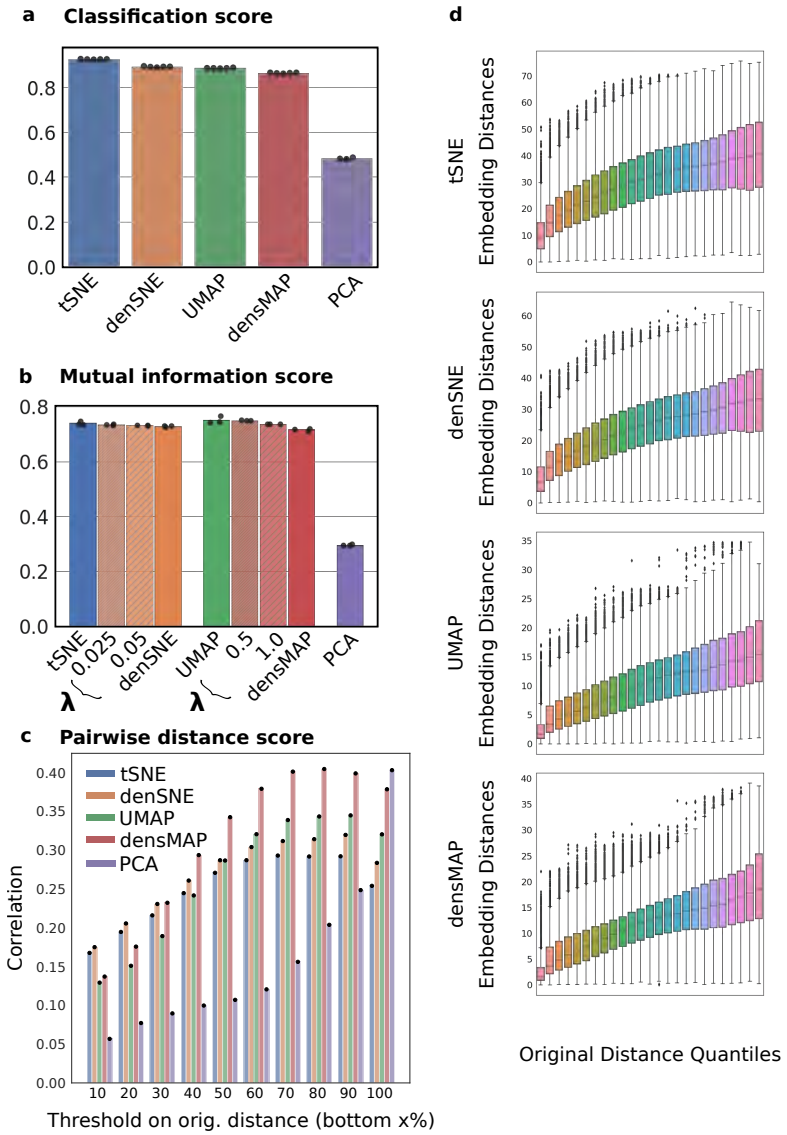

Figure G.6: Density-preserving methods achieve competitive performance on existing metrics for visualization quality on *C. elegans* data. We computed the classification score (**a**), the mutual information score (**b**), and the pairwise distance scores (**c** and **d**) as proposed in the literature (Methods) for t-SNE, den-SNE, UMAP, and densMAP, additionally including PCA as a baseline representing traditional dimensionality reduction approach. **a.** Both densMAP and den-SNE perform comparably to their counterpart on classifying cell types using embedding coordinates as features. Each bar is the mean of five subsample, with dots showing each individual measure. **b.** Both densMAP and den-SNE largely retain the superior performance of nonlinear data visualization methods on MIS compared to PCA, albeit with a slight reduction in performance of density-preserving methods. We believe this is due to the less clear cluster boundaries in our visualizations owing to their sparse nature. We also varied the weight λ of the density preserving methods from its default value to zero. The MIS increases as λ decreases, indicating a tunable tradeoff between clustering performance and density preservation (see Figure G.20). Each bar is the mean of three subsample, with dots showing each individual measure. **c.** We plot the pairwise distance score (Pearson correlation coefficient between the pairwise distances in the original and the embedding space) for the bottom x% of pairwise distances in the original space (indicated on the x-axis). Note that the density-preserving algorithms outperform their counterpart on all deciles. Each measure is performed on one subsample. **d.** We assign pairwise distances in the original space to 25 quantile bins and plot the corresponding distribution of distances in the embedding space. The boxes span the 25th to the 75th percentiles of the distribution, with the median marked. The whiskers extend to extrema (except outliers, individually marked outside of whiskers, which are defined as those points more than 1.5x the interquartile range (IQR) away from the box boundaries).

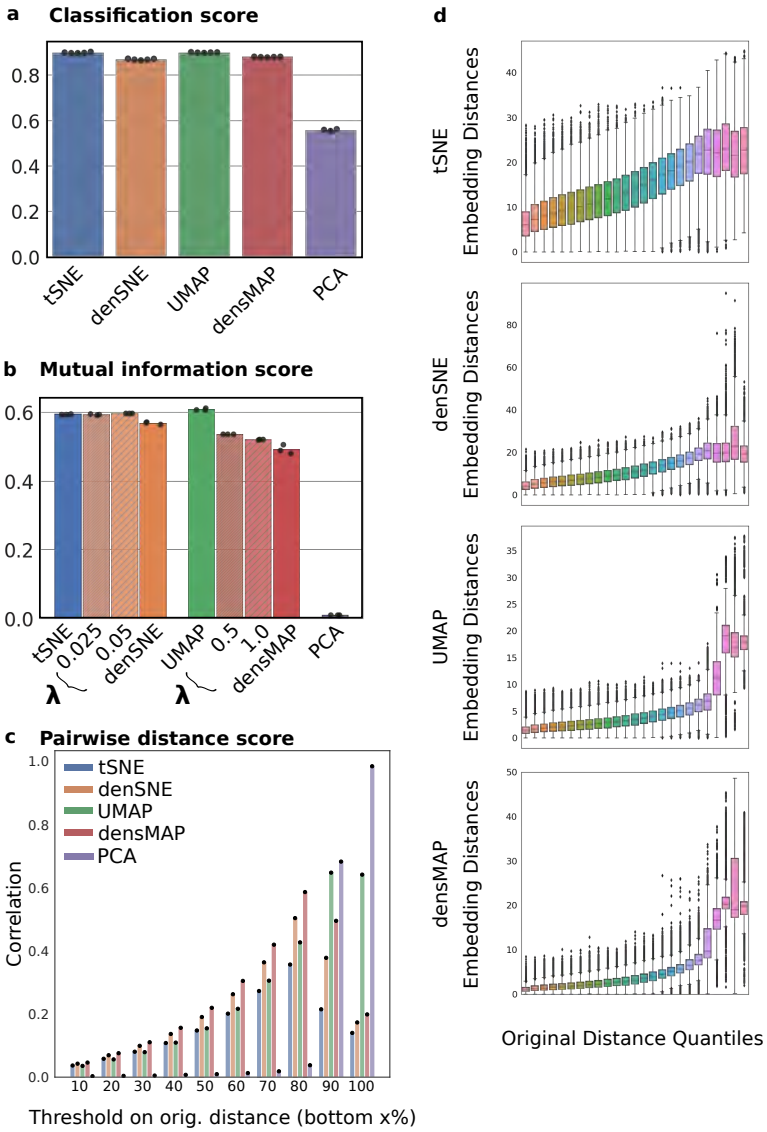

Figure G.7: Density-preserving methods achieve competitive performance on existing metrics for visualization quality on UK Biobank data. We computed the classification score (**a**), the mutual information score (**b**), and the pairwise distance scores (**c** and **d**) as proposed in the literature (Methods) for t-SNE, den-SNE, UMAP, and densMAP, additionally including PCA as a baseline representing traditional dimensionality reduction approach. **a.** Both densMAP and den-SNE perform comparably to their counterpart on classifying cell types using embedding coordinates as features. Each bar is the mean of five subsample, with dots showing each individual measure. **b.** Both densMAP and den-SNE largely retain the superior performance of nonlinear data visualization methods on MIS compared to PCA, albeit with a slight reduction in performance of density-preserving methods. We believe this is due to the less clear cluster boundaries in our visualizations owing to their sparse nature. We also varied the weight λ of the density preserving methods from its default value to zero. The MIS increases as λ decreases, indicating a tunable tradeoff between clustering performance and density preservation (see Figure G.20). Each bar is the mean of three subsample, with dots showing each individual measure. **c.** We plot the pairwise distance score (Pearson correlation coefficient between the pairwise distances in the original and the embedding space) for the bottom x% of pairwise distances in the original space (indicated on the x-axis). Note that the density-preserving algorithms outperform their counterpart on all except the last two deciles. Each measure is performed on one subsample. **d.** We assign pairwise distances in the original space to 25 quantile bins and plot the corresponding distribution of distances in the embedding space. The boxes span the 25th to the 75th percentiles of the distribution, with the median marked. The whiskers extend to extrema (except outliers, individually marked outside of whiskers, which are defined as those points more than 1.5x the interquartile range (IQR) away from the box boundaries).

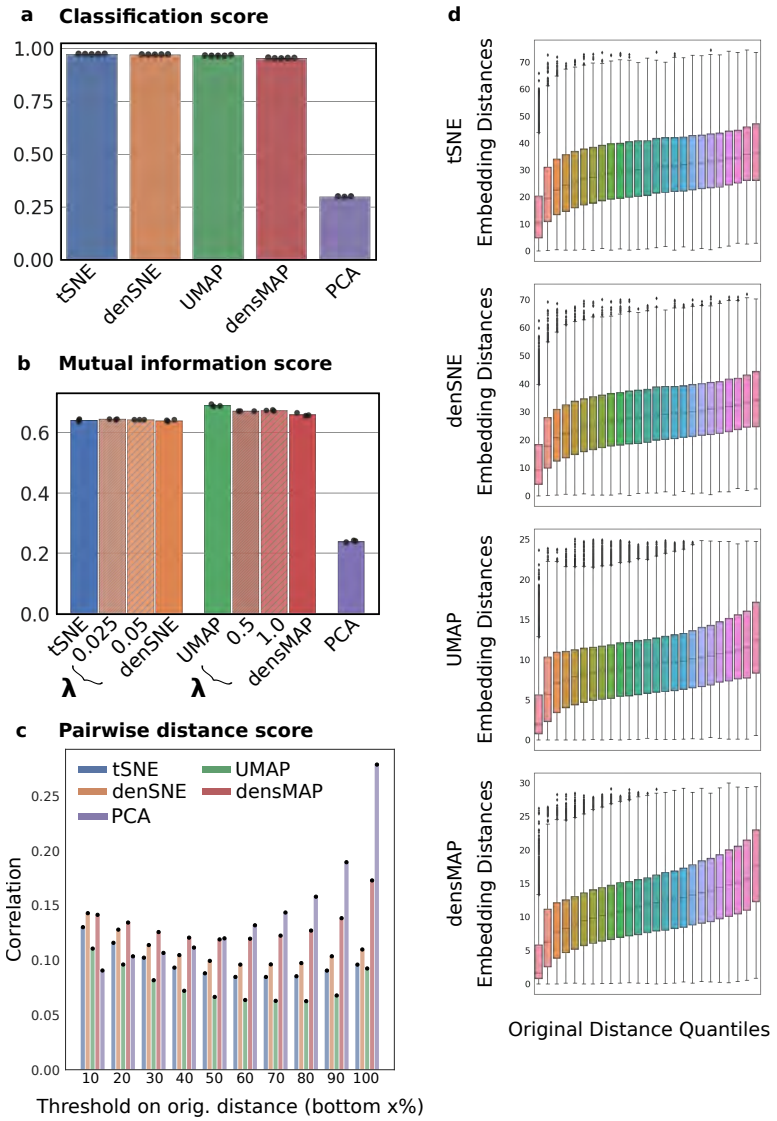

Figure G.8: Density-preserving methods achieve competitive performance on existing metrics for visualization quality on MNIST data. We computed the classification score (**a**), the mutual information score (**b**), and the pairwise distance scores (**c** and **d**) as proposed in the literature (Methods) for t-SNE, den-SNE, UMAP, and densMAP, additionally including PCA as a baseline representing traditional dimensionality reduction approach. **a.** Both densMAP and den-SNE perform comparably to their counterpart on classifying cell types using embedding coordinates as features. Each bar is the mean of five subsample, with dots showing each individual measure. **b.** Both densMAP and den-SNE largely retain the superior performance of nonlinear data visualization methods on MIS compared to PCA, albeit with a slight reduction in performance of density-preserving methods. We believe this is due to the less clear cluster boundaries in our visualizations owing to their sparse nature. We also varied the weight λ of the density preserving methods from its default value to zero. The MIS increases as λ decreases, indicating a tunable tradeoff between clustering performance and density preservation (see Figure G.20). Each bar is the mean of three subsample, with dots showing each individual measure. **c.** We plot the pairwise distance score (Pearson correlation coefficient between the pairwise distances in the original and the embedding space) for the bottom x% of pairwise distances in the original space (indicated on the x-axis). Note that the density-preserving algorithms outperform their counterpart on all deciles. Each measure is performed on one subsample. **d.** We assign pairwise distances in the original space to 25 quantile bins and plot the corresponding distribution of distances in the embedding space. The boxes span the 25th to the 75th percentiles of the distribution, with the median marked. The whiskers extend to extrema (except outliers, individually marked outside of whiskers, which are defined as those points more than 1.5x the interquartile range (IQR) away from the box boundaries).

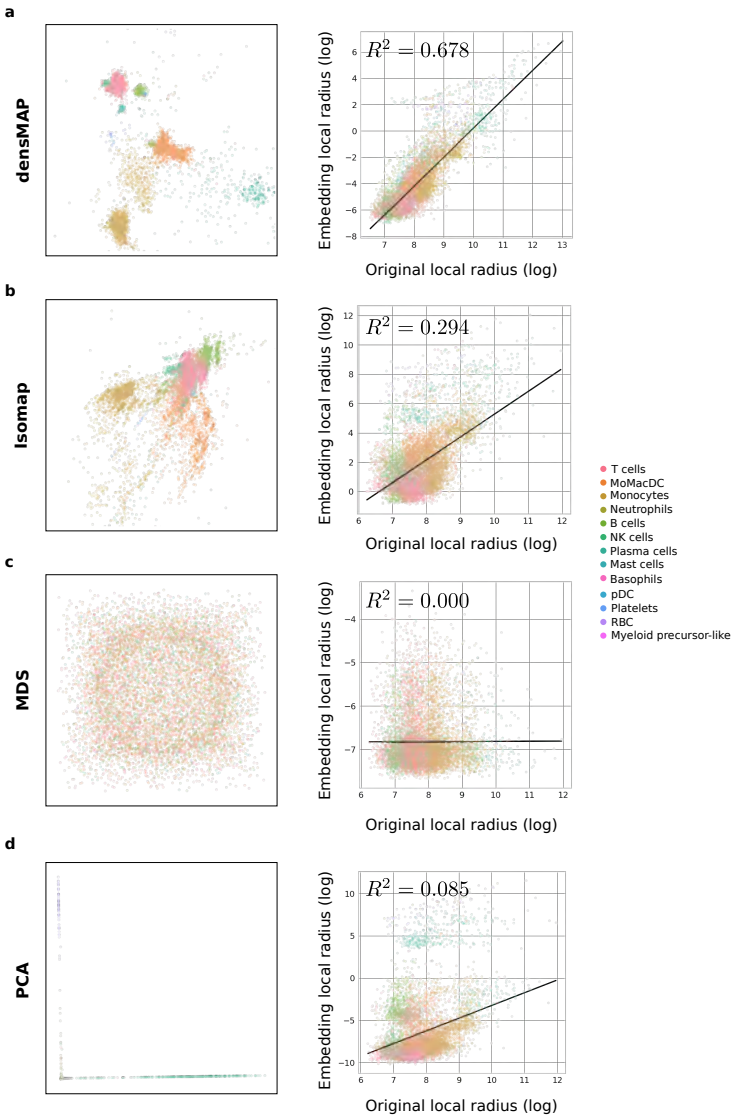

Figure G.9: Traditional dimensionality reduction algorithms struggle to produce informative visualizations of scRNA-seq data. Traditional dimensionality reduction algorithms, such as principal component analysis (PCA), Isomap, and multidimensional scaling (MDS), do not use an adaptive length-scale to model the data manifold, thus having the potential to preserve density better than t-SNE and UMAP. We tested these methods on the lung cancer dataset, subsampled to 10,000 cells (since Isomap and MDS do not scale to larger datasets). The resulting visualization and a scatter plot comparing local radius in the original space and in the visualization are shown for each method: densMAP (**a**), Isomap (**b**), MDS (**c**), and PCA (**d**). Isomap struggles to separate the clusters as well as densMAP; while its performance on the local radius correlation is better than that of UMAP (Figure G.2), it is substantially worse on this metric compared to densMAP. MDS attempts to preserve *all* pairwise distances and therefore struggles with high dimensional data; no clusters are visible in the embedding and density is not preserved. Similarly, PCA fails to clearly visualize the clustering structure of the dataset. Although PCA performance on local radius correlation is marginally better than UMAP (Figure G.2), it is significantly worse than densMAP.

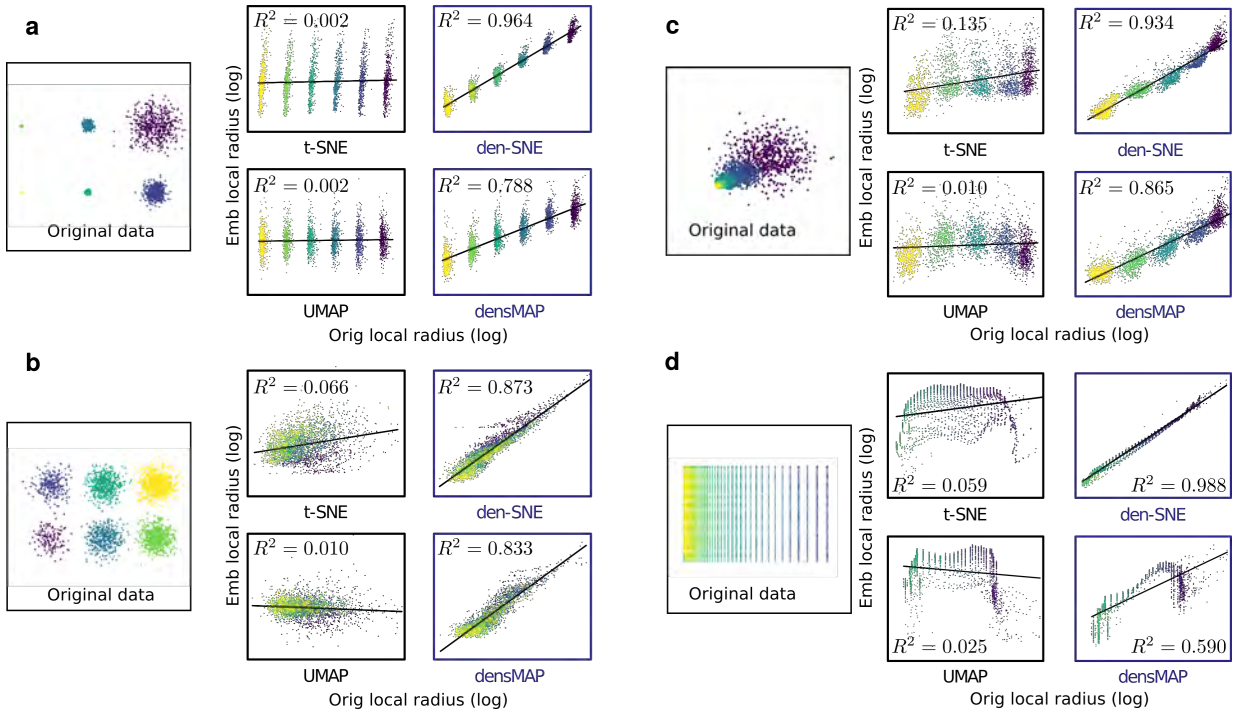

Figure G.10: Quantitative evaluation of density preservation on simulated datasets. We computed the local radius for each algorithm on the simulated datasets shown in Figure 4.2 to show the improvement that den-SNE and densMAP yield upon t-SNE and UMAP, respectively, with respect to density preservation. For each dataset, we include a scatter plot comparing the log local radius in the original space and in the embedding and the R^2 value of correlation between the two. Each dataset illustrates a different pattern of heterogeneity in local density: **a.** Gaussian point clouds with increasing variance and the same number of points; **b.** Gaussian point clouds with the same variance but increasing number of points; **c.** overlapping Gaussian point clouds with increasing variance; **d.** A grid of points, where density grows linearly in one direction. Consistent with the visual observation that our visualizations more accurate portray the original density landscape (Figure 4.2), the correlation in local radius is significantly higher for our methods compared to t-SNE and UMAP for all datasets.

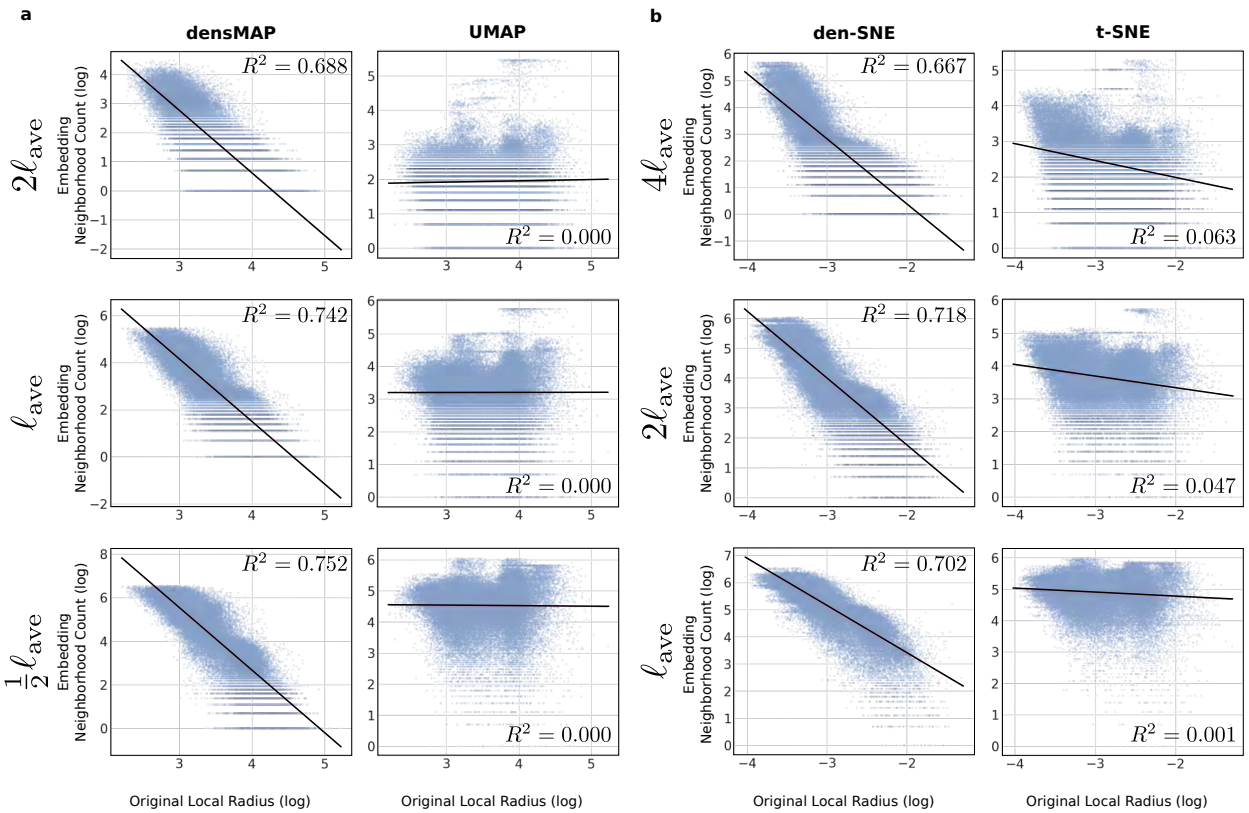

Figure G.11: Density-preserving methods preserve density robustly at different scales on PBMC data based on neighborhood count. We compared the local radius of each point in the original PBMC dataset to its neighborhood count in the visualizations (a measure of visual density; see Methods) for (**a**) densMAP and UMAP; and (**b**) den-SNE and t-SNE. We chose for each embedding, a length-scale ℓ_{ave} and multiples of that length-scale for which to compute the neighborhood counts (Methods): $\{\frac{1}{2}\ell_{\text{ave}}, \ell_{\text{ave}}, 2\ell_{\text{ave}}\}$ for densMAP and UMAP, and $\{\ell_{\text{ave}}, 2\ell_{\text{ave}}, 4\ell_{\text{ave}}\}$ for den-SNE and t-SNE. Since neighborhood count represents the density around a given point, a visualization that preserves density information will have higher neighborhood counts for points with smaller local radii in the original space. Note that this negative correlation is significantly stronger for our density-preserving tools than for t-SNE and UMAP, and this pattern holds across the different length-scales.

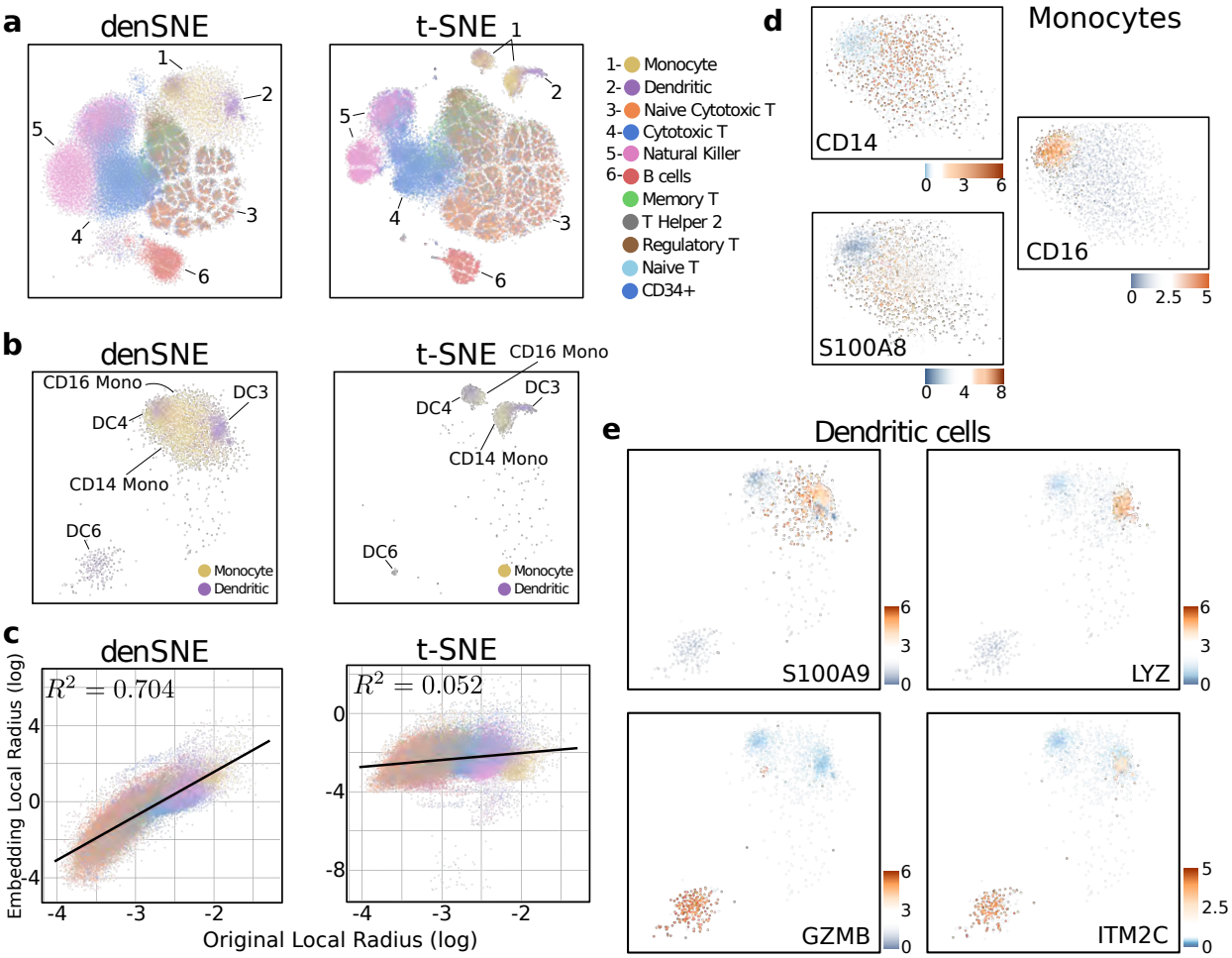

Figure G.12: Visualizing PBMC data using den-SNE recapitulates densMAP results. We repeat the analyses presented in Figure 6.2 using den-SNE and t-SNE. **a.** den-SNE (left) and t-SNE (right) visualizations of the data, colored by cell-type. The group of clusters corresponding to monocytes (cluster 1) and dendritic cells (DCs; cluster 2) showed the most pronounced difference between the two visualizations. **b.** For a detailed comparison, we plotted the same visualizations restricted to the monocyte-DC subset, which revealed distinct subtypes of monocytes (CD16 Mono and CD14 Mono) and DCs (DC3, DC4, and DC6) with clear density differences in den-SNE. Each subtype is annotated using the classification from the PBMC2 study (Villani et al., 2017; Methods) based on marker gene expression. Although the same subtypes are visible in t-SNE, their relative density differences are lost. **c.** Scatter plots comparing the local radii, our measure of local density (Methods), in the original space and in the visualization (embedding). Points are colored by cell type, and the R^2 value of the correlation is shown for each plot. Higher correlation of den-SNE shows that den-SNE more accurately conveys the density landscape of the original data than t-SNE. Scatter plots based on neighborhood count (another measure of visual density; Methods) are included in Figure G.11. **d.** Gene expression heatmaps of monocyte marker genes CD14, S100A8, and CD16 in the den-SNE visualization restricted to monocytes. The patterns of expression support our classification of the dense cluster as CD16 Mono and the sparse cluster as CD14 Mono. **e.** Gene expression heatmaps of DC marker genes from the PBMC2 study (Villani et al., 2017; Methods) for DC3 (top) and DC6 (bottom) in the densMAP visualization restricted to DCs. These support our assignment of DC clusters to DC3 and DC6.

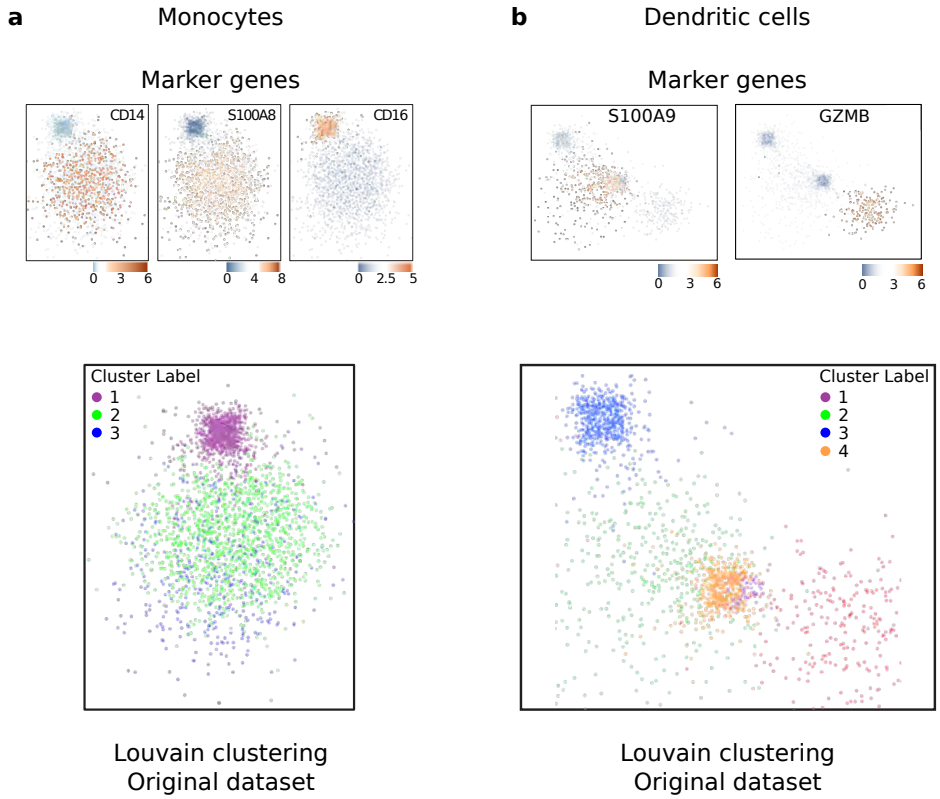

Figure G.13: Monocyte and dendritic cell subtypes with density differences correspond to distinct clusters in the original dataset. To test whether the monocyte (**a**) and dendritic cell (**b**) subtypes with density differences highlighted by our visualizations reflect distinct subpopulations of cells, we performed Louvain clustering of each cell type in the PBMC dataset based on their high-dimensional gene expression profiles. The top plots show the heatmaps of marker gene expression as shown in Figure 6.1 for reference, and the bottom plot shows the densMAP visualization colored by labels from the high-dimensional clustering. In both cell types, the identified clusters indeed correspond to subtypes with clear density differences.

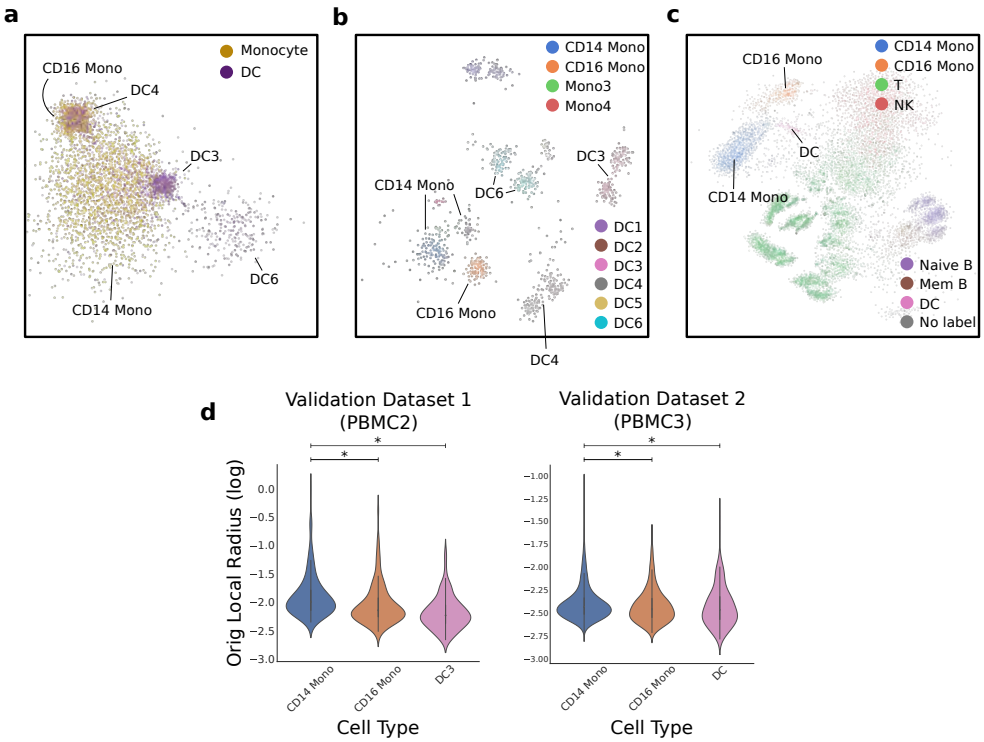

Figure G.14: Density differences among monocytes and dendritic cell subtypes are validated on additional datasets. Density-preserving visualizations of the (**a**) PBMC data, zoomed in on the monocytes and DCs (densMAP); (**b**) PBMC2 data (den-SNE); and (**c**) PBMC3 data (den-SNE) (Methods). In (**a**), the labels assigned are hypothesized based on subtypes determined in the PBMC2 dataset (see Figure G.15). The visualizations of PBMC2 and PBMC3 data recapitulate the density differences between CD14 Mono (which are sparse) and CD16 Mono/DC3 subsets (which are dense) observed in PBMC data. **d**. We further validated these density difference on the PBMC2 and PBMC3 data based on the original local radii computed for each of these datasets. For both PBMC2 (left) and PBMC3 (right), the local radius in the original data is significantly larger in CD14 monocytes than in CD16 monocytes. Similarly, DC3 has significantly smaller local radii than CD14 monocytes in the PBMC2 data; in the PBMC3 dataset, DC subtype labels were not available but the local radii of CD14 monocytes are still larger than that of the DCs. In PBMC2, there are 163 CD14+ monocytes, 122 CD16+ monocytes, and 107 DC3 cells; in PBMC3, there are 1,264 CD14+ monocytes, 398 CD16+ monocytes, and 142 DCs. The $*$ indicates a p-value less than 5×10^{-4} for a one-sided Mann-Whitney U test statistic (see Methods). NK: Natural killer cells; Mem B: Memory B cells

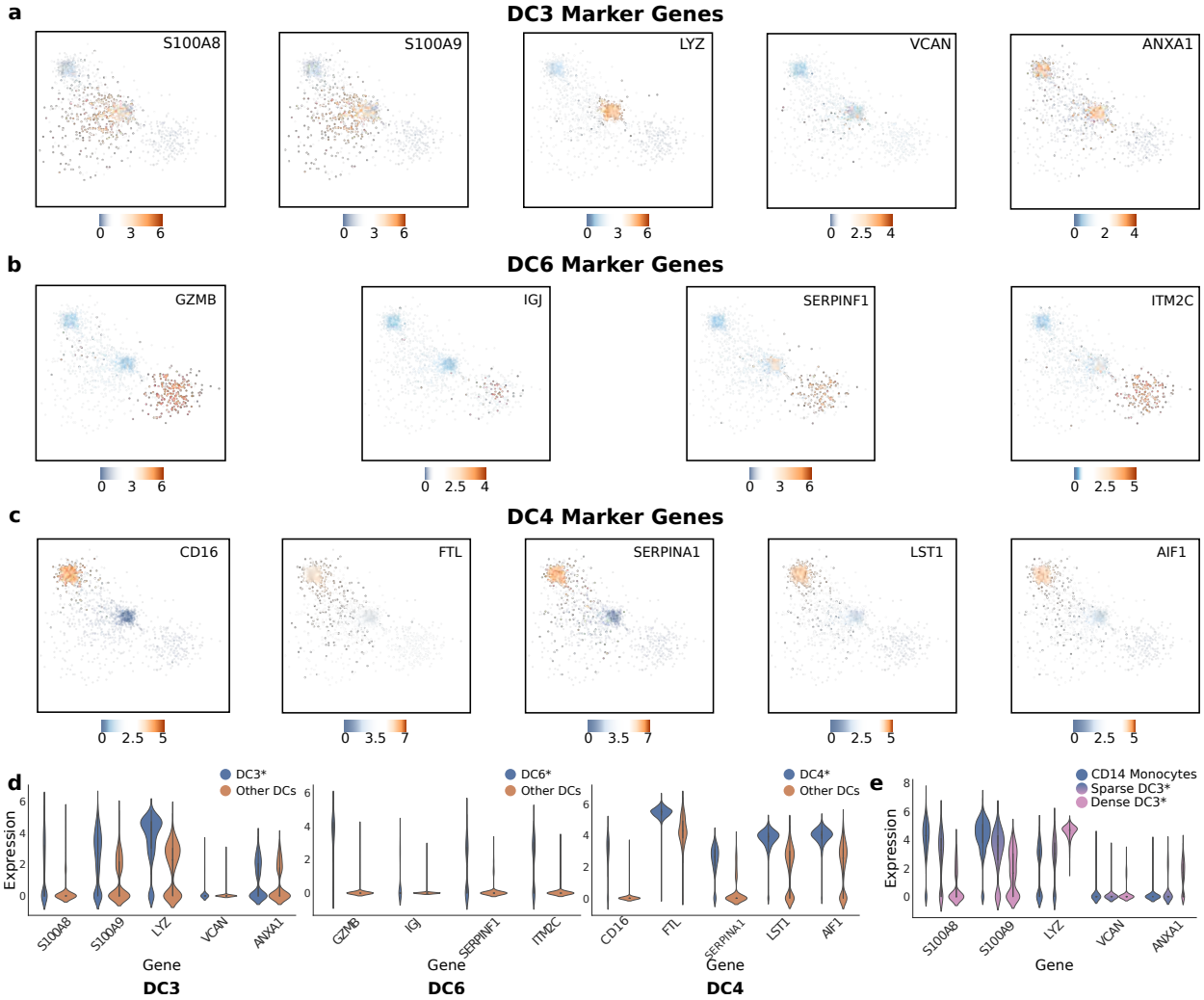

Figure G.15: Marker gene expressions for dendritic cell subtypes in the PBMC dataset. We plot gene expression heatmaps for the marker genes identified in the original study of PBMC2 for the dendritic cell (DC) subtypes (**a**) DC3, (**b**) DC6, and (**c**) DC4 on our densMAP visualization of the PBMC data, restricted to DCs. **d.** Violin plots showing the expression of marker genes identified by the PBMC2 study in our putative DC subtypes in the PBMC dataset: DC3 (left), DC6 (middle), and DC4 (right). The higher expression of these genes in our assigned subtypes and the expression patterns in (**a**) through (**c**) support our assignment of the DCs in the original PBMC dataset to the known subtypes in the PBMC2 dataset. The asterisk indicates our putative cell type assignment based on marker gene expression. **e.** Noting the existence of sparse and dense parts of the DC3 cluster in Figure 6.2, we compare the expression of DC3 and classical monocyte marker genes in the dense DCs (log local radius less than 3.9), sparse DCs (log local radius greater than 3.9), and classical monocytes; the violin plot indicates that sparse DCs are intermediate in expression of these marker genes between dense DCs and classical monocytes, potentially indicating a transition between the two states. The asterisk indicates our putative cell type assignment based on marker gene expression.

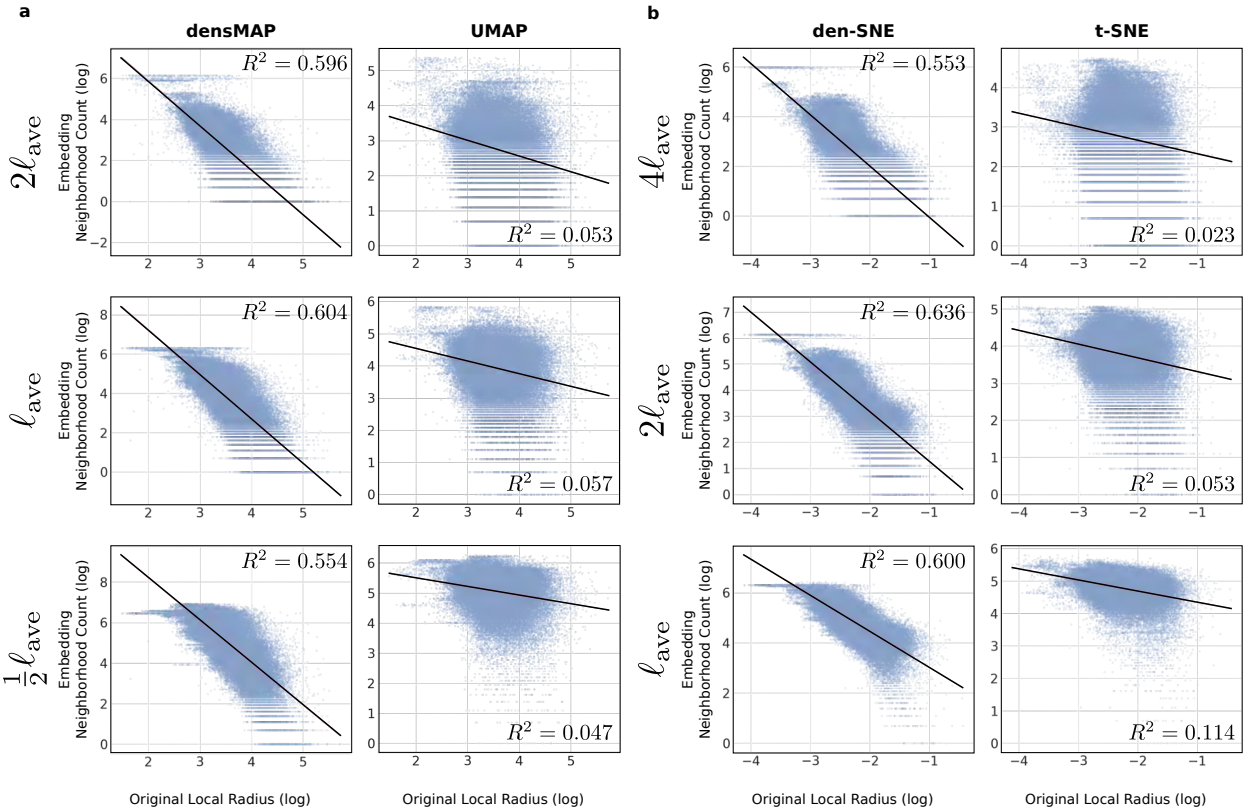

Figure G.16: Density-preserving methods preserve density robustly at different scales on *C. elegans* embryo development data based on neighborhood count. We compared the local radius of each point in the original *C. elegans* embryo development dataset to its neighborhood count in the visualizations (a measure of visual density; see Methods) for (**a**) densMAP and UMAP; and (**b**) den-SNE and t-SNE. We chose for each embedding, a length-scale ℓ_{ave} and multiples of that length-scale for which to compute the neighborhood counts (Methods): $\{\frac{1}{2}\ell_{\text{ave}}, \ell_{\text{ave}}, 2\ell_{\text{ave}}\}$ for densMAP and UMAP, and $\{\ell_{\text{ave}}, 2\ell_{\text{ave}}, 4\ell_{\text{ave}}\}$ for den-SNE and t-SNE. Since neighborhood count represents the density around a given point, a visualization that preserves density information will have higher neighborhood counts for points with smaller local radii in the original space. Note that this negative correlation is significantly stronger for our density-preserving tools than for t-SNE and UMAP, and this pattern holds across the different length-scales.

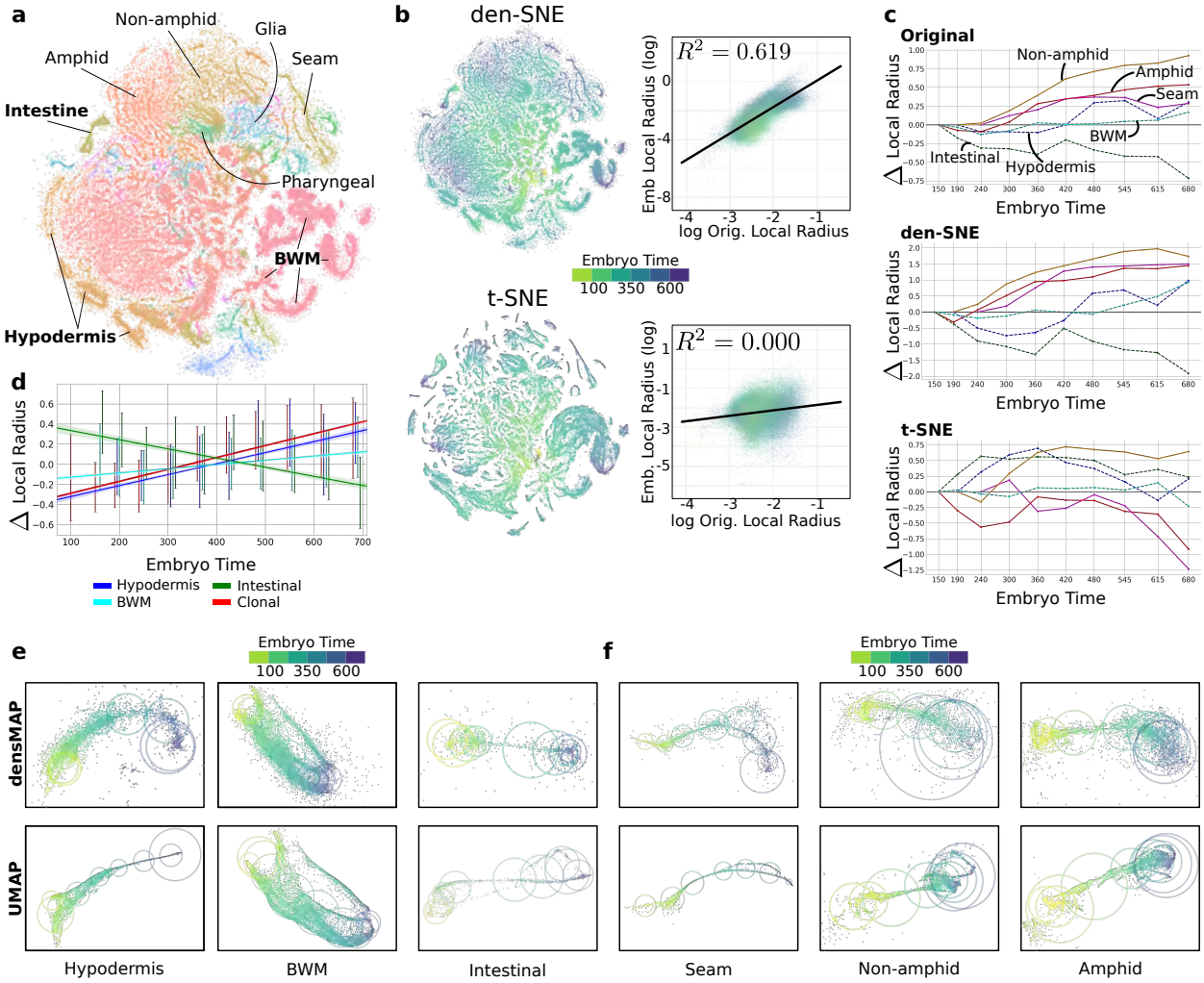

Figure G.17: Visualizing the *C. elegans* embryo development data with den-SNE recapitulates densMAP results. We repeat the analysis presented in Figure 6.3 using t-SNE and den-SNE. **a.** den-SNE embedding of dataset, colored by cell-type (t-SNE ommitted for space). **b.** Same data, now colored by embryo time, with den-SNE above and UMAP below. The scatter plots on the right compare the local radius, our measure of local density, in the original space, with the local radius in the embedding (Methods). Points are colored by embryo time, and the R^2 value of the correlation is shown for each plot. The higher correlation in den-SNE than in t-SNE supports the increase in transcriptomic variability over time seen in the den-SNE plot. Analogous correlation plots based on neighborhood count, our other measure of local distance (Methods), are included in Figure G.16. **c.** To assess lineage-specific patterns, we consider the mean local radius within different time bins for six different cell-types in the original data (top), den-SNE (middle), and t-SNE (bottom). In the original data, the plots illustrate the temporal changes in the underlying local density, while for den-SNE and t-SNE, they illustrate the apparent changes in density based on the visualizations. Time intervals were given in the original dataset, and the y-axis shows the change in average local radius compared to the earliest time interval in log-scale. Note that the trajectories traced out by den-SNE follow those of the original dataset more closely than those from t-SNE, supporting the relative temporal homogeneity seen in the den-SNE plots of semi-clonal cell types (hypodermis [7,746 cells], intestinal [1,732 cells], and BWM [17,520 cells]) compared to the other cell types. **d.** We show the best linear fit of local radius v. embryo time for the three semi-clonal cell-types against all clonal cells (59,026 cells), again aggregating cells within the time-intervals given in the publication, with error bars showing one standard deviation in local radius for all cells within the interval. The slope of the linear fit for clonal cells is significantly higher than those of the semi-clonal cells: 99% two-sided confidence intervals of the slope coefficients do not overlap. We show densMAP and UMAP plots restricted to the examples of semi-clonal (**e**) and clonal (**f**) cell types; circles are centered at the centroid of the points in each time bin, and radius indicates one standard deviation of these coordinates (both based on visualization coordinates). densMAP more accurately portrays that the variability of the semi-clonal cell types (**e**) is more homogeneous compared to the clonal cell types (**f**), whereas UMAP produces misleading visualizations.

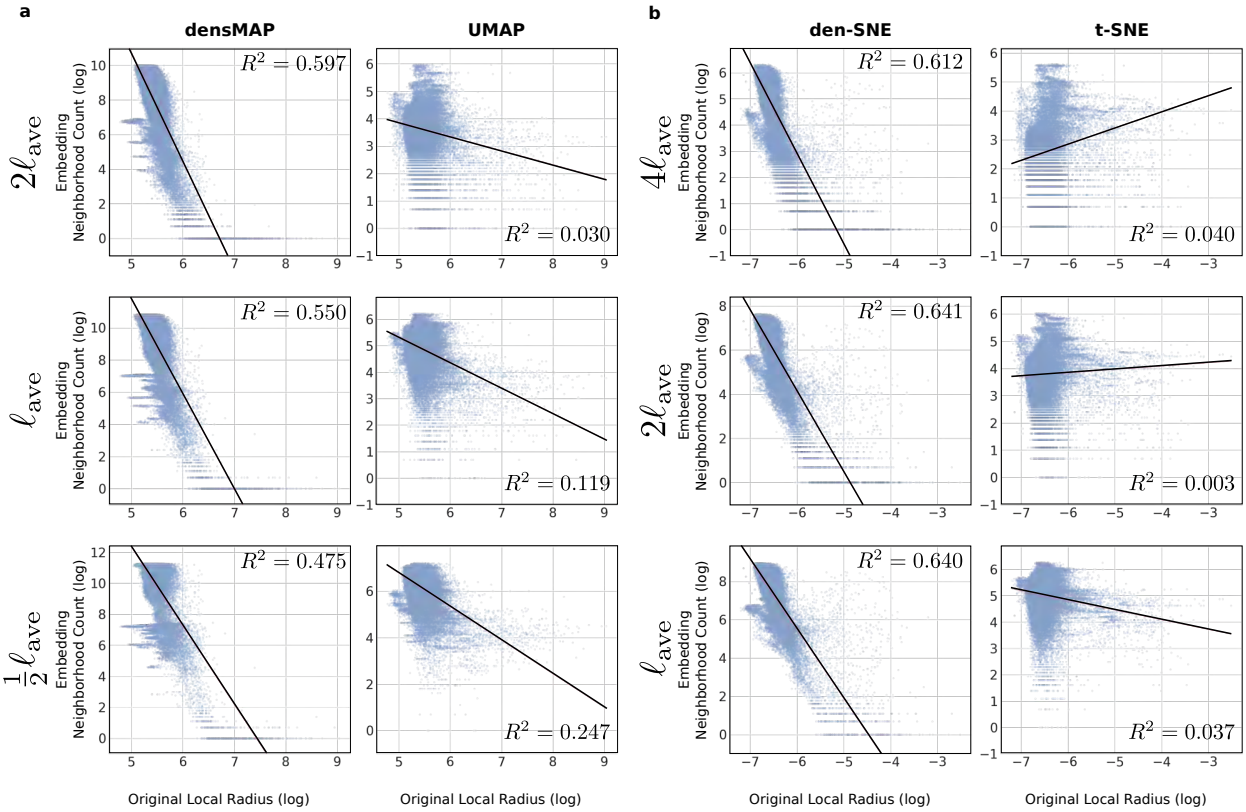

Figure G.18: Density-preserving methods preserve density robustly at different scales on UKBB data based on neighborhood count. We compared the local radius of each point in the original UKBB dataset (20% subsample) to its neighborhood count in the visualizations (a measure of visual density; see Methods) for (**a**) densMAP and UMAP; and (**b**) den-SNE and t-SNE. We chose for each embedding, a length-scale ℓ_{ave} and multiples of that length-scale for which to compute the neighborhood counts (Methods): $\{\frac{1}{2}\ell_{\text{ave}}, \ell_{\text{ave}}, 2\ell_{\text{ave}}\}$ for densMAP and UMAP, and $\{\ell_{\text{ave}}, 2\ell_{\text{ave}}, 4\ell_{\text{ave}}\}$ for den-SNE and t-SNE. Since neighborhood count represents the density around a given point, a visualization that preserves density information will have higher neighborhood counts for points with smaller local radii in the original space. Note that this negative correlation is significantly stronger for our density-preserving tools than for t-SNE and UMAP, and this pattern holds across the different length-scales.

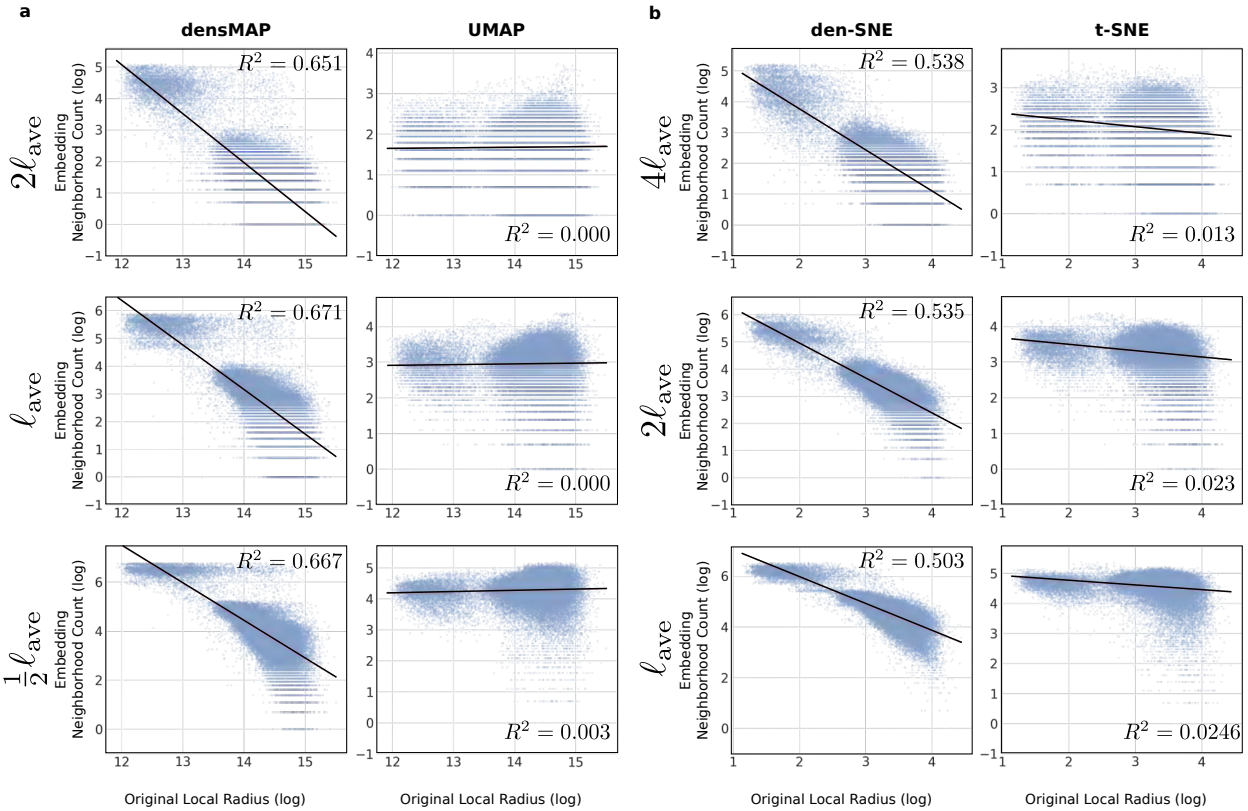

Figure G.19: Density-preserving methods preserve density robustly at different scales on MNIST data based on neighborhood count. We compared the local radius of each point in the original MNIST dataset to its neighborhood count in the visualizations (a measure of visual density; see Methods) for (**a**) densMAP and UMAP; and (**b**) den-SNE, and t-SNE. We chose for each embedding, a length-scale ℓ_{ave} and multiples of that length-scale for which to compute the neighborhood counts (Methods): $\{\frac{1}{2}\ell_{\text{ave}}, \ell_{\text{ave}}, 2\ell_{\text{ave}}\}$ for densMAP and UMAP, and $\{\ell_{\text{ave}}, 2\ell_{\text{ave}}, 4\ell_{\text{ave}}\}$ for den-SNE and t-SNE. Since neighborhood count represents the density around a given point, a visualization that preserves density information will have higher neighborhood counts for points with smaller local radii in the original space. Note that this negative correlation is significantly stronger for our density-preserving tools than for t-SNE and UMAP, and this pattern holds across the different length-scales.

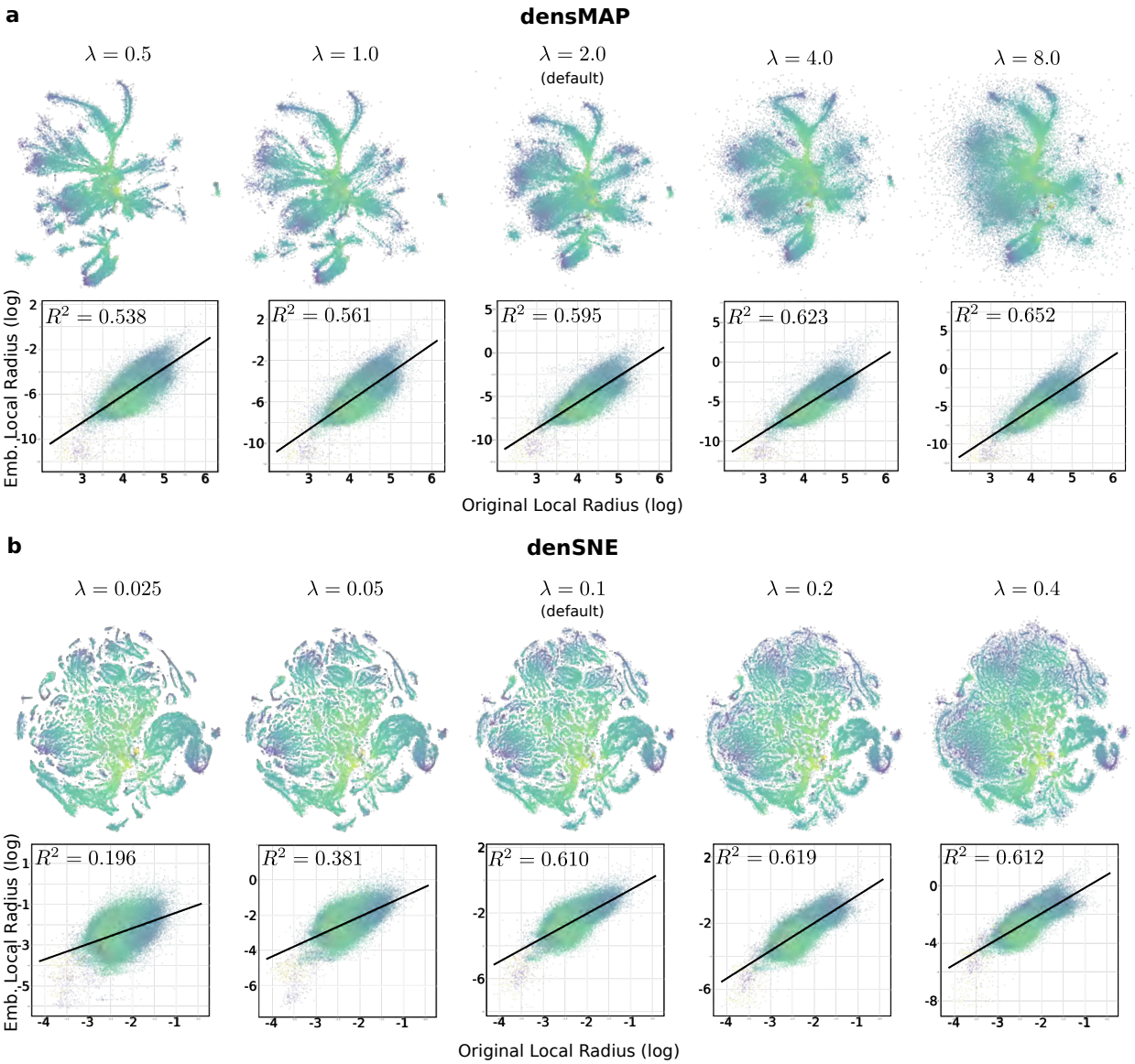

Figure G.20: Varying the density weight parameter in densMAP and den-SNE controls the trade-off between density preservation and cluster separation. We demonstrate on the *C. elegans* dataset the effects of varying the weight λ of the density-preservation term in the objective function of (**a**) densMAP and (**b**) den-SNE. As λ increases, so does the correlation between the log local radii in the original data and the embedding, but the clusters begin to fade into each other, likely due to lack of space in the visualization. As λ decreases, the correlation becomes worse and the embeddings become closer to those of t-SNE and UMAP. Based on our results from a wide range of datasets, we recommend the default values of $\lambda = 0.1$ for den-SNE and $\lambda = 2.0$ for densMAP.

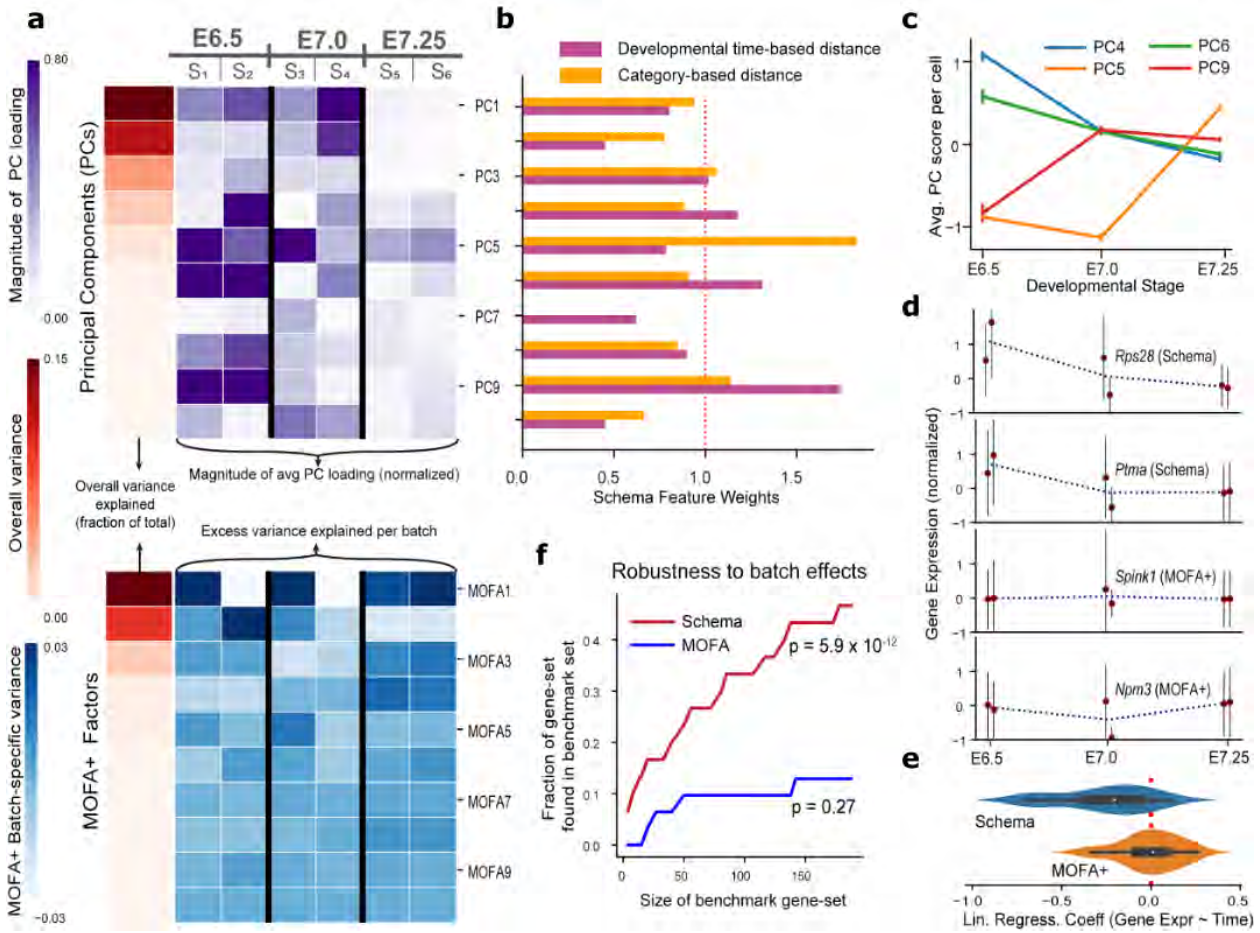

Figure G.21: Batch-effect adjusted identification of differentially expressed genes along a developmental time course. **a.** We obtained a dataset of developing mouse epiblast cells spanning three timepoints, with two experimental batches per timepoint. PCA and MOFA+ components show significant within-timepoint variability. In this panel, loadings of each principal component (PC) were normalized to zero mean and unit standard deviation. **b, c.** Weights computed by Schema after accounting for batch effects and developmental age with two different distance metrics, one that provides Schema with temporal-ordering and another that does not provide this order. When incorporating order information, Schema down-weights PC5, which shows substantial within-timepoint, batch-related variability, and up-weights PC9, which has higher correlation with time. Correspondingly identified PCs reflect the effect of these metric. **d, e.** Schema identifies genes with monotonically changing expression. For each gene identified by Schema or MOFA+, we regressed its expression (normalized to zero mean and unit standard deviation) against developmental time, encoding stages E6.25, E7.0 and E7.25 as timepoints 1, 2 and 3, respectively. Consistent with stage-dependent monotonicity in expression, the fitted slopes for Schema genes were significantly different from zero (two-sided t-test, $p = 3.83 \times 10^{-6}$); this was not true of MOFA+ ($p = 0.77$). **f.** Schema has stronger overlap with batch-effect adjusted benchmark sets of differentially expressed genes (hypergeometric test with Bonferroni correction, $p = 5.9 \times 10^{-12}$ for the benchmark set of size 188).

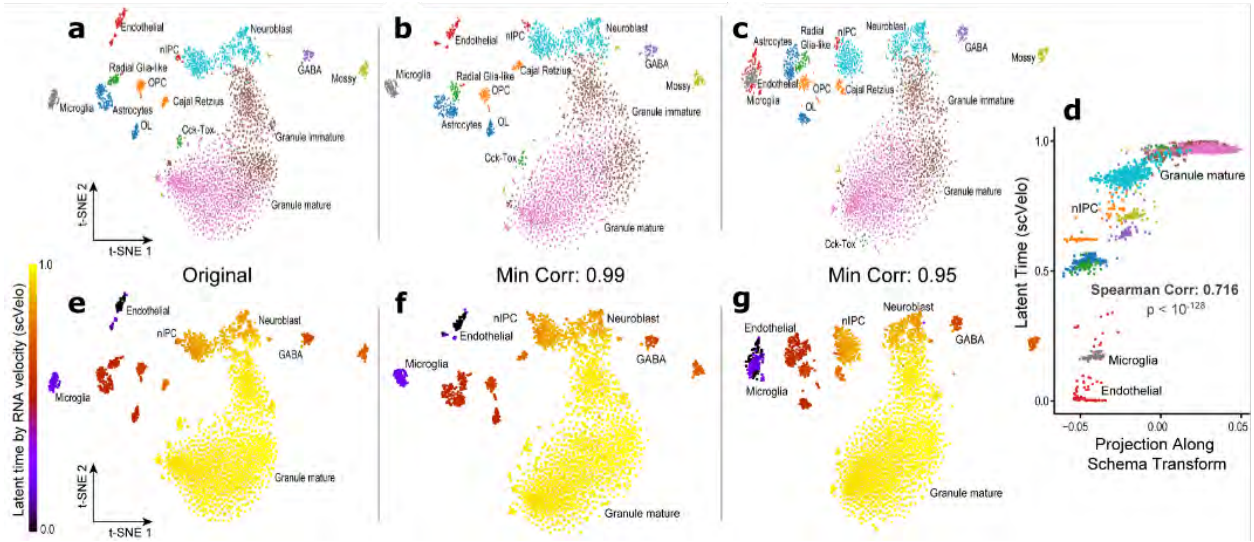

Figure G.22: Synthesis of spliced and unspliced mRNA counts recovers RNA velocity and enables informative visualization. a. t-SNE visualization of the spliced mRNA counts **b, c.** We synthesized spliced and unspliced mRNA counts, with the former as the primary and the latter as the secondary modality, respectively. Schema's transformation picks up the time derivative of gene expression, thus accentuating the cell differentiation process. t-SNE visualizations of synthesized data with 0.99 and 0.95 minimum correlation, respectively, are shown. **d.** Schema's results are in agreement with the RNA velocity tool, scVelo. By measuring each cell's Schema transformation, we computed a pseudotime estimate which we found to be significantly correlated with scVelo's latent-time estimate (Spearman rank correlation 0.716, two-sided t-test $p < 10^{-128}$). **e–g.** Same t-SNE visualizations as above, but with cells colored by their scVelo latent-time, showing that Schema puts cells at similar differentiation stages progressively closer.

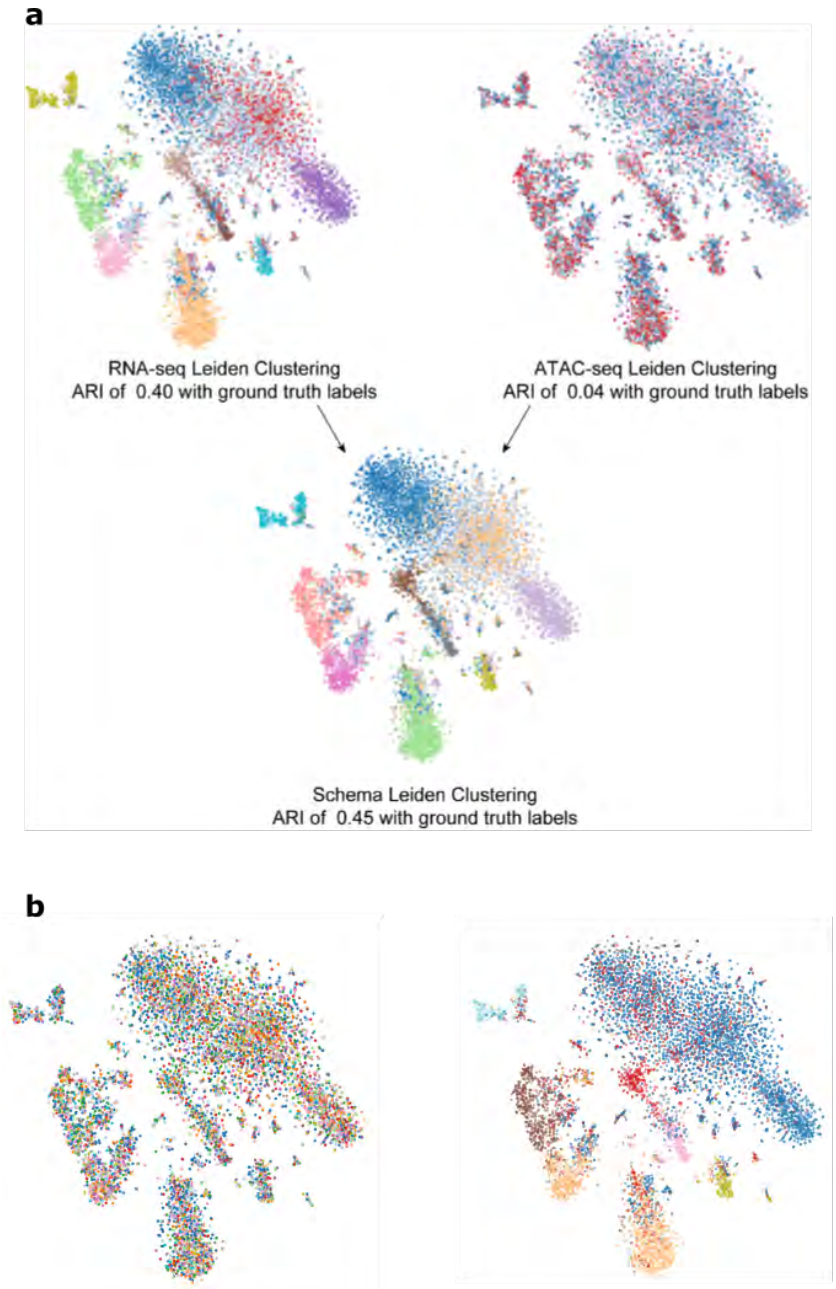

Figure G.23: a. Leiden clustering for RNA-seq and ATAC-seq data individually and Schema's synthesis. **b.** Leiden clustering for totalVI and CCA synthesis

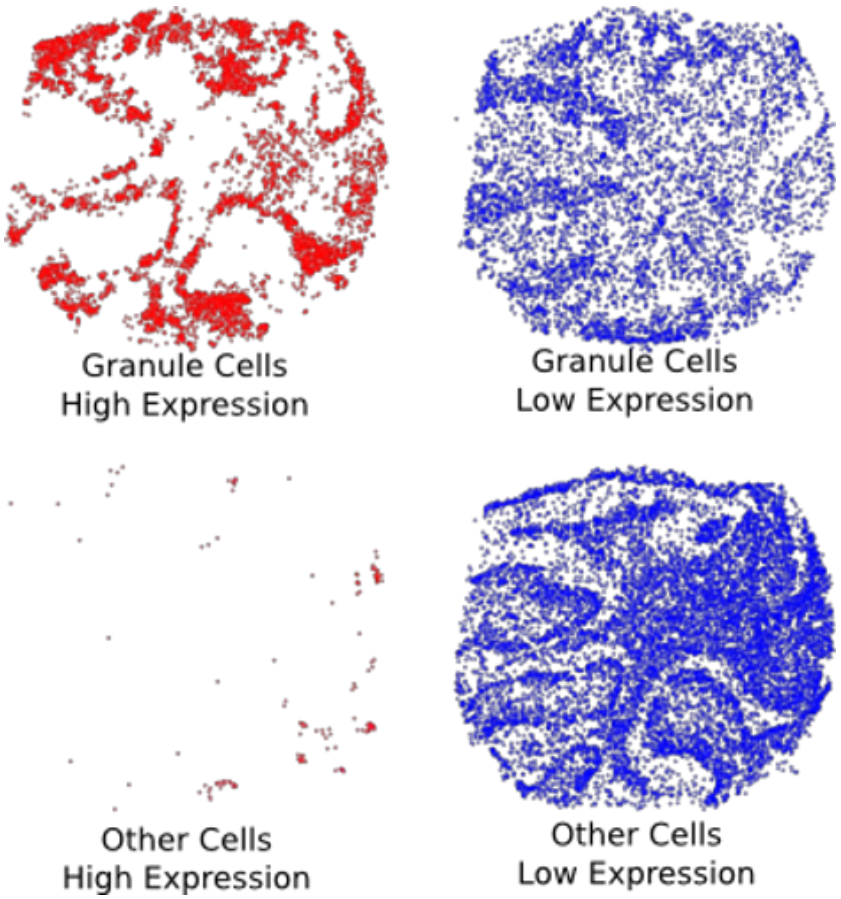

Figure G.24: Visually, CCA seems to be effective at identifying a gene set that is differentially expressed only in densely-located granule neurons. The Slide-seq sample used here (Puck 180430_1) is the same as in Figure 5.3d. However, as shown in Figure 5.3e, the gene ranking computed by Schema is better preserved across three Slide-Seq samples than those produced by CCA (median sample-pair Spearman rank correlation of 0.675 and 0.457, respectively)

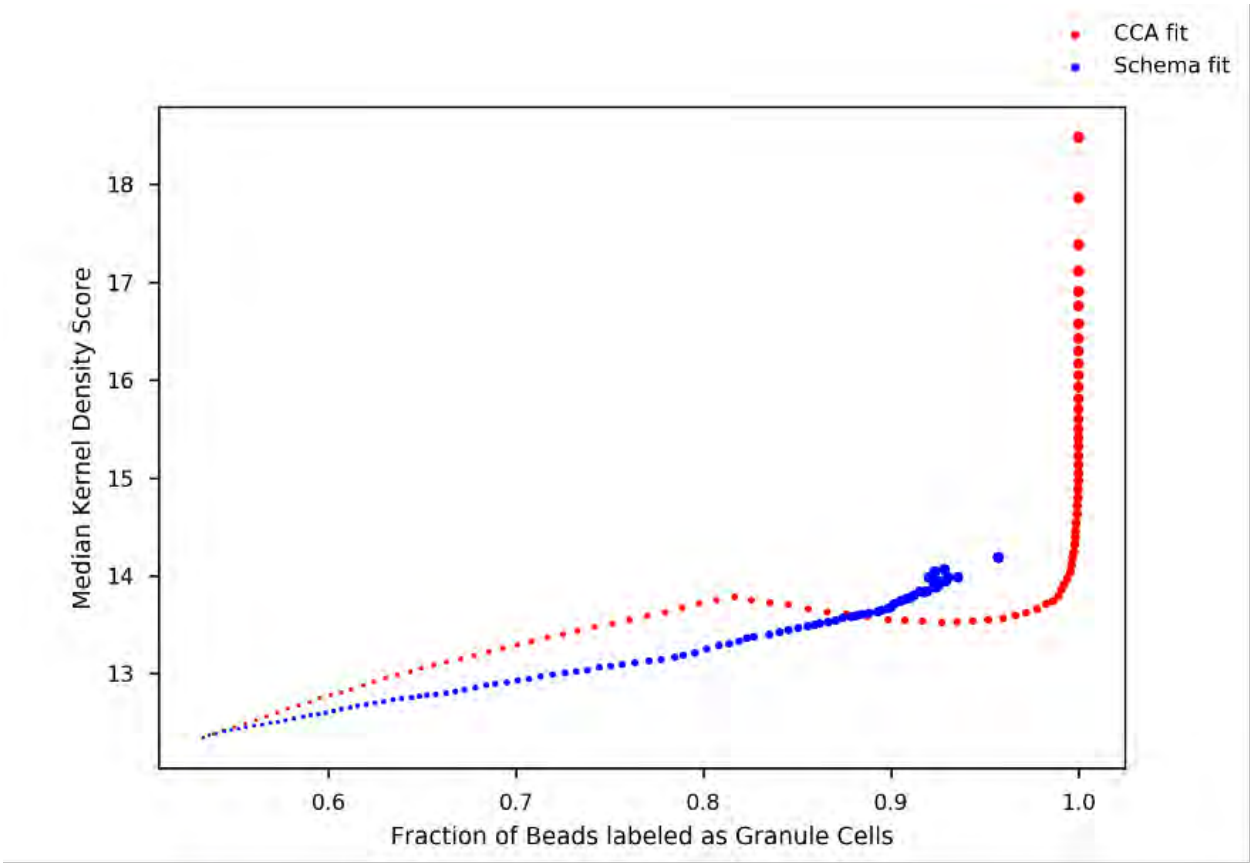

Figure G.25: Investigation of CCA and Schema cell loadings. We sorted Slideseq transcriptomes by their loading on the Schema-implied gene scores and investigated how the exposure to secondary modalities (cell-type labels and spatial density) varied in this ordering; we repeated the analysis for CCA cell loadings. For $k = 1, \ldots, 99$, we selected cells with loadings in the percentile range $[k, 100]$ and computed the frequency of granule-cell labels and the average Gaussian kernel density score of a cell in this set; higher values of these measures indicate stronger agreement with the cell type and spatial density modalities, respectively. In the plot, the size of a point is proportional to k. For both Schema and CCA, the higher cell loadings typically correlate with a higher granule-cell fraction and higher kernel density, as both the methods transform the primary gene-expression modality to align it with the secondary modalities. However, for Schema this relationship plateaus after a point because Schema's regularization mechanism limits the distortion of the primary modality, constraining the extent of match with the secondary modalities. In contrast, the unconstrained framework of CCA produces loadings where the 99th percentile cell loading has significantly higher spatial density exposure than the 95th percentile. This may lead to overfitting as CCA computes gene rankings that are overly determined by sample-specific artifacts. In contrast, the regularization mechanism of Schema produces gene rankings that are better preserved across samples.

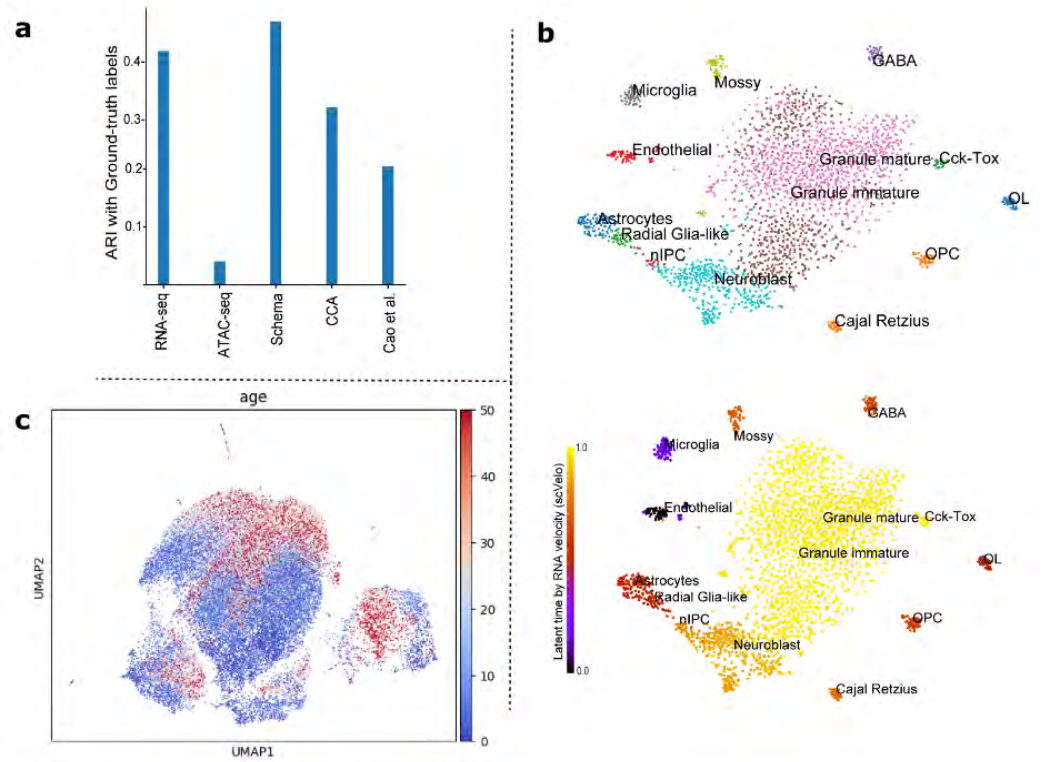

Figure G.26: Evaluation of canonical correlation analysis (CCA) performance. a. Inferring cell types by synthesizing RNA-seq and ATAC-seq data. The metric of evaluation here is the agreement between Leiden clustering on the synthesized dataset and ground truth cell-type labels, measured using the adjusted Rand index (ARI), with higher scores indicating greater agreement. This panel contains the same information as Figure 5.1e and is reproduced here for convenience. **b.** Inferring RNA velocity by synthesizing spliced and unspliced mRNA counts. Spliced and unspliced data were correlated using CCA and the synthesized data was visualized with t-SNE. The bottom half of the panel colors cells by their scVelo latent-time estimate. This panel should be compared with Figure G.22b–f , which display the corresponding plots produced by Schema synthesis of the data. CCA's synthesis does not place cells with similar stages of differentiation as closely together as Schema. Quantitatively, the Spearman rank correlation between t-SNE distances and latent-time difference is 0.163 for CCA, less than the correlation achieved using just the spliced mRNA counts (0.397); in contrast, the Schema transformation corresponding to a minimum correlation constraint of 0.95 results in a correlation of 0.432. **c.** Schema highlights secondary patterns while preserving primary structure. RNA-seq data was synthesized with cell age metadata using CCA. Compared to a synthesis by Schema (Figure 5.2b–d), the CCA-based visualization less clearly communicates the developmental trajectory. We quantified the age-related structure in the transformed dataset by a diffusion pseudotime analysis. The Spearman rank correlation between the pseudotime estimate and the ground-truth cell age is only 0.059 for the CCA-synthesized data while it 0.365 in the original, untransformed dataset and 0.436 in the Schema-transformed dataset corresponding to a minimum correlation constraint of 0.99.

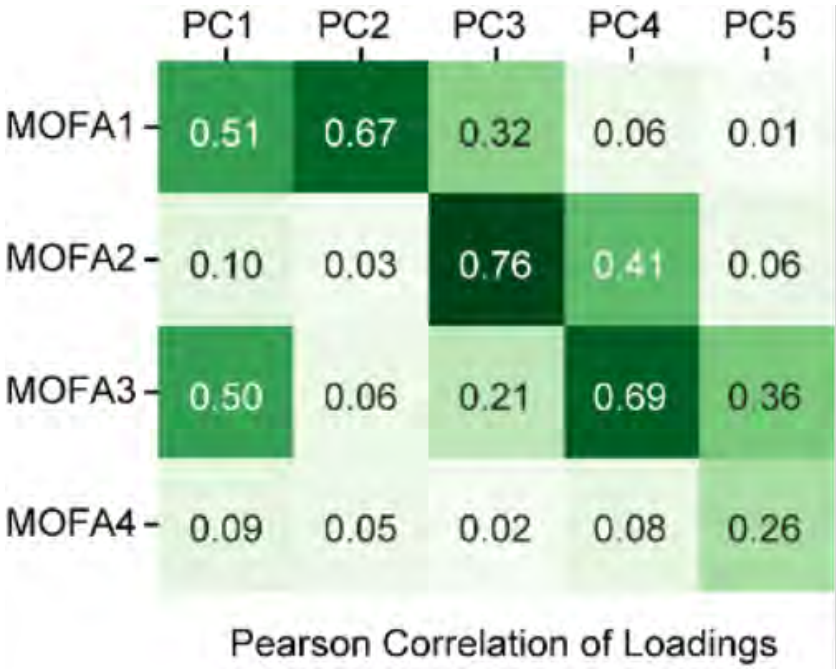

Figure G.27: Correlation of factor loadings between MOFA+ factors and principal components.

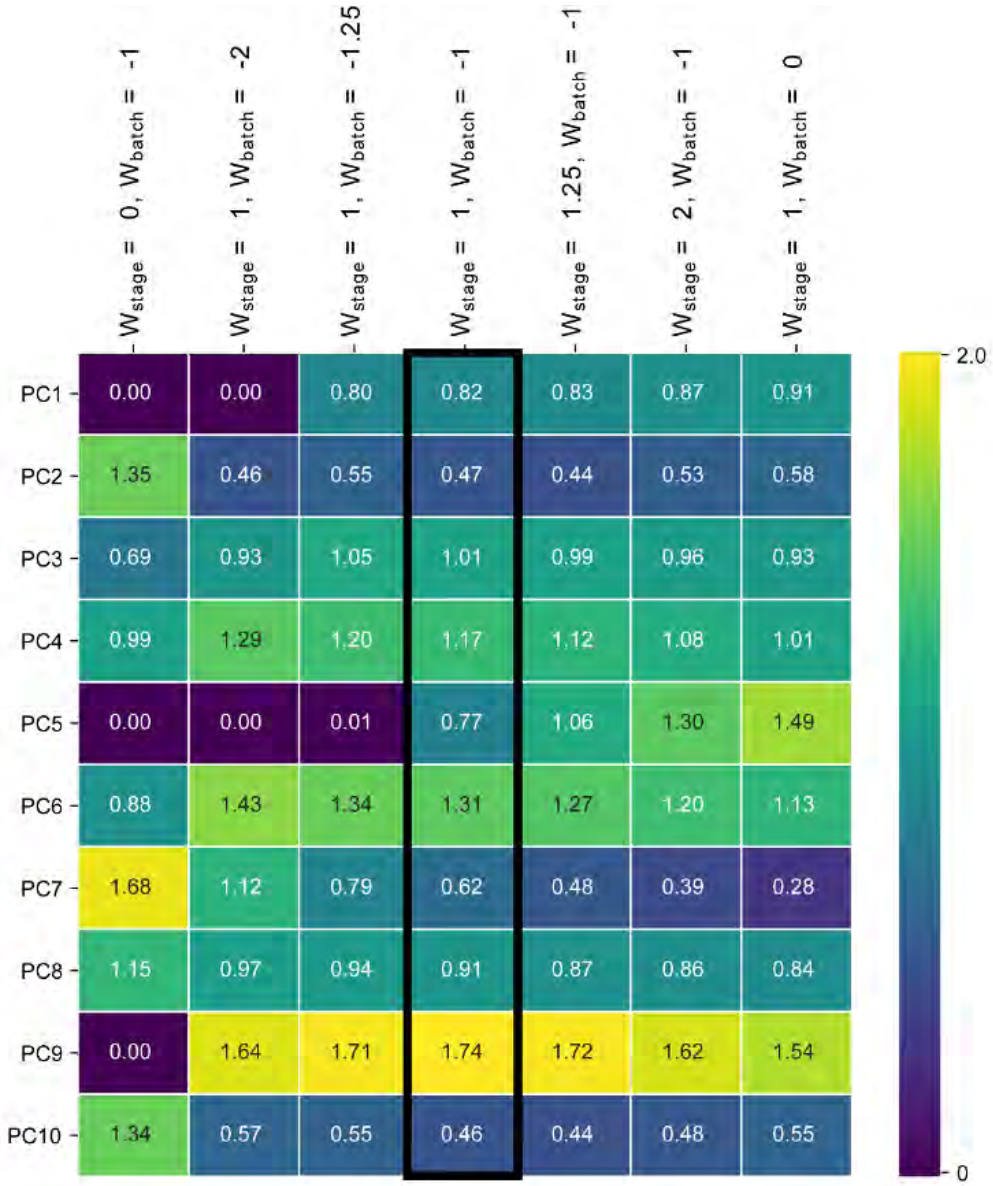

Figure G.28: Differential expression analysis while accounting for batch effects and developmental stage. Schema feature-selection results for different weights of the developmental-stage and batch-effect modalities. The middle column is the one shown in the main text: equal (and opposite) weights for the batch-effects and developmental-stage modalities. The left-most column corresponds to using only the batch-effect modality while the right-most column corresponds to using only the developmental-stage modality.

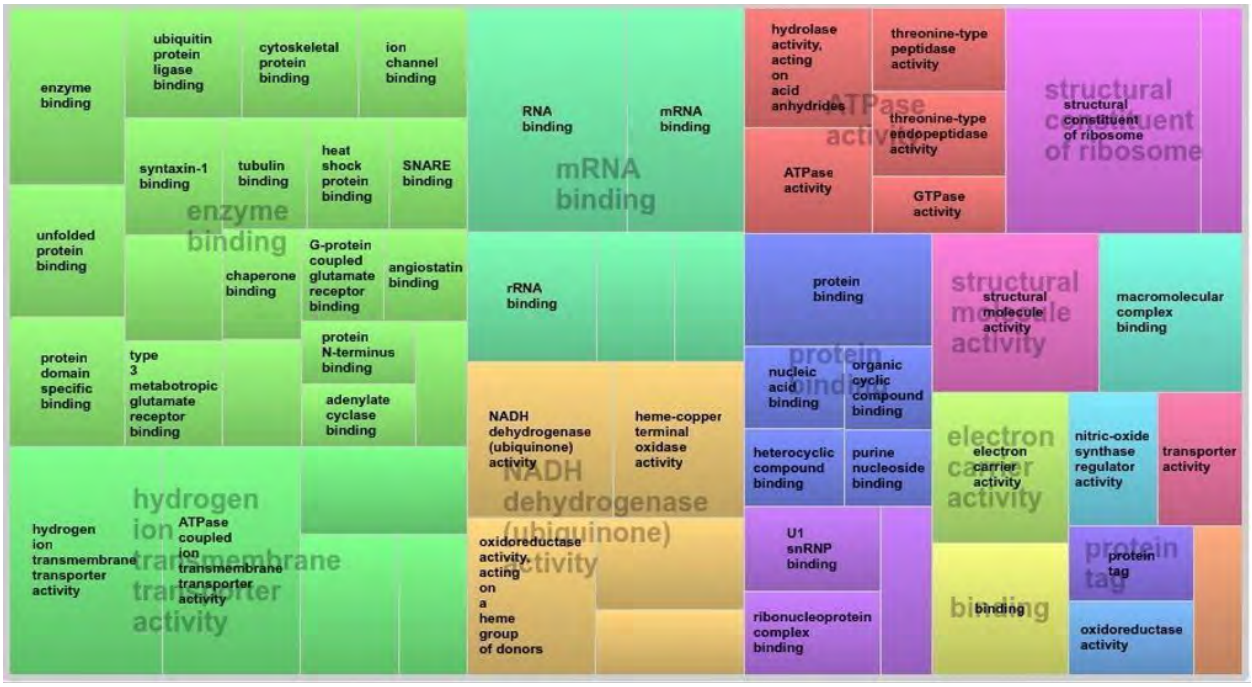

Figure G.29: Visualization of enriched GO terms in Schema-ranked genes across 3 samples of Slide-Seq data. (via REViGO [167], using $\lfloor \log_{10} p \rfloor$)

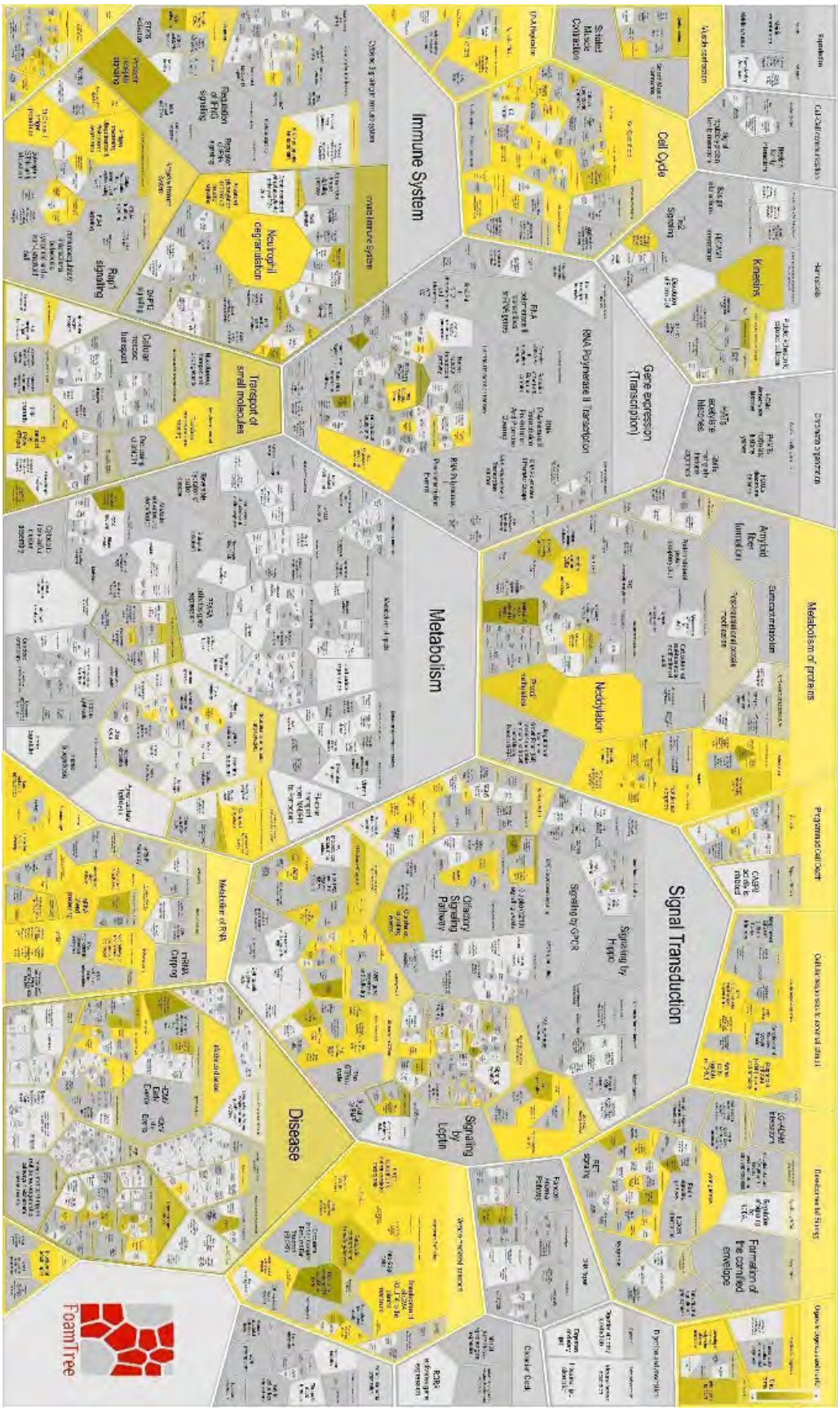

Figure G.30: Voronoi-tessellation visualization of REACTOME pathways enriched in Schema-ranked genes across 3 samples of Slide-Seq data

H Supplementary Tables

Table H.1: Runtime comparison of Schema with CCA, SpatialDE and Trendsceek. The values above were averaged over the three previously described mouse cerebellum samples from the Slide-seq dataset. All programs were run on a Linux server with 24 Intel Xeon 2.40 GHz cores and 386 GB RAM. Each program was allowed to use as many cores as were available; Schema, CCA and SpatialDE did so, but Trendsceek did not. The server is a shared resource and while we did periodically check it to confirm that ample system resources were available, the runtime estimates above may be influenced by the load from other programs. For Schema, the runtime includes the time for preprocessing and encompasses the complete ensemble of sub-runs on different parameter choices. The runtime for CCA also encompasses the entire pipeline: pairwise modality combinations and then a final integration. For Schema and CCA, we were able to use all the data of each sample. For SpatialDE and Trendsceek, we experimented with small subsets of data and increased the subset size until the demand on the shared resource (server) became infeasible. The subset of cells for SpatialDE and Trendsceek were randomly chosen, with an equal split between granule and non-granule cells; for both, genes were selected based on high expression variability.

Program	**Average per sample**		
	# of cells	# of genes	Runtime (min)
Schema	20823	17607	34
CCA	20823	17607	50
SpatialDE	16000	9000	244
Trendsceek	2000	3000	338

Gene Expression in Tumor vs. Blood (T Cells CD8)

					Permutation Test p-value (Bonferroni-corrected)		
Rank	Gene	Δ(Variance)	Δ(Dispersion)	Δ(Mean)	Dispersion	Mean	Variance
1	DUSP4	1.034	0.492	0.919	**<2E-4**	**<2E-4**	**<2E-4**
2	RGS1	0.995	0.009	0.926	1.0	**<2E-4**	**<2E-4**
3	RGCC	0.922	0.045	1.024	1.0	**<2E-4**	**<2E-4**
4	TNFAIP3	0.816	-0.173	0.970	1.0	**<2E-4**	**<2E-4**
5	NR4A2	0.800	0.023	0.933	1.0	**<2E-4**	**<2E-4**
6	ZFP36	0.792	0.128	0.864	0.1346	**<2E-4**	**<2E-4**
7	CCL4	0.783	0.619	0.358	**<2E-4**	**<2E-4**	**<2E-4**
8	CREM	0.764	-0.201	0.811	1.0	**<2E-4**	**<2E-4**
9	JUNB	0.742	0.116	0.736	0.676	**<2E-4**	**<2E-4**
10	HSP90AA1	0.701	0.252	0.605	**<2E-4**	**<2E-4**	**<2E-4**
11	DNAJB1	0.693	0.055	0.616	1.0	**<2E-4**	**<2E-4**
12	FOSB	0.678	0.594	0.565	**<2E-4**	**<2E-4**	**<2E-4**
13	RPS26	0.675	0.199	0.516	**<2E-4**	**<2E-4**	**<2E-4**
14	ZNF331	0.670	0.456	0.641	**<2E-4**	**<2E-4**	**<2E-4**
15	JUND	0.656	-0.027	0.906	1.0	**<2E-4**	**<2E-4**
16	FTH1	0.588	0.141	0.618	0.0516	**<2E-4**	**<2E-4**
17	FOSL2	0.585	0.008	0.599	1.0	**<2E-4**	**<2E-4**
18	IGKC	0.579	1.135	0.335	**<2E-4**	**<2E-4**	**<2E-4**
19	YPEL5	0.569	0.193	0.575	**0.005**	**<2E-4**	**<2E-4**
20	TSC22D3	0.560	0.231	0.476	**<2E-4**	**<2E-4**	**<2E-4**

Table H.2: Genes with largest difference in variance between blood and tumor CD8 T cells. The columns Δ {Dispersion, Mean, Variance} show the changes in the corresponding statistic for each listed gene in tumor (1,621 cells) relative to blood (443 cells). We performed one-sided permutation tests (last three columns) to calculate the significance of the change in mean, dispersion, and variance, and those p-values which are significant after Bonferroni correction ($p < 0.01$) are shown in boldface. Zero p-values are shown as <2E-4, which is the smallest possible non-zero p-value we could obtain based on 100k permutation trials after Bonferroni correction.

Gene Expression in Tumor vs. Blood (T Cells CD4 Memory Resting)

					Permutation Test p-value (Bonferroni-corrected)		
Rank	Gene	Δ(Variance)	Δ(Dispersion)	Δ(Mean)	Dispersion	Mean	Variance
1	RGS1	1.013	-0.009	1.069	1.0	**<2E-4**	**<2E-4**
2	DUSP4	0.952	0.295	0.753	**<2E-4**	**<2E-4**	**<2E-4**
3	TNFAIP3	0.818	-0.205	1.099	1.0	**<2E-4**	**<2E-4**
4	JUNB	0.810	-0.070	1.110	1.0	**<2E-4**	**<2E-4**
5	ZFP36	0.809	0.039	1.045	1.0	**<2E-4**	**<2E-4**
6	RGCC	0.779	0.101	0.922	0.0124	**<2E-4**	**<2E-4**
7	CREM	0.762	0.111	0.937	0.1886	**<2E-4**	**<2E-4**
8	HSP90AA1	0.730	0.185	0.795	**<2E-4**	**<2E-4**	**<2E-4**
9	NR4A2	0.669	0.216	0.737	**<2E-4**	**<2E-4**	**<2E-4**
10	HSPA1A	0.664	0.911	0.367	**<2E-4**	**<2E-4**	**<2E-4**
11	LMNA	0.656	0.455	0.556	**<2E-4**	**<2E-4**	**<2E-4**
12	DNAJB1	0.610	0.201	0.566	**<2E-4**	**<2E-4**	**<2E-4**
13	JUND	0.598	-0.049	0.979	1.0	**<2E-4**	**<2E-4**
14	FTH1	0.583	0.041	1.067	1.0	**<2E-4**	**<2E-4**
15	SLC2A3	0.561	0.091	0.631	0.0196	**<2E-4**	**<2E-4**
16	ZNF331	0.560	0.029	0.586	1.0	**<2E-4**	**<2E-4**
17	YPEL5	0.557	0.052	0.657	1.0	**<2E-4**	**<2E-4**
18	HSPH1	0.553	0.297	0.493	**<2E-4**	**<2E-4**	**<2E-4**
19	TSC22D3	0.549	0.213	0.534	**<2E-4**	**<2E-4**	**<2E-4**
20	FOSL2	0.541	0.211	0.612	**<2E-4**	**<2E-4**	**<2E-4**

Table H.3: Genes with largest difference in variance between blood and tumor cells) memory resting CD4 T cells. The columns Δ {Dispersion, Mean, Variance} show the changes in the corresponding statistic for each listed gene in tumor (9,019 cells) relative to blood (1,036 cells). We performed one-sided permutation tests (last three columns) to calculate the significance of the change in mean, dispersion, and variance, and those p-values which are significant after Bonferroni correction ($p < 0.01$) are shown in boldface. Zero p-values are shown as <2E-4, which is the smallest possible non-zero p-value we could obtain based on 100k permutation trials after Bonferroni correction.

Table H.4: Genes with largest difference in variance between blood and tumor CD4 naïve T cells. The columns Δ {Dispersion, Mean, Variance} show the changes in the corresponding statistic for each listed gene in tumor (61 cells) relative to blood (437 cells). We performed one-sided permutation tests (last three columns) to calculate the significance of the change in mean, dispersion, and variance, and those p-values which are significant after Bonferroni correction ($p < 0.01$) are shown in boldface. Zero p-values are shown as <2E-4, which is the smallest possible non-zero p-value we could obtain based on 100k permutation trials after Bonferroni correction.

Gene Expression in Tumor vs. Blood (T Cells Naïve CD4)

					Permutation Test p-value (Bonferroni-corrected)		
Rank	Gene	Δ(Variance)	Δ(Dispersion)	Δ(Mean)	Dispersion	Mean	Variance
1	MT-ATP6	1.143	0.416	-0.331	**<2E-4**	**0.0078**	**<2E-4**
2	RPS26	1.114	0.047	0.97	1.0	**<2E-4**	**<2E-4**
3	FTH1	1.052	0.432	0.787	**<2E-4**	**<2E-4**	**<2E-4**
4	RPS14	0.899	0.404	-0.048	**<2E-4**	1.0	**<2E-4**
5	CREM	0.888	0.628	0.66	1.0	**<2E-4**	**<2E-4**
6	MT-RNR2	0.887	0.253	0.82	0.0128	**<2E-4**	**<2E-4**
7	RPLP1	0.833	0.372	0.007	**<2E-4**	1.0	**<2E-4**
8	RPS27	0.832	0.299	-0.097	**0.0046**	1.0	**<2E-4**
9	RPS3	0.828	0.457	-0.198	**<2E-4**	0.497	**<2E-4**
10	MTRNR2L12	0.82	0.274	0.797	0.0194	**<2E-4**	**<2E-4**
11	CXCR4	0.814	0.366	0.661	0.166	**<2E-4**	**<2E-4**
12	SRGN	0.81	0.377	0.678	0.129	**<2E-4**	**<2E-4**
13	MT-ND1	0.762	0.315	-0.155	**0.0042**	1.0	**<2E-4**
14	RPL34	0.754	0.361	-0.426	**<2E-4**	**<2E-4**	**<2E-4**
15	TTC19	0.753	0.414	0.026	**<2E-4**	1.0	**<2E-4**
16	PABPC1	0.744	0.481	-0.099	**<2E-4**	1.0	**<2E-4**
17	MT-CO2	0.733	0.307	0.407	**0.0084**	**<2E-4**	**<2E-4**
18	RPL11	0.721	0.469	-0.601	**<2E-4**	**<2E-4**	**<2E-4**
19	MT-CO1	0.719	0.391	-0.118	**<2E-4**	1.0	**<2E-4**
20	RPL27A	0.716	0.387	-0.289	**<2E-4**	0.0138	**<2E-4**

Table H.5: Genes with largest difference in variance between blood and tumor memory B cells. The columns Δ {Dispersion, Mean, Variance} show the changes in the corresponding statistic for each listed gene in tumor (4,811 cells) relative to blood (67 cells). N/A values in the Δ(Dispersion) and Dispersion p-value columns indicate that the gene had zero mean-expression in blood, and so dispersion is undefined. We performed one-sided permutation tests (last three columns) to calculate the significance of the change in mean, dispersion, and variance, and those p-values which are significant after Bonferroni correction ($p < 0.01$) are shown in boldface. Zero p-values are shown as <2E-4, which is the smallest possible non-zero p-value we could obtain based on 100k permutation trials after Bonferroni correction.

Gene Expression in Tumor vs. Blood (B Cells Memory)

					Permutation Test p-value (Bonferroni-corrected)		
Rank	Gene	Δ(Variance)	Δ(Dispersion)	Δ(Mean)	Dispersion	Mean	Variance
1	RGS1	0.882	0.644	0.764	**<2E-4**	**<2E-4**	**<2E-4**
2	NR4A2	0.839	N/A	1.022	N/A	**<2E-4**	**<2E-4**
3	JUNB	0.742	-0.074	0.896	1.0	**<2E-4**	**<2E-4**
4	CD83	0.674	0.230	0.771	0.224	**<2E-4**	**<2E-4**
5	RGS2	0.647	-0.177	0.711	1.0	**<2E-4**	**<2E-4**
6	HSP90AA1	0.644	0.227	0.682	0.3564	**<2E-4**	**<2E-4**
7	FOSB	0.635	N/A	0.667	N/A	**<2E-4**	**<2E-4**
8	JUND	0.598	0.151	0.893	1.0	**<2E-4**	**<2E-4**
9	GPR183	0.586	0.366	0.539	0.0246	**<2E-4**	**<2E-4**
10	HSPH1	0.565	0.244	0.555	0.3082	**<2E-4**	**<2E-4**
11	ZNF331	0.560	0.489	0.557	**<2E-4**	**<2E-4**	**<2E-4**
12	SRGN	0.541	0.047	0.619	1.0	**<2E-4**	**<2E-4**
13	DUSP4	0.529	N/A	0.404	N/A	**<2E-4**	**<2E-4**
14	TSC22D3	0.515	-0.158	0.794	1.0	**<2E-4**	**<2E-4**
15	NR4A3	0.512	N/A	0.454	N/A	**<2E-4**	**<2E-4**
16	HSPA1A	0.507	0.957	0.350	**<2E-4**	**<2E-4**	**<2E-4**
17	SLC2A3	0.504	-0.317	0.569	1.0	**<2E-4**	**<2E-4**
18	LY9	0.503	0.098	0.569	1.0	**<2E-4**	**<2E-4**
19	HSP90AB1	0.499	0.122	0.573	1.0	**<2E-4**	**<2E-4**
20	MCL1	0.496	0.215	0.492	0.3442	**<2E-4**	**<2E-4**

Gene Expression in Tumor vs. Blood (B Cells Naïve)

					Permutation Test p-value (Bonferroni-corrected)		
Rank	Gene	Δ(Variance)	Δ(Dispersion)	Δ(Mean)	Dispersion	Mean	Variance
1	JUNB	0.967	0.264	1.094	0.322	**<2E-4**	**<2E-4**
2	CD83	0.852	-0.013	0.937	1.0	**<2E-4**	**<2E-4**
3	RPS27	0.827	0.329	-0.239	**<2E-4**	0.6354	**<2E-4**
4	NR4A2	0.811	N/A	0.876	N/A	**<2E-4**	**<2E-4**
5	JUND	0.681	-0.087	1.025	1.0	**<2E-4**	**<2E-4**
6	FOS	0.660	N/A	0.515	N/A	**<2E-4**	**<2E-4**
7	FOSB	0.652	0.320	0.573	0.4072	**<2E-4**	**<2E-4**
8	RPL13A	0.634	0.287	-0.362	**<2E-4**	**0.008**	**<2E-4**
9	DUSP1	0.629	-0.188	0.632	1.0	**<2E-4**	**<2E-4**
10	IGHM	0.625	0.506	-0.928	**<2E-4**	**<2E-4**	**<2E-4**
11	RGS1	0.620	N/A	0.416	N/A	**<2E-4**	**<2E-4**
12	ZNF331	0.616	-0.140	0.584	1.0	**<2E-4**	**<2E-4**
13	TSC22D3	0.609	0.048	0.715	1.0	**<2E-4**	**<2E-4**
14	JUN	0.590	-0.875	0.498	1.0	**<2E-4**	**<2E-4**
15	HERPUD1	0.581	-0.033	0.583	1.0	**<2E-4**	**<2E-4**
16	RPS15A	0.578	0.394	-0.490	**<2E-4**	**<2E-4**	**<2E-4**
17	LY9	0.575	0.474	0.548	**<2E-4**	**<2E-4**	**<2E-4**
18	RPS12	0.567	0.292	-0.434	**0.0052**	**<2E-4**	**<2E-4**
19	SLC2A3	0.552	0.070	0.560	1.0	**<2E-4**	**<2E-4**
20	RPS19	0.549	0.374	-0.403	**<2E-4**	**<2E-4**	**<2E-4**

Table H.6: Genes with largest difference variance between blood and tumor naïve B cells. The columns Δ {Dispersion, Mean, Variance} show the changes in the corresponding statistic for each listed gene in tumor (396 cells) relative to blood (83 cells). N/A values in the Δ(Dispersion) and Dispersion p-value columns indicate that the gene had zero mean-expression in blood, and so dispersion is undefined. We performed one-sided permutation tests (last three columns) to calculate the significance of the change in mean, dispersion, and variance, and those p-values which are significant after Bonferroni correction ($p < 0.01$) are shown in boldface. Zero p-values are shown as <2E-4, which is the smallest possible non-zero p-value we could obtain based on 100k permutation trials after Bonferroni correction.

T Cells CD8 (GO Enrichment)

GO Biological Process Complete	Bgnd	Count	Expected	+/-	Fold Enrich	P-value
negative regulation of interleukin-2 production (GO:0032703)	24	3	0.03	+	>100	2.55E-02
response to bacterium (GO:0009617)	691	7	0.72	+	9.68	4.06E-02
negative regulation of transcription by RNA polymerase II (GO:0000122)	886	8	0.93	+	8.63	1.63E-02
negative regulation of RNA metabolic process (GO:0051253)	1371	9	1.43	+	6.27	4.13E-02
regulation of transcription by RNA polymerase II (GO:0006357)	2255	11	2.36	+	4.66	3.96E-02
regulation of nucleobase-containing compound metabolic process (GO:0019219)	4019	15	4.21	+	3.57	4.81E-03
regulation of RNA metabolic process (GO:0051252)	3768	14	3.94	+	3.55	1.88E-02
regulation of cellular macromolecule biosynthetic process (GO:2000112)	3894	14	4.08	+	3.44	2.84E-02
regulation of macromolecule biosynthetic process (GO:0010556)	4033	14	4.22	+	3.32	4.38E-02
negative regulation of cellular process (GO:0048523)	4768	15	4.99	+	3.01	4.78E-02
negative regulation of biological process (GO:0048519)	5355	16	5.6	+	2.86	2.91E-02
regulation of nitrogen compound metabolic process (GO:0051171)	5821	17	6.09	+	2.79	1.11E-02
regulation of primary metabolic process (GO:0080090)	6004	17	6.28	+	2.71	1.79E-02
regulation of macromolecule metabolic process (GO:0060255)	6140	17	6.43	+	2.65	2.53E-02
regulation of cellular metabolic process (GO:0031323)	6212	17	6.5	+	2.62	3.03E-02
Analysis Type:	PANTHER Overrepresentation Test (Released 20190711)					
Annotation Version and Release Date:	GO Ontology database Released 2020-02-21					

Table H.7: GO enrichment analysis of differentially variable genes between tumor and blood for CD8 T cells in the lung cancer dataset. We obtained the gene ontology (GO) terms significantly enriched in the top twenty genes ranked by increase in variance in tumor v. blood for CD8 T cells (shown in Table H.2). The analysis was performed using the web service available at: http://geneontology.org/. Significance is calculated using Fisher's exact test, and p-values are Bonferroni corrected. Bgnd: Background count

T Cells CD4 Memory Resting (GO Enrichment)

GO Biological Process Complete	Bgnd	Count	Expected	+/-	Fold Enrich	P-value
negative regulation of interleukin-2 production (GO:0032703)	24	3	0.03	+	>100	2.55E-02
chaperone cofactor-dependent protein refolding (GO:0051085)	30	3	0.03	+	95.56	4.74E-02
regulation of cellular response to heat (GO:1900034)	79	4	0.08	+	48.38	1.38E-02
response to unfolded protein (GO:0006986)	166	5	0.17	+	28.78	6.92E-03
response to topologically incorrect protein (GO:0035966)	188	5	0.2	+	25.41	1.26E-02
negative regulation of transcription by RNA polymerase II (GO:0000122)	886	8	0.93	+	8.63	1.63E-02
cellular response to stress (GO:0033554)	1744	11	1.83	+	6.03	3.05E-03
regulation of transcription by RNA polymerase II (GO:0006357)	2255	11	2.36	+	4.66	3.96E-02
positive regulation of macromolecule metabolic process (GO:0010604)	3388	13	3.55	+	3.67	4.34E-02
regulation of nucleobase-containing compound metabolic process (GO:0019219)	4019	15	4.21	+	3.57	4.81E-03
regulation of macromolecule biosynthetic process (GO:0010556)	4033	15	4.22	+	3.55	5.04E-03
regulation of cellular macromolecule biosynthetic process (GO:2000112)	3894	14	4.08	+	3.44	2.84E-02
regulation of biosynthetic process (GO:0009889)	4258	15	4.46	+	3.37	1.05E-02
negative regulation of cellular process (GO:0048523)	4768	15	4.99	+	3.01	4.78E-02
regulation of macromolecule metabolic process (GO:0060255)	6140	17	6.43	+	2.65	2.53E-02
Analysis Type:	PANTHER Overrepresentation Test (Released 20190711)					
Annotation Version and Release Date:	GO Ontology database Released 2020-02-21					

Table H.8: GO enrichment analysis of differentially variable genes between tumor and blood for CD4 memory resting T cells in the lung cancer dataset. We obtained the gene ontology (GO) terms significantly enriched in the top twenty genes ranked by increase in variance in tumor v. blood for CD4 memory resting T cells (shown in Table H.3). The analysis was performed using the web service available at: `http://geneontology.org/`. Significance is calculated using Fisher's exact test, and p-values are Bonferroni corrected. Bgnd: Background count

Table H.9: GO enrichment analysis of differentially variable genes between tumor and blood for CD4 naïve T cells in the lung cancer dataset. We obtained the gene ontology (GO) terms significantly enriched in the top twenty genes ranked by increase in variance in tumor v. blood for CD4 naïve T cells (shown in Table H.4). The analysis was performed using the web service available at: http://geneontology.org/. Significance is calculated using Fisher's exact test, and p-values are Bonferroni corrected. Bgnd: Background count

CD4 T Cells Naïve

GO biological process complete	Bgnd	Count	Expected	+/-	Fold Enrich	P-value
SRP-dependent cotranslational protein targeting to membrane (GO:0006614)	96	8	0.1	+	83.55	4.25E-10
cotranslational protein targeting to membrane (GO:0006613)	101	8	0.1	+	79.41	6.26E-10
nuclear-transcribed mRNA catabolic process, nonsense-mediated decay (GO:0000184)	120	9	0.12	+	75.19	2.01E-11
protein targeting to ER (GO:0045047)	110	8	0.11	+	72.92	1.2E-09
establishment of protein localization to endoplasmic reticulum (GO:0072599)	114	8	0.11	+	70.36	1.58E-09
viral transcription (GO:0019083)	115	8	0.11	+	69.75	1.68E-09
translational initiation (GO:0006413)	142	9	0.14	+	63.55	8.55E-11
viral gene expression (GO:0019080)	132	8	0.13	+	60.76	4.84E-09
protein localization to endoplasmic reticulum (GO:0070972)	138	8	0.14	+	58.12	6.8E-09
nuclear-transcribed mRNA catabolic process (GO:0000956)	194	9	0.19	+	46.51	1.27E-09
protein targeting to membrane (GO:0006612)	177	8	0.18	+	45.32	4.62E-08
mRNA catabolic process (GO:0006402)	213	9	0.21	+	42.36	2.87E-09
oxidative phosphorylation (GO:0006119)	109	4	0.11	+	36.79	0.0388
RNA catabolic process (GO:0006401)	247	9	0.25	+	36.53	1.04E-08
establishment of protein localization to membrane (GO:0090150)	289	8	0.29	+	27.75	0.00000203
nucleobase-containing compound catabolic process (GO:0034655)	372	9	0.37	+	24.26	0.000000366
translation (GO:0006412)	378	9	0.38	+	23.87	0.000000421
peptide biosynthetic process (GO:0043043)	403	9	0.4	+	22.39	0.000000734
protein targeting (GO:0006605)	371	8	0.37	+	21.62	0.000014
heterocycle catabolic process (GO:0046700)	429	9	0.43	+	21.03	0.00000126
cellular nitrogen compound catabolic process (GO:0044270)	430	9	0.43	+	20.98	0.00000129
aromatic compound catabolic process (GO:0019439)	444	9	0.44	+	20.32	0.0000017
organic cyclic compound catabolic process (GO:1901361)	478	9	0.48	+	18.88	0.00000323
establishment of protein localization to organelle (GO:0072594)	448	8	0.45	+	17.9	0.0000596
peptide metabolic process (GO:0006518)	519	9	0.52	+	17.39	0.00000659
amide biosynthetic process (GO:0043604)	526	9	0.52	+	17.15	0.0000074
protein localization to membrane (GO:0072657)	514	8	0.51	+	15.6	0.000171
mRNA metabolic process (GO:0016071)	690	9	0.69	+	13.08	0.0000767
protein localization to organelle (GO:0033365)	761	9	0.76	+	11.86	0.000178
cellular amide metabolic process (GO:0043603)	777	9	0.77	+	11.61	0.000212
viral process (GO:0016032)	784	9	0.78	+	11.51	0.000229
symbiotic process (GO:0044403)	876	9	0.87	+	10.3	0.00059
cellular macromolecule catabolic process (GO:0044265)	907	9	0.9	+	9.95	0.000793
macromolecule catabolic process (GO:0009057)	1048	9	1.05	+	8.61	0.00269
intracellular protein transport (GO:0006886)	997	8	0.99	+	8.04	0.0254
organonitrogen compound biosynthetic process (GO:1901566)	1382	10	1.38	+	7.25	0.00218
cellular nitrogen compound biosynthetic process (GO:0044271)	1638	10	1.63	+	6.12	0.0105
negative regulation of gene expression (GO:0010629)	1757	10	1.75	+	5.71	0.0199
cellular localization (GO:0051641)	2393	11	2.39	+	4.61	0.0382

Analysis Type: PANTHER Overrepresentation Test (Released 20200407)
Annotation Version and Release Date: GO Ontology database Released 2020-02-21

Table H.10: GO enrichment analysis of differentially variable genes between tumor and blood for memory B cells in the lung cancer dataset. We obtained the gene ontology (GO) terms significantly enriched in the top twenty genes ranked by increase in variance in tumor v. blood for memory B cells (shown in Table H.5). The analysis was performed using the web service available at: http://geneontology.org/. Significance is calculated using Fisher's exact test, and p-values are Bonferroni corrected. Bgnd: Background count

B Cells Memory

GO Biological Process Complete	Bgnd	Count	Expected	+/-	Fold Enrich	P-value
chaperone-mediated protein complex assembly (GO:0051131)	18	3	0.02	+	>100	1.00E-02
regulation of cellular response to heat (GO:1900034)	79	4	0.08	+	50.8	1.12E-02
leukocyte activation involved in immune response (GO:0002366)	615	7	0.61	+	11.42	1.30E-02
cell activation involved in immune response (GO:0002263)	619	7	0.62	+	11.35	1.35E-02
response to organic substance (GO:0010033)	3009	12	3	+	4	4.55E-02

Analysis Type: PANTHER Overrepresentation Test (Released 20190711)
Annotation Version and Release Date: GO Ontology database Released 2020-02-21

B Cells Naïve

GO Biological Process Complete	Bgnd	Count	Expected	+/-	Fold Enrich	P-value
response to cAMP (GO:0051591)	101	6	0.1	+	59.6	6.52E-06
cellular response to calcium ion (GO:0071277)	85	5	0.08	+	59.02	2.14E-04
SRP-dependent cotranslational protein targeting to membrane (GO:0006614)	96	5	0.1	+	52.26	3.83E-04
cotranslational protein targeting to membrane (GO:0006613)	101	5	0.1	+	49.67	4.88E-04
protein targeting to ER (GO:0045047)	110	5	0.11	+	45.61	7.35E-04
establishment of protein localization to endoplasmic reticulum (GO:0072599)	114	5	0.11	+	44.01	8.72E-04
viral transcription (GO:0019083)	115	5	0.11	+	43.62	9.10E-04
response to organophosphorus (GO:0046683)	142	6	0.14	+	42.4	4.63E-05
nuc.-transc. mRNA catab. proc., nonsense-med. decay (GO:0000184)	120	5	0.12	+	41.81	1.12E-03
response to calcium ion (GO:0051592)	151	6	0.15	+	39.87	6.61E-05
response to purine-containing compound (GO:0014074)	157	6	0.16	+	38.34	8.27E-05
viral gene expression (GO:0019080)	132	5	0.13	+	38.01	1.76E-03
protein localization to endoplasmic reticulum (GO:0070972)	138	5	0.14	+	36.35	2.18E-03
translational initiation (GO:0006413)	142	5	0.14	+	35.33	2.51E-03
protein targeting to membrane (GO:0006612)	177	5	0.18	+	28.34	7.23E-03
cellular response to metal ion (GO:0071248)	193	5	0.19	+	25.99	1.10E-02
nuclear-transcribed mRNA catabolic process (GO:0000956)	194	5	0.19	+	25.86	1.12E-02
mRNA catabolic process (GO:0006402)	213	5	0.21	+	23.55	1.76E-02
response to mechanical stimulus (GO:0009612)	218	5	0.22	+	23.01	1.97E-02
cellular response to inorganic substance (GO:0071241)	221	5	0.22	+	22.7	2.10E-02
RNA catabolic process (GO:0006401)	247	5	0.25	+	20.31	3.58E-02
response to metal ion (GO:0010038)	370	6	0.37	+	16.27	1.16E-02
cellular response to hormone stimulus (GO:0032870)	612	7	0.61	+	11.48	1.26E-02
response to bacterium (GO:0009617)	691	7	0.69	+	10.16	2.79E-02
response to organic cyclic compound (GO:0014070)	926	8	0.92	+	8.67	1.46E-02
response to organonitrogen compound (GO:0010243)	1006	8	1	+	7.98	2.70E-02
response to other organism (GO:0051707)	1322	10	1.32	+	7.59	1.43E-03
response to external biotic stimulus (GO:0043207)	1324	10	1.32	+	7.58	1.45E-03
response to biotic stimulus (GO:0009607)	1356	10	1.35	+	7.4	1.81E-03
response to nitrogen compound (GO:1901698)	1089	8	1.09	+	7.37	4.85E-02
interspecies interaction between organisms (GO:0044419)	1964	12	1.96	+	6.13	4.25E-04
cellular nitrogen compound biosynthetic process (GO:0044271)	1638	10	1.63	+	6.13	1.04E-02
RNA metabolic process (GO:0016070)	1679	10	1.67	+	5.98	1.30E-02
cellular macromolecule biosynthetic process (GO:0034645)	1694	10	1.69	+	5.92	1.41E-02
macromolecule biosynthetic process (GO:0009059)	1753	10	1.75	+	5.72	1.93E-02
negative regulation of gene expression (GO:0010629)	1757	10	1.75	+	5.71	1.97E-02
response to external stimulus (GO:0009605)	2443	12	2.43	+	4.93	4.76E-03
negative regulation of macromolecule metabolic process (GO:0010605)	2682	12	2.67	+	4.49	1.32E-02
negative regulation of metabolic process (GO:0009892)	2942	12	2.93	+	4.09	3.58E-02
response to stress (GO:0006950)	3572	13	3.56	+	3.65	3.63E-02
Analysis Type:	PANTHER Overrepresentation Test (Released 20190711)					
Annotation Version and Release Date:	GO Ontology database Released 2020-02-21					

Table H.11: GO enrichment analysis of differentially variable genes between tumor and blood for naïve B cells in the lung cancer dataset. We obtained the gene ontology (GO) terms significantly enriched in the top twenty genes ranked by increase in variance in tumor v. blood for naïve B cells (shown in Table H.6). The analysis was performed using the web service available at: http://geneontology.org/. Significance is calculated using Fisher's exact test, and p-values are Bonferroni corrected. Bgnd: Background count

Table H.12: Validation of top differentially variable genes between tumor and blood in CD8 T cells on a secondary dataset. We repeat the tests for difference in mean, variance, and dispersion index of 19 of the top 20 genes from Table H.2 for CD8 T cells in blood (1,250 cells) and lung cancer (2,123 cells) on a secondary dataset from Guo et al. (2018) (IGKC was not found in the dataset). Other cell types we analyzed did not have a close match in this dataset and were omitted from the analysis. The columns Δ {Dispersion, Mean, Variance} show the changes in the corresponding statistic for each listed gene in tumor relative to blood. We performed one-sided permutation tests (last three columns) to calculate the significance of the change in dispersion, mean, and variance, and those p-values which are significant after Bonferroni correction ($p < 0.01$) are shown in boldface. We see that 9 of the 19 genes are significant in differential variance in this dataset as well and TSC22D3 is also significantly overdispersed in tumor in this dataset. Zero p-values are shown as <2E-4, which is the smallest possible non-zero p-value we could obtain based on 100k permutation trials after Bonferroni correction.

T Cells CD8 (Validation)							
					Permutation Test p-value (Bonferroni-corrected)		
Rank	Gene	Δ(Variance)	Δ(Dispersion)	Δ(Mean)	Dispersion	Mean	Variance
1	RGS1	8.099	-0.782	4.155	1.0	**<2E-4**	**<2E-4**
2	DUSP4	4.170	0.522	1.238	1.0	**<2E-4**	**<2E-4**
3	FOSB	4.143	0.126	1.182	1.0	**<2E-4**	**<2E-4**
4	RGCC	3.299	-0.697	0.941	1.0	**<2E-4**	**<2E-4**
5	NR4A2	2.788	0.496	0.631	0.1368	**<2E-4**	**<2E-4**
6	TSC22D3	2.517	0.585	-0.766	**<2E-4**	**<2E-4**	**<2E-4**
7	CREM	2.475	-0.437	0.765	1.0	**<2E-4**	**<2E-4**
8	TNFAIP3	1.747	-1.071	1.789	1.0	**<2E-4**	**<2E-4**
9	JUND	0.785	-0.334	0.335	1.0	**<2E-4**	**<2E-4**
10	ZNF331	0.433	-0.115	0.127	1.0	0.687	0.8736
11	DNAJB1	0.397	0.077	0.058	1.0	1.0	0.6108
12	JUNB	0.350	-0.666	0.805	1.0	**<2E-4**	1.0
13	YPEL5	0.249	-0.586	0.504	1.0	**<2E-4**	1.0
14	FOSL2	0.007	-0.161	0.018	1.0	1.0	1.0
15	FTH1	-0.051	-0.011	0.086	1.0	1.0	1.0
16	ZFP36	-0.286	-0.287	0.687	1.0	**<2E-4**	1.0
17	HSP90AA1	-0.435	-0.191	0.395	1.0	**<2E-4**	1.0
18	RPS26	-0.452	-0.330	-0.045	1.0	1.0	1.0
19	CCL4	-2.924	-1.309	2.323	1.0	**<2E-4**	1.0