Algorithms for Understanding and Fighting Infectious Disease

by

Brian Lance Hie

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

Author .....
Department of Electrical Engineering and Computer Science
February 26, 2021

Certified by .....
Bonnie A. Berger
Simons Professor of Mathematics
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Algorithms for Understanding and Fighting Infectious Disease

by

Brian Lance Hie

Submitted to the Department of Electrical Engineering and Computer Science
on February 26, 2021, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Infectious disease is a persistent and substantial threat to human health, with consequences that include widespread mortality, suffering, and economic disruption. This thesis presents several algorithmic advances that, when coupled with biotechnologies for data collection and perturbation, are aimed at understanding infectious disease and using this knowledge to fight it. First, this thesis develops geometric algorithms that enable a panoramic understanding of the systems biology of the human immune system and of infectious pathogens at single-cell resolution. Next, this thesis will show how state-of-the-art Bayesian machine learning can explore complex biological spaces to search for new therapies that fight infectious disease. Finally, this thesis develops neural language models that can predict how pathogens mutate to evade human immunity, potentially enabling more broadly effective vaccines and therapies. Taken together, this thesis outlines a highly interdisciplinary, algorithmic approach to infectious disease research, with broader implications for computation and biology more generally.

Thesis Supervisor: Bonnie A. Berger
Title: Simons Professor of Mathematics

Acknowledgments

The overall story of modern infectious disease research is that of humanity overcoming existential challenges through resilience, cleverness, and, perhaps most importantly, scientific collaboration. In a similar if microcosmic way, doing a Ph.D. not only requires hard work and aptitude but also the support of many. I am foremost indebted to my advisors, Bonnie Berger and Bryan Bryson, for their constant support, technical guidance, and mentorship. I am especially thankful for Bonnie's constant encouragement to have my research be rooted in long-term vision and for Bryan's inspiration to use my work in service of those who may be overlooked by the mainstream.

I am also thankful for my undergraduate advisors, Irene Kaplow and Hunter Fraser, for teaching me when I knew almost nothing about computation or biology, for setting my research on a strong foundation, and for encouraging me to apply to Ph.D. programs when I was unsure about my career path. I am sincerely grateful to Patrice Macaluso for her kindness and for her unwavering support administratively and beyond. I also thank some of the other mentors I had for brief portions of my Ph.D.: Hyunghoon Cho, Natasha Patel-Murray, and Ernest Fraenkel who advised me during rotations; Yong Li, Hongxu Ma, Bin Ni, and Anu Thugabere who supervised my internships at Illumina and X; and Guy Bresler and Greg Wornell who ran 6.438 when I had the fortune of serving as a teaching assistant. I am grateful to Alicia Duarte, Janet Fischer, Leslie Kolodziejski, and Kathy McCoy in the EECS graduate office for their helpful and kind assistance throughout my Ph.D. I also thank Timothy Lu for his helpful guidance and advice as a member of my thesis committee.

Part of the privilege of doing this thesis at MIT is the astoundingly high quality of peers and collaborators. My coauthors on the papers described in this thesis helped enrich and complete the stories detailed here; my gratitude in particular extends to Hyunghoon Cho, Benjamin DeMeo, Ashwin Narayan, Sarah Nyquist, Josh Peters, Rohit Singh, and Ellen Zhong. I also thank the broader coterie of fellow graduate students and the postdocs in the Berger and Bryson labs for making my Ph.D. so much richer; this group includes (in addition to those mentioned above) Tristan Bepler,

Bariş Ekim, Younhun Kim, Andrew Morin, Sumaiya Nazeen, Ibrahim Numanagic, Perry Palmedo, Ariya Shajii, Max Sherman, Sam Sledzieski, and Alex Wu in the Berger lab and Cal Gunnarsson, Chris Itoh, Owen Leddy, Bianca Lepe, Akeem Ngomu, and Sydney Solomon in the Bryson lab. I am also thankful to call many of these academic peers my friends as well.

I am grateful for the support of friends throughout my Ph.D. I am lucky to have made new friends in Boston, and I would like to thank in particular Eddie Irvine and Lauren Yeager for their personal support. I am deeply thankful for Nate Adams, Kyle Becker, David Cheng, Curtis Cook, Eric Hambley, Andrew Johnson, James Lee, Mike and Lexi Miltenberger, Sam Park, Eduardo and Shannon Pujol, John Sullivan, Randy Song, Wendy Wei, and others at Hope Fellowship Church for their love and care. I would like to thank John Lee, Betty Lemma, Wenhao Liao, Taylor Madigan, and Samuel Tovmasian for their friendship since high school; An Nguyen, Michael Tran, and the other members of 970 Meridian, 414 3rd Ave, and 2745 35th Ave for their friendship and camaraderie; and the Oxford Clique and the Alumni of Room 319, i.e., Robert Chun, Paul Carroll, Ryan Hoaglund, Noah Johnston, Soo Ji Lee, Michael Limandibrata, John Newcomb, Loren Pilorin, Thomas Plank, Ishita Prasad, Andrew Sierra, Paul Shields, Kamaria Taylor, and Alonzo Virata for their friendship since college.

Finally, my deepest thanks goes to my family, to whom this thesis is dedicated.

San Diego, California
February 2021

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Ils se croyaient libres et personne ne sera jamais libre
tant qu'il y aura des fléaux.
(They thought they were free and no one will ever be free
as long as there are plagues.)

—Albert Camus, *La Peste* (1947)

1.1 Overview: Old scourge, new hope

Humans have been dealing with infectious disease for our entire biological history. Scientific breakthroughs in understanding the biology of infectious disease and the human immune response has led to antibiotics and vaccines, which have enabled better control or even the eradication of once formidable diseases like plague, polio, and smallpox [HSS+19]. Still, infectious disease remains a persistent and global threat. The leading cause of infectious disease in modern times has been tuberculosis, caused by a bacterial infection that leads to ~1 million deaths a year [FCP19]. Viral pandemics including flu, AIDS, and COVID-19 have also collectively claimed many millions of lives [KWW18, DOPB15].

Part of the reason why infectious disease remains such a difficult challenge is biological complexity. For example, much is still unknown about how infection with

the bacteria *Mycobacterium tuberculosis* (Mtb), the causative agent of tuberculosis disease, progresses from latent disease (in ~90% of infected individuals) to active and potentially fatal disease [FCP19]. Many pathogens, like influenza virus and HIV, are constantly evolving to evade immune responses like those induced by vaccines. And even if all currently known infectious pathogens were eliminated, the emergence of novel pathogens with unknown biology is always a possibility.

Fortunately, the twenty-first century has seen a rapid growth in biological knowledge and computational advances that can expand this knowledge, driven by two major recent trends. The first is an exponential increase in the generation of biological data using new biotechnologies that enable high-throughput, massively-parallel interrogation of biological systems. The second is the increasing sophistication of computer algorithms that learn patterns from these large biological datasets and translate them into new insights and predictions. A promise of these two technological revolutions is that, together, they might lead to new interventions that reduce human suffering due to infectious disease.

This thesis makes a number of novel algorithmic contributions to infectious disease research, as well as to computation and biology more broadly. There are two main motifs that recur throughout this thesis, the first of which is:

Question 1: *How do we simplify immense biological complexity?*

Understanding the biology of infectious disease involves understanding the host, the pathogen, and the host-pathogen interactions. Biological systems are combinatorially complex at multiple levels, from genetic sequence to genetic interactions to multicellular organization. Biological complexity quickly overwhelms the capacity of an individual scientist to understand the full picture all at once.

In this thesis, we will see how algorithms can help researchers distill very complex information into a simpler set of abstractions. This often takes the form of unsupervised learning algorithms that find shared, common patterns across large biological datasets. Algorithms can also highlight significant relationships among a large number of variables, allowing researchers to reason about correlation or causality in a system.

Furthermore, computers can analyze terabytes of otherwise unintuitive data, like large corpuses of biological sequences, and thereby complement the human biologist's ability to formulate creative, biological hypotheses using a more limited amount of data. This leads to the second, high-level theme in this thesis:

Question 2: *How can algorithms discover new biology?*

New biology can come in the form of new knowledge about a fundamental biological mechanism or a new way to prevent or treat disease. The gold standard in computational biology is when an algorithm makes a prediction about some biological phenomenon that is then validated by (reproducible) experimentation. For example, an algorithm can predict that a certain mutation gives rise to a given physical trait, which can be validated by actually making the mutation in the laboratory. Or, an algorithm can identify a potential drug for a given disease, which is then advanced into the clinical trial process.

The goal of this thesis, ultimately, is to design computational models (i.e., models that exist *in silico*) that can make predictions that agree with or complement biological models inside a laboratory test environment (i.e., *in vitro*) or inside a living organism (i.e., *in vivo*). A fundamental reason why *in silico* models are preferable to *in vitro* or *in vivo* models is that it is much easier to evaluate a computational model (e.g., by running a program on a compute cluster) than it is to perform the experiment in the laboratory (e.g., experimenting on laboratory mice or human tissue). An algorithm that goes beyond a human's capacity for processing biological information, yields creative biological insight, and comes with a lower resource cost than traditional wetlab biology would be a very useful tool for infectious disease research and biology more broadly.

The challenge for the computer scientist, therefore, is to make the computational model as good at capturing real biology as the best biological models. This means that different types of computation will be better suited to different problems, e.g., depending on resource constraints or data types. The complexity of biological problems requires a broad computational toolkit that spans multiple subfields within computer

science. This thesis likewise reflects the breadth of modeling techniques required to make advances against complex, real-world problems.

1.2 Thesis organization

This thesis begins with some basic background, then describes our algorithmic contributions to understanding and fighting disease, and then ends with a perspective on future directions. The computational methods presented throughout the thesis cover a breadth of topics—from geometric algorithms to quadratic programming to Bayesian machine learning to neural language models—and are meant to arm a computational reader with the diverse technical skills required to do biological research, especially in the complex settings associated with infectious disease. Each chapter describing algorithmic results can be read somewhat independently of the rest of thesis (though Chapters 6 and 7 are more closely linked). However, all of the ideas do form a coherent narrative from the beginning to the end, and reading the thesis this way should hopefully facilitate better appreciation for some of the higher-level ideas described in this introduction.

The thesis begins with a chapter providing general background on fundamentals in both biology and computation that are relevant to this thesis. Biological basics can be helpful to those coming to this thesis from a more computational background and include overviews of topics in biochemistry, molecular biology, immunology, microbiology, and pharmacology. For those coming to this thesis from a more biological background, we also provide high-level overviews of algorithms, machine learning, and computational geometry.

We then begin an extended discussion of how to better *understand* infectious disease using modern high-throughput biology. In Chapter 3, we discuss a fundamental problem in these analyses: how do you compare complex patterns across different experiments and studies? Our proposed solution uses pattern matching techniques inspired by algorithms from computer vision for panoramic stitching, which we use to integrate multiple biological datasets. The bulk of this chapter is based on the paper:

- Brian Hie, Bryan Bryson, and Bonnie Berger. "Efficient integration of heterogeneous single-cell transcriptomes using Scanorama." *Nature Biotechnology,* 37:6 (2019) [HBB19].

Integrating and combining multiple large-scale biological datasets into a single panorama leads to problems with scalability, so in Chapter 4, we discuss an algorithm for accelerating data analysis. In particular, we propose a method for downsampling or "sketching" a dataset such that more redundant information is removed while preserving biological diversity. We then show how this approach can accelerate otherwise time-consuming biological analyses. This chapter is based on the paper:

- Brian Hie, Hyunghoon Cho, Benjamin DeMeo, Bryan Bryson, and Bonnie Berger. "Geometric sketching compactly summarizes the single-cell transcriptomic landscape." *Cell Systems,* 8:6 (2019) [HCD$^+$19].

We then move from a discussion on comparing patterns across experiments to instead comparing patterns across data modalities in Chapter 5. This work addresses an increasingly common biological experiment in which multiple data types are measured for the same biological sample within the same experiment, e.g., simultaneously measuring the gene expression, protein expression, and spatial localization of a single cell. We develop a general approach for synthesizing information across biological data types and modalities, which is based on the paper:

- Rohit Singh, Brian Hie, Ashwin Narayan, and Bonnie Berger. "Metric learning enables synthesis of heterogeneous single-cell modalities." *bioRxiv* (2020) [SHNB20].

To make progress against infectious disease, we ultimately need to translate knowledge about the disease into tangible ways to control or eradicate the disease. This thesis therefore pivots from a focus on understanding infectious disease to an extended discussion on *fighting* infectious disease, i.e., algorithms that propose new interventions or aid therapeutic design. In Chapters 6 and 7, we lay out different algorithmic approaches for drug discovery. Beginning in Chapter 6, we describe the

drug-target interaction prediction problem and develop an initial approach based on a neural network, which is described in the paper:

- Brian Hie, Hyunghoon Cho, and Bonnie Berger. "Realizing private and practical pharmacological collaboration." *Science,* 362:6417 (2018) [HCB18].

We show that our model achieves state-of-the-art performance while enabling scalability to millions of training examples, but also highlight some shortcomings especially when the algorithm is used to make predictions for experimental validation. We therefore revisit the biomolecular interaction prediction problem in Chapter 7 but with a different approach that enables a machine learning model to quantify the *uncertainty* of its predictions. Using this new approach, we discover a number of novel, potent biomolecular interactions that include compounds that inhibited the growth of Mtb, suggesting molecular structures that could be useful for tuberculosis drug development. These results are based on the paper:

- Brian Hie, Bryan Bryson, and Bonnie Berger. "Leveraging uncertainty in machine learning accelerates biological discovery and design." *Cell Systems,* 11:5 (2020) [HBB20].

Even after a therapy has been developed, however, a pathogen can acquire resistance to the therapy through evolution. In Chapter 8, we therefore discuss how algorithms might also mitigate this threat as well. We develop a machine learning algorithm for predicting viral resistance to immune selection, or viral escape. Our approach could be used to predict resistance before it occurs and therefore design better therapies and vaccines. These results are based on the papers:

- Brian Hie, Ellen Zhong, Bryan Bryson, and Bonnie Berger. "Learning mutational semantics." *Neural Information Processing Systems* (2020) [HZBB20].

- Brian Hie, Ellen Zhong, Bonnie Berger, and Bryan Bryson. "Learning the language of viral evolution and escape." *Science,* 371:6526 (2021) [HZBB21].

In the final chapter, we reflect on the overall themes throughout the thesis and lay out ways these themes can drive further scientific discovery. We discuss ideas that are

both near-term extensions and remixes of the work done in this thesis as well as ideas that are longer-term or more general research directions. Throughout the entirety of the thesis, we hope to convey that computational biology, especially in the context of infectious disease research, is a highly interdisciplinary and exciting field with much room for novel contributions.

1.2.1 Computational contributions

This thesis makes a number of contributions to computational biology, algorithms, and machine learning methods, which are summarized below:

- In Chapter 3, we contribute an efficient algorithm for heterogeneous data integration, implemented with randomized singular value decomposition and approximate nearest neighbors search, that has empirical performance (in terms of both speed and accuracy) within the top tier of similar methods according to a comprehensive, independent benchmark (Section 3.5.6).

- In Chapter 4, we develop a novel algorithm for diversity-preserving random subsampling with near-linear scalability and that can efficiently process datasets with millions of examples with high dimensionality.

- In Chapter 5, we contribute an elegant conceptual model of multimodal data analysis based on a quadratic programming approach to metric learning.

- In Chapter 6, we advance the state-of-the-art in compound-target interaction prediction and demonstrate scalability that also enables cryptographically secure neural network training.

- In Chapter 7, we demonstrate the value of uncertainty in biological discovery and are the first to apply Gaussian process regression to the compound-target interaction prediction problem, which we validate using multiple rounds of laboratory experiments and discover novel, high-affinity biomolecular interactions.

- In Chapter 8, we introduce the problem of finding the single-token mutation to a sequence, constrained by a grammar, that induces the highest semantic change, which we call a "constrained semantic change search"; we implement this problem using a novel interpretation of language models that combines both "grammaticality" and "semantic change"; and we use this problem formulation to achieve state-of-the-art prediction of viral escape mutations.

1.3 Chapter organization

Each of the main chapters, i.e., Chapters 3 through 8, is organized similarly. The chapter begins with a high-level overview, followed by a glossary that is meant to highlight some of the most important concepts of that chapter and provide useful information and definitions. The glossary is followed by chapter-specific preliminary information on the greater context of the work described in the chapter (both biological and computational) and also includes important technical content, e.g., reviews on key subroutines or concepts leveraged by the algorithm. The glossary and the preliminaries section are meant to facilitate better appreciation of the contributions in the chapter but do stand apart somewhat from the main chapter narrative.

Then, each chapter presents the main algorithm, motivates the specific algorithmic approach, discusses some of the theory and modeling assumptions behind the algorithm, and then provides empirical benchmarking of the algorithm on simulated and real data. Each chapter concludes with a special application note, meant to highlight how the algorithm described in the chapter can be used to discover biological insights in an infectious disease context. These application notes are chosen to reflect a wide array of topics within the study of infectious disease and host-pathogen interactions. They also help root each chapter, which can often involve a lot of theoretical description, more firmly in the ultimate goal of this thesis, which is to be practically useful to infectious disease biology.

1.4 Who is this thesis for?

While I hope this thesis is useful for all readers, when writing this thesis I particularly had in mind an interested scientist very early in their career, like a first-year graduate student, who is deciding on their graduate research topic. Or, perhaps, a later-stage scientist wishing to pivot to a new research direction. This thesis is meant to be a helpful primer into the current state of the field and will hopefully lead its readers into academically fertile areas. I do also hope this thesis can inspire others, particularly those from computational backgrounds, to see that computation can be directed at improving human lives in a fairly direct way, and that computational biology has matured to a level where algorithms are beginning to regularly drive substantive biological discovery.

Chapter 2

Background

Let's start at the beginning,
A very good place to start.

—Julie Andrews, "Do-Re-Mi" (1965)

The biology of infectious disease is intimidatingly complex, so any computational work aimed at understanding and fighting infection will necessarily sit at the intersection of a host of fields, subfields, disciplines, and philosophies. This thesis is no exception and its contents move across many areas within biology, especially immunology and microbiology, as well as within computation, especially machine learning and geometric algorithms.

The goal of this background chapter is to provide very brief essential knowledge within the fields fundamental to this thesis to aid readers coming from different areas of expertise. Each chapter also begins with a section of preliminaries specific to that chapter. References in each section also point the reader to helpful books and review articles that offer deeper surveys of the discussed topics. Readers should feel free to read only the sections they deem relevant, or to skip this chapter entirely.

2.1 Biology basics

While a full picture of biology is not required to understand this thesis, biological knowledge should help a reader better appreciate some of the thesis's results, particularly in its practical applications. A high-level survey of fundamental biological knowledge relevant to this thesis is provided here, which may be especially useful for those coming to this thesis from a computational background.

2.1.1 Biochemistry

Living organisms are sustained by and influence their environments through chemical processes. The most important element in biochemistry is carbon, which has the ability to form strong covalent bonds with many atoms including itself, allowing carbon to form a chemical "backbone" for many molecules. Organic chemistry is the subfield of chemistry that deals with these carbon-based molecules [RUC+10].

Organic compounds with a low molecular weight are referred to as *small molecules*. Small molecules are important in biomedicine because they are often used as drugs that interfere with normal or disordered biological processes. Carbon can also form very large molecules, some of which achieve highly complex biological functions:

- *Nucleic acids* are large macromolecules composed of nucleotides that encode the information of all known life. *Deoxyribonucleic acid (DNA)* is composed of four nucleotide bases: adenine, cytosine, guanine, and thymine, commonly known as "`A`," "`C`," "`G`," and "`T`," respectively. *Ribonucleic acid (RNA)* replaces thymine with the nucleotide base uracil.

- *Amino acids* are biomolecules that can be combined into a chain called a *polypeptide*; a large polypeptide is called a *protein*. Proteins accomplish a vast diversity of functions and are critical to more complex forms of life.

- *Lipids* are biomolecules that do not easily dissolve in water and are therefore used as the main component of cell and viral membranes; they are also used for energy storage in fat.

- *Carbohydrates* are biomolecules that play an important role in energy storage and metabolism. The carbohydrate sugars ribose and deoxyribose form the molecular backbones of RNA and DNA, respectively.

Biochemistry studies how these molecules are generated and interact in the context of living organisms [RUC+10]. Biomolecules *bind* when two or more molecules are more energetically favorable together than apart. Binding interactions are critical in biology; for example, a small molecule drug will bind a protein target to inhibit that protein's function, or a protein called a transcription factor will bind DNA to initiate gene expression. We discuss biomolecular binding interactions in the context of transcription factor binding in Chapter 5, in the context of drug discovery in Chapters 6 and 7, and in the context of antibody neutralization in Chapter 8.

2.1.2 Molecular biology and the central dogma

Molecular biology is based on a *central dogma* that describes the information flow observed in all living organisms [RUC+10]. In nearly all life, DNA molecules form a central repository for all the information required for that organism to maintain life and to reproduce [WC53]. Subsequences of a DNA molecule called *genes* encode the information required to make a protein. The full DNA sequence within an organism is called the *genome*.

Information from DNA then flows through an intermediate RNA molecule. A gene is "transcribed" by copying the subsequence of the DNA molecule into a single messenger RNA (mRNA) molecule [JM61]. In *eukaryotic* cells, mRNA molecules must pass through a membrane separating the nucleus, where DNA is stored, from the rest of the cell. When passing through the nuclear membrane, subsequences of the mRNA molecules can be removed in a process called splicing. The totality of mRNA molecules inside a biological sample is called the *transcriptome*. Transcriptomics (i.e., the study of transcriptomes) makes up the bulk of our discussion in Chapters 3, 4, and 5.

These mRNA molecules are then "translated" into protein. A part of a cell

called a ribosome will sequentially read an mRNA sequence in order to biochemically synthesize a sequence of amino acid residues that eventually becomes a protein. A key biochemical property of proteins is that the amino acid residue sequence will fold into a three-dimensional structure, enabling complex biological function. The totality of protein inside a biological sample is called the *proteome*.

The information encoded by nucleic acids is referred to as a *genotype*, whereas the information corresponding to the structure and function of proteins is referred to as a *phenotype*. In its most general form, the central dogma of molecular biology states that information can flow from nucleic acid to protein (i.e., from genotype to phenotype), but not back from protein to nucleic acid. Typically, information also flows only from DNA to RNA, but in rarer instances information encoded as RNA can be re-encoded as DNA through a process called *reverse transcription*, which is a strategy used by some viruses like HIV to encode their genetic material into a host genome. The information link between genotype and phenotype dictated by the central dogma is critical to the analytic assumptions in much of this thesis, described further in Section 3.2.1.

In some settings, *gene expression* refers to the entire process in which the information in a gene goes from DNA to RNA to protein. In this thesis, we use gene expression to refer to the DNA to RNA *transcription* step and we use *protein expression* to refer to the RNA to protein *translation* step.

2.1.3 Microbes and pathogens

Microbes are small biological agents that are difficult to see with the naked eye. When they infect a larger host and cause disease, microbes become pathogenic. The three most important microbiological pathogens, which we focus on in this thesis, are viruses, bacteria, and protozoa.

Viruses

Viruses are efficient replication machines. Their primary job is to enter a host cell, hijack the cell's machinery for building proteins and other macromolecules, and create new copies of the virus that exit the cell and go on to infect other cells. Importantly, viruses do not have a cellular structure and cannot reproduce without the help of a host organism. Viruses do have RNA or DNA genomes that are packaged by a protein coat called a capsid. The viruses that are particularly relevant to this thesis, particularly to Chapter 8 are "enveloped viruses": *influenza virus*, *HIV*, and *coronavirus*. These viruses are surrounded by a lipid membrane envelope (using material taken from the host cell membrane) with surface proteins embedded in the envelope that are key to attaching to new host cells [EK01].

A scientific debate concerns whether or not viruses can be classified as living organisms, since they share many characteristics with organisms accepted as living (e.g., they have genes and evolve through natural selection) but lack other characteristics (e.g., they cannot replicate outside a host, they lack a metabolism, and they do not have a cellular structure). This thesis holds to the commonly accepted notion that viruses are not fully "alive," but we still use terms associated with life for linguistic convenience (e.g., a drug can "kill" a virus or affect viral "viability") [RUC+10].

Bacteria

Bacteria are single-celled organisms that are more complex than viruses. While viruses are typically parasitic, many bacteria have a neutral (commensal) or positive (symbiotic) benefit on their hosts. Nearly all bacteria can reproduce outside a host via cell division. Bacteria have DNA genomes and are *prokaryotic* in that their genomes are not separated from the rest of the cell by a nuclear membrane; bacterial genes are therefore translated into mRNA molecules that are then immediately translated into protein. Bacteria surround their cell membranes with a cell wall that makes it more difficult for foreign substances, including antibiotic drugs, to enter the cell [RUC+10].

Pathogenic bacteria pose a substantial threat to human health. One bacterial

disease we focus on in this thesis in Chapter 7 is tuberculosis, caused by *Mycobacterium tuberculosis* (Mtb), which has been the leading cause of infectious disease death in modern times [FCP19]. Mtb infects cells in the lungs called alveolar macrophages, where they establish long-lasting infection. While some hosts can control Mtb infection as part of latent disease, uncontrolled infection leads to active tuberculosis disease, respiratory failure, and death.

Protozoa

Protozoa are single-celled eukaryotic microbes that can, like bacteria, become pathogenic. Protozoa reproduce via cell division and establish infection in different human organs and physiological systems. Eukaryotes differ from bacteria in that their DNA genomes are encased by a nuclear membrane within the cell itself (most multicellular life, including plants and animals, consists of eukaryotic cells) [RUC+10].

In Chapter 3, we focus on two important protozoal pathogens. The first is *Toxoplasma gondii*, which infects most birds and mammals and causes the disease toxoplasmosis, which can be deadly for immunocompromised humans [XTR+20]. *Toxoplasma* has also been shown to have neurological effects, causing reduced fear in laboratory mice and potentially making them more vulnerable to predation; it therefore may make sense that *Toxoplasma* is often found in cat feces [TM17].

The second is *Plasmodium*, a genus of protozoa that causes the disease malaria, which continues to be extremely deadly worldwide [HRA+19]. *Plasmodium* is typically transmitted between humans via mosquitoes, where it infects red blood cells and cells in the liver called hepatocytes as part of its reproductive lifecycle. In humans, the main species of *Plasmodium* that cause malaria are *Plasmodium falciparum* and *Plasmodium vivax*.

2.1.4 The immune system

Fortunately, the human body has an intricate and mostly effective defense against potential microbial invaders. This task is nontrivial, as the lungs and the digestive

system must process gallons of unsterile air, water, and food on a daily basis. The *immune system* is composed of a variety of biochemical processes and types of cells that work in concert to prevent infection.

Innate versus adaptive immunity

The first important distinction in immunology is between *innate* and *adaptive* immunity [MW16]. Innate immunity is a nonspecific defense directed against foreign invasion in general. Anatomical barriers that prevent pathogens from entering the body, like skin, are considered a part of innate immunity. Innate immunity also consists of different types of cells, of which the most important for the purposes of this thesis are cells called *phagocytes*. Phagocytes engulf foreign pathogens by extending their cell membranes to envelop the pathogen, forming a compartment in the cell called a phagosome; enzymes, acids, and other toxic material is then transported into the phagosome to kill and digest the pathogen. An important type of phagocyte is the *macrophage*, a cell type that is highly efficient at phagocytosis and is present in many major organs, including the blood, lungs, and brain. The application note in Chapter 4 discusses how we might better understand macrophage function, much of which is still unknown.

Adaptive immunity builds up a specific immune response against a particular antigen [MW16]. The first step in adaptive immunity is *antigen presentation*, in which cells throughout the body present potentially foreign material to adaptive immune cells. Foreign material could either be from inside a cell (e.g., following infection) or from outside a cell (e.g., following phagocytosis). Most of this material is displayed on the surface of the cell by a complex of proteins called the *major histocompatibility complex (MHC)*. Then, adaptive immune cells called *lymphocytes* learn to recognize foreign material displayed as antigens; an important job of these lymphocytes is to distinguish foreign material from self, where imperfect classification could lead to autoimmunity. Once a foreign antigen has been identified, lymphocytes will build up specific defenses against that antigen and a memory lymphocyte will store information about that antigen for a prolonged period of time (some memory lasts for a lifetime).

Different components of adaptive immunity, described below, are interrogated at multiple points in this thesis, including in Chapters 5 and 8.

Humoral versus cell-mediated immunity

The second important distinction in immunology is between *humoral* and *innate* adaptive immunity [MW16]. Humoral immunity builds up a biomolecular response to pathogens. The most important component of both humoral immunity is the generation of antibodies, which are proteins that have a highly variable segment that is meant to bind diverse antigens. Antibodies are produced by lymphocytes known as *B cells*. Each B cell produces a unique antibody sequence; during infection, B cells that have stronger binding affinity for a foreign antigen are selected for multiplication by the immune system. Some of these B cells are longer lasting memory B cells that can persist for years after an infection event. The innate immune system also assists with humoral immunity by the production of proteins that can enhance antibody binding; this innate, humoral system is called the complement system. Antibody-based immunity is an important part of our discussion of viral escape in Chapter 8.

Cell-mediated immunity is driven by lymphocytes called *T cells* [MW16]. A type of T cell called the killer T cell will recognize foreign antigens displayed by infected cells and then kill those infected cells, thereby removing a place for a pathogen to replicate. Other types of T cells called helper T cells and regulatory T cells do not directly kill infected cells but coordinate different immunological mechanisms, including the activation of B cells. Memory T cells, like their B-cell analogs, also persist for a long period of time after an initial pathogen exposure. We go into more detail into how T cells sense foreign antigens, a process that is still not completely understood, in the application note of Chapter 5.

2.1.5 Cell types

Throughout our discussion on the immune system, we have described different *cell types* like macrophages, T cells, and B cells. Different cell types accomplishing varied

functions are present throughout the human body. While some cell type divisions are clear, e.g., the difference between a T cell and a macrophage, the general definition of a cell type is still a matter of scientific debate [CRE+17], which readers should keep in mind as they encounter the term "cell type" throughout this thesis, particularly in our exploration of algorithms for single-cell biology in Chapters 3, 4, and 5.

A *developmental* definition of cell type is based on an understanding of how cells mature, where different developmental processes and chemical stimuli lead to different "types" or "lineages" of cells. This definition is complicated by our current lack of insight into the full complexity of cellular development. Mapping these cell lineages and understanding how they are produced is a major open area of biological research.

A *functional* understanding of cell type says that cells are defined by the set of functions they accomplish; for example, red blood cells transport hemoglobin while neurons conduct electrical signals. This definition is more ambiguous since it also requires a formal definition of "function" and is complicated by cells that are traditionally thought to be of the same type but that, on closer inspection, have large amounts of functional heterogeneity; for example, macrophages can have different functions beyond phagocytosis depending on tissue and disease context.

A *descriptive* definition of cell type is similar to the functional definition but with less definitional ambiguity; here, cell types are defined based on the measurement of a set of features, and cells that quantitatively share similar features are assigned to the same cell type. Traditional immunology relies on a descriptive understanding of cell type by measuring the expression of different proteins on the surface of immune cells in order to distinguish, e.g., a $CD4^+$ helper T cell from a $CD8^+$ killer T cell. More modern, high-throughput techniques define cell types similarly, e.g., based on single-cell gene expression measurements. A shortcoming of this understanding of cell type is that the set of measured properties may be incomplete, so two cells may appear to be similar but might profoundly disagree in modalities that are not measured.

All three approaches to cell type—developmental, functional, and descriptive—are useful in understanding what a cell type is. There is probably not a single neat definition of cell type given the complexity of biological systems, but some definitional

clarity may emerge as our understanding of cell biology matures, especially using modern techniques for profiling millions of cells within a biological system.

2.1.6 Pharmacology

Antivirals and antibiotics

An antiviral is defined broadly as anything that slows or prevents viral infection, replication, or transmission at a biochemical level. Antibiotics are defined similarly but for living pathogenic microorganisms like bacteria and protozoa [JCP18]. These drugs can take the form of small molecules (e.g., that bind to and inhibit a virus's replication machinery) or larger biomolecules like proteins (e.g., an artificial antibody that binds to and neutralizes a virus's surface protein). A drug *cocktail* refers to a mixture of distinct drugs used as a combination therapy [BFW$^+$20]. Examples of potential new small molecule antibiotics appear in Chapter 7.

Vaccines

Vaccines exploit the long-term memory of adaptive immunity by exposing the immune system to antigens from a pathogen [HSS$^+$19]. Ideally, adaptive immunity (e.g., antigen presentation followed by antibody generation and T-cell-mediated immunity) should then be triggered against the vaccine antigen(s), including the generation of longer term memory. In theory, this memory should enable the immune system to respond quicker to future exposure to that antigen and therefore prevent or mitigate future disease. The oldest type of vaccine simply involves inoculating a person with a low dose of the pathogen or a weakened form of the pathogen, but this has the danger of actually causing disease. Modern approaches to vaccination expose the immune system to a whole inactivated virus, just the viral protein antigen, or to mRNA encoding a viral antigen (where the body then manufactures the antigen from the mRNA); these vaccines also typically include a substance called an adjuvant aimed at inducing a stronger immune response upon vaccination. We focus on the relationship between this immunity and viral evolution in Chapter 8.

2.2 Computation basics

A substantial amount of computational preliminaries are deferred to the individual chapters, which may be of more interest to those from computational backgrounds. However, since we hope this thesis will be useful to disciplines outside of computation, we include a very high-level overview of the computational ideas that are essential to this thesis, though we do assume some degree of mathematical maturity.

2.2.1 Algorithms

In computer science, an *algorithm* is a list of instructions to be performed by a computer [Knu97, CLRS09]. Computer scientists are often interested in whether an algorithm is correct, i.e., to what degree it achieves a specified objective, and also how long an algorithm takes to complete (or whether it terminates at all) and how much memory it requires. Runtime analysis looks at how much an algorithm's time depends on the size of the input data as it asymptotically approaches infinity. For example, if an algorithm is given a data input of size N and the algorithm's runtime is based on $2N + 4$ operations, the N term dominates asymptotically and the runtime is said to grow linearly with the input size, denoted as a runtime of $O(N)$. If the algorithm's runtime has a quadratic asymptotic dependence on the input size (e.g., a runtime based on $3N^2 + 5N + 7$ operations), then the runtime is $O(N^2)$. If the algorithm's runtime is constant and does not depend on the input size, then the runtime is $O(1)$. Similar analysis can be performed for memory usage in addition to runtime. We use this "big-O notation" throughout the thesis to describe how time and memory usage of an algorithm grows with the amount of input data.

2.2.2 Machine learning

Supervised learning

Machine learning uses "training data" combined with an objective function to determine the rules of an algorithm with the goal of making predictions or decisions [Ng, GBC16,

TCwC+07]. In *supervised machine learning*, the training data consists a set of samples where each sample has a set of features (e.g., the pixels in an image) paired with a set of labels (e.g., whether that image is of a cat or not a cat). A supervised machine learning algorithm is trained to predict the correct label(s) given a set of features. Predicting discrete labels is referred to as *classification* and predicting continuous labels is referred to as *regression*. Once trained, this algorithm can then assign labels to new, unlabeled data (e.g., given a new image, is it a cat or not?). Chapters 6 and 7 perform supervised learning in the context of drug discovery.

Unsupervised learning

An *unsupervised learning* algorithm only has access to training data where samples are described by a set of features but there are no labels available to the algorithm. Instead, the goal of unsupervised learning is to identify structures based on patterns in the dataset alone. A common unsupervised learning problem is clustering, in which an algorithm assigns a cluster label to a dataset such that "similar" samples are assigned to the same cluster. Clustering can help reduce the complexity of further analysis; rather than reason about millions of datapoints, a data analyst can spend time reasoning about tens of clusters. Unsupervised structure-finding is a key part of our discussions on understanding infectious disease, particularly in Chapters 3 and 4. We also use unsupervised learning to extract data related to viral escape from large unlabeled protein sequence corpuses in Chapter 8.

Representation learning

While we discussed that samples in a training set are described by a set of features, a lot of the success of a machine learning algorithm is determined by what these features are. Anything provided to a machine learning algorithm must be encoded in a way that is intelligible to a computer; for example, a natural scene can be encoded by red-green-blue color intensities in a two-dimensional coordinate system, i.e., an image. A similar encoding must happen for physical objects like an organic small molecule or a nucleic acid. The computational representation of an entity is called an *embedding*.

Representation learning constructs these embeddings based on data [BCV13], mostly using unsupervised learning approaches. Often, samples will first be encoded using relatively simpler, human-crafted features that make intuitive sense, e.g., a tensor of pixel values or a one-hot-encoded sequence matrix. Then, a machine learning algorithm will use patterns identified across a large training set to learn more general properties that are encoded into a learned embedding. A common modern technique for learning these embeddings is to train a neural network "autoencoder" that takes a sample as input and tries to reconstruct the same sample in its output; internal-layer outputs of the neural network can be interpreted as a learned embedding. We take a similar approach to constructing embeddings in Chapter 8. Learned embeddings can also be used as features in supervised learning, particularly when the embeddings can be extracted from a large unlabeled dataset and then reused in a supervised setting with limited labeled data. We rely on this approach, referred to as *pretraining* [EBC+10], in our methods in Chapter 7.

Active learning

Standard formulations of supervised and unsupervised learning assume that the training data is fixed. In *active learning*, not only does an algorithm make predictions, but it also interacts with a *teacher* that provides the algorithm with the labels of new samples during the training process. Often, queries to the teacher are expensive, so an active learning algorithm must decide what new samples to query based on an objective function. The most important consideration of this objective function is whether to query the teacher about new samples that are more similar or more different to those in the training data, which is called the *exploration/exploitation trade-off*. For example, consider the gold mining problem: given a geographic location, an algorithm tries to predict how much gold there is at that location. At first, it might be desirable to explore an entire geography for any signs of gold at all; then, if some sites yield a small amount of gold, it may make more sense to exploit regions close to those sites to see if an even greater amount of gold exists nearby. Active learning can also be applied to searching for biomolecular interactions, as described in Chapter 7.

2.2.3 Computational geometry

Many data-driven algorithms can be understood through a geometric interpretation [CLRS09]. A dataset can often be described as points in a geometric space; typically, a sample that is described by N real-valued features can be thought of as a point in $\mathbb{R}^N$. The most intuitive geometric space is $\mathbb{R}^3$, which can be used to describe the everyday physical space that we inhabit. The similarity of two points can be defined as the distance in the geometric space between those two points, where distance "metrics" are discussed in greater detail in Section 3.2.3.

A very common problem in computational geometry is how to efficiently process a large number of datapoints. Solutions often require recognizing patterns in the data to make it easier to search for one or more datapoints of interest. A common technique for doing so is to construct a *data structure* that organizes the points according to some property. For example, a data structure could partition the geometric space such that similar datapoints are grouped together within a partition, enabling locality-based search.

Another common problem is the "curse of dimensionality" when dealing with very high-dimensional data [AHK01]. As the number of dimensions increases linearly, the volume of the space increases exponentially, leading to problems that arise due to the complexity of the space and sparsity of observed datapoints within that space. A number of observations help to mitigate problems with high-dimensional datasets. First, while a dataset can have seemingly high dimension, the dataset could be described exactly or almost exactly using a much smaller number of "effective" dimensions. Data from many natural systems often "lie close to a low-dimensional manifold" in that a given point is often close to only a few other datapoints, so most of the geometric structure in the dataset can be captured by just considering these nearest-neighbor relationships [YDDB15]. This *manifold* assumption underlies *dimensionality reduction*, a common technique in computational geometry that attempts to preserve the "information" in a set of points but in a lower dimensional space.

There are many areas in which machine learning and computational geometry

intersect. For example, finding nearest neighbors, a geometric task described further in Section 3.2.4, is used as a subroutine as part of classification or regression in supervised learning. Or, as another example, unsupervised learning techniques that find axes of maximum variation can be used to perform dimensionality reduction.

Chapter 3

Understanding Disease I: Integration

What dependence maintaines any relation, between that arm which was lost in Europe, and that legge that was lost in Afrique or Asia, scores of yeers between? And still, still God knows in what part of the world every graine of every mans dust lies; and he whispers, he hisses, he beckens for the bodies of his Saints, and in the twinckling of an eye, that body that was scattered over all the elements, is sate down at the right hand of God, in a glorious resurrection.

—John Donne, marriage sermon (1627)

Before fighting an enemy, it helps to understand what the enemy is and how it operates. In the context of infectious disease, this looks like understanding the biology of the pathogen and how it infects and interacts with its host. Much of the systems biology underlying host-pathogen interactions is still unknown, but recent biotechnologies promise to advance biological knowledge by augmenting the traditional experimental lifecycle. Traditional biology relies on a set of targeted, reproducible experiments aimed at supporting or disproving a given expert hypothesis [Bac20, Pop59]. In contrast, modern biology often first conducts a massive, high-throughput experiment

that measures many features of many biological samples. Statistical analysis of the resulting massive, high-dimensional dataset highlights significant phenomena for more traditional experimental follow-up.

In systems biology, one of the most influential high-throughput technologies of the past five years is the ability to sequence the mRNAs within a single cell and to do so in a massively parallel way, producing expression profiles of all genes in the genome across ~10^3–10^6 cells in a single experiment [HPN+20]. This technology, called single-cell RNA-sequencing (scRNA-seq), has enabled researchers to characterize different types of cells in a biological sample based on their transcriptomes. The scRNA-seq dataset could then, for example, reveal new cell types or unknown transitional states that appear during cellular development that can be confirmed using orthogonal, traditional experiments.

This chapter first provides preliminary background on common computational tasks in single-cell data analysis and then expands on a fundamental problem: how do you compare transcriptomes from different datasets? For example, scRNA-seq technologies that differ in their underlying chemistry could have small yet systematic differences even when profiling the same tissue, resulting in downstream analysis confounded by technical biases. This chapter then describes Scanorama[1], a computational tool we developed based on efficient dimensionality reduction and nearest neighbors search [HBB19]. Scanorama constructs an embedding space in which cells from similar cell types are close, despite coming from disparate studies or experimental conditions, while dissimilar cells are kept far apart—a problem termed "integration" [HBB19]. Then, as a concrete application note, this chapter describes how data integration with Scanorama highlights unexpected transcriptomic similarities between the lifecycles of two single-celled eukaryotic pathogens: *Toxoplasma gondii*, which causes toxoplasmosis disease and altered neurological behavior, and *Plasmodium berghei*, which causes malaria in mice and is a model organism for the study of human malaria [XTR+20].

[1]Software available at `http://scanorama.csail.mit.edu` and at `https://github.com/brianhie/scanorama`.

3.1 Glossary

- *Single-cell RNA-sequencing (scRNA-seq).* A technology that measures gene expression within each of potentially millions of single cells.

- *scRNA-seq integration.* The problem of learning a shared representation across single cells, separated by confounding variables like experimental batch or donor-specific differences, such that similar cell types are close in representation embedding space and disparate cell types are far in embedding space.

- *Nearest neighbor search.* Given a set of points in a geometric space and a new query point, a nearest neighbor search algorithm will return the closest point in the set to the query point according to some distance metric.

- *Mutual nearest neighbors matching.* Points x in dataset $\mathcal{X}$ and y in dataset $\mathcal{Y}$ are mutual nearest neighbors if x is among the closest points in $\mathcal{X}$ to y and if y is among the closest points in $\mathcal{Y}$ to x; when $\mathcal{X}$ and $\mathcal{Y}$ are single-cell datasets, mutual nearest neighbors matches help reduce matches involving dataset-specific cell types.

3.2 Preliminaries

3.2.1 scRNA-seq technologies and preprocessing

Though a deep dive into the biochemistry of scRNA-seq is out of the scope of this thesis, a few high-level preliminaries into the technology are useful for understanding the underlying data. RNA-sequencing (RNA-seq) leverages modern technologies referred to as *next generation sequencing (NGS)* that allows researchers to read massive amounts of nucleic acid sequences from a biological sample [HPN$^+$20, ZTB$^+$17]. RNA-seq is often used to gain insight into the functional state of a biological sample based on which genes are or are not expressed. Though function is almost completely determined by proteins, protein expression is more difficult to measure in a high-throughput way, so

researchers leverage the central dogma to use RNA abundances as an approximation of protein abundances.

While a typical "bulk" RNA-seq experiment simply measures aggregate RNA in a biological sample, a scRNA-seq experiment aims to measure the RNA abundances for each cell in a sample individually. The goal of a scRNA-seq experiment is to produce a gene expression matrix describing gene abundances per cell. To do so, single cells from a biological sample (e.g., a bacterial colony or a tissue sample) must first be physically segregated. The two most popular technologies separate cells into nanoliter oil droplets or into microwells on a plate [ZTB+17, GWH+17]. Within each droplet or microwell is a unique RNA sequence, or a "barcode," that gets attached to each of the mRNAs within that droplet or microwell. After mRNAs are barcoded, they can then be sequenced using NGS technologies to yield a set of reads. A read is assigned to a gene based on its mRNA sequence and is assigned to a cell based on its barcode sequence. The result is a cell-by-gene matrix where the values are gene expression abundances that are based on the number of reads that map to a cell-gene pair. We will denote this matrix as $\mathbf{X} \in \mathbb{R}_{\geq 0}^{N \times M}$ where N is the number of cells and M is the number of genes.

3.2.2 Standard scRNA-seq data analysis

Single-cell data analysis relies of a number of computational techniques to facilitate biological interpretation. The high dimensionality of cellular profiles (e.g., ~20K genes for a human) complicates intuitive interpretation of the data and increases the computational burden of data analysis. Many *dimensionality reduction* techniques combine information across multiple genes into a compact set of features. Many such techniques are based on matrix decomposition models like principal component analysis (PCA; finds orthogonal features of maximum variation), independent component analysis (ICA; finds statistically independent features that best reconstruct the original data), or nonnegative matrix factorization (NMF; finds features, often interpreted as gene modules, that combine expression across multiple correlated genes); these methods are reviewed in the context of genomic data in reference [SOAC+18].

An important dimensionality reduction problem is *visualization*, i.e., learning a two- or three-dimensional embedding of each cell that captures some of the dataset structure in a more human-intuitive feature space [HPN+20, NBC21]. Visualizations of scRNA-seq data must then take the form a scatterplot in which each point corresponds to a single cell, which in many instances results in beautiful, pointillistic displays. A common visualization approach based on PCA plots cells along the two axes of maximum variation across cells. Nonlinear algorithms like t-SNE [vdMH08] and UMAP [MH18], both reviewed in reference [NBC21], aim to preserve "neighborhoods" of locally proximal datapoints from the original embedding space within the visualization space while allowing for greater distance distortion if two points are distal in the original embedding space.

Clustering is used to separate cells into these groups to allow for downstream comparison between groups, i.e., a clustering algorithm learns a function $f_{\text{cluster}} : \mathbb{R}^M_{\geq 0} \to \mathcal{C}$ that maps a gene expression profile $\mathbf{x}_i$ to a cluster in $\mathcal{C} \triangleq \{c_1, \ldots, c_L\}$; some clustering algorithms (e.g., k-means clustering) leave L, the number of clusters, as a user parameter, while others (e.g., Louvain clustering) also try to learn a value of L based on various heuristics. Different clustering algorithms for single-cell data are reviewed in reference [HPN+20]. Clusters obtained via unsupervised learning approaches are often interpreted as computationally-defined "cell types," since cells within the same cluster have more similar transcriptomes, which approximates more similar function.

3.2.3 Distance metrics

In reasoning about high-dimensional data, a very useful concept is the notion of a *distance metric*. A distance metric is a function d defined with respect to some set $\mathcal{X}$ that takes in two members of the set $\mathcal{X}$ and outputs a non-negative number, i.e., $d : \mathcal{X} \times \mathcal{X} \to [0, \infty)$ and where the following three axioms are satisfied:

1. *Identity of indiscernibles:* $d(x, y) = 0 \iff x = y$.

2. *Symmetry:* $d(x, y) = d(y, x)$.

3. *Triangle inequality:* $d(x, y) \leq d(x, z) + d(z, y)$.

The most intuitive distance metric is the three-dimensional Euclidean distance, which we use to measure physical distance in everyday life. For the Euclidean distance, $\mathcal{X} = \mathbb{R}^3$ and $d(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$, where a point $\mathbf{x}$ is a vector of three coordinates, x_1, x_2, and x_3, and $\mathbf{y}$ is defined similarly. A low distance means two points are close together, and a high distance means two points are far apart.

In high-dimensional data analysis, a given measurement is often represented by a vector $\mathbb{R}^M$. While perhaps less intuitive, it is possible to generalize the Euclidean distance to arbitrary dimensions as

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2 \triangleq \left(\sum_{i=1}^{M} (x_i - y_i)^2 \right)^{\frac{1}{2}}$$

for points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^M$. The Euclidean norm $\|\cdot\|_2$ is also called the ℓ_2 norm. A generalization of the Euclidean distance takes the form

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_p \triangleq \left(\sum_{i=1}^{M} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

for $p \geq 1$. Another widely used distance metric is when $p = 1$, which is called the Manhattan distance or the ℓ_1 norm.

3.2.4 Nearest neighbors search

For the work presented in this chapter, a particularly important algorithmic subroutine is *nearest neighbors search*. Given a set $\mathcal{X}$ of N points in $\mathbb{R}^M$, the goal of nearest neighbors search is to build a data structure that, given any point $\mathbf{y} \in \mathbb{R}^M$, returns a point in $\mathcal{X}$ that is closest to $\mathbf{y}$, i.e., its "nearest neighbor" in $\mathcal{X}$, based on a distance metric d. In single-cell analysis, d is typically induced by the ℓ_1 or the ℓ_2 norm (i.e., the "Manhattan" or the "Euclidean" distance, respectively) and is used to measure transcriptomic similarity. An extension of this problem is to return the top k nearest

neighbors (instead of the single nearest neighbor neighbor).

Naively, a single nearest neighbor query requires N comparisons each considering M dimensions, which is often an impractical computational cost. To reduce time, memory, or both, we can instead use approximate nearest neighbors (ANN) search where the goal is to return a point $\mathbf{x}' \in \mathcal{X}$ such that $d(\mathbf{x}', \mathbf{y}) \leq c \cdot \min_{\mathbf{x} \in \mathcal{X}} d(\mathbf{x}, \mathbf{y})$ for some constant $c \geq 1$; intuitively, the point returned by the ANN algorithm is similar in its distance to the query point $\mathbf{y}$ as the actual nearest neighbor. An influential approach underlying many ANN search algorithms is to partition $\mathbb{R}^M$ to enable recursive elimination of large portions of the search space [DF08]. ANN search is reviewed at length in reference [AIR18]. ANN search is critical for nearest neighbor search in single-cell applications since both N and M are often very large.

3.3 Toward single-cell panoramics

While individual scRNA-seq experiments can already provide insight into novel cell states or cellular differentiation trajectories, global efforts like the Human Cell Atlas are now generating large collections of scRNA-seq datasets that profile cells from diverse tissues, disease states, or organisms. Assembling large, unified reference datasets, however, may be compromised by differences due to experimental batch, sample donor, or experimental technology. Initial attempts at integrating scRNA-seq studies across multiple experiments assumed that all datasets share at least one cell type in common [HLMM18] or that the gene expression profiles share largely the same correlation structure across all datasets [BHS$^+$18]. These methods are therefore prone to overcorrection, especially when integrating collections of datasets with considerable differences in cellular composition.

We therefore developed Scanorama, a strategy for efficiently integrating multiple scRNA-seq datasets, even when they are composed of heterogeneous transcriptional phenotypes. Our approach is inspired by computer vision algorithms for panorama stitching that identify images with overlapping content and merge these into a larger panorama [DOR$^+$15]. Analogously, Scanorama automatically identifies scRNA-seq

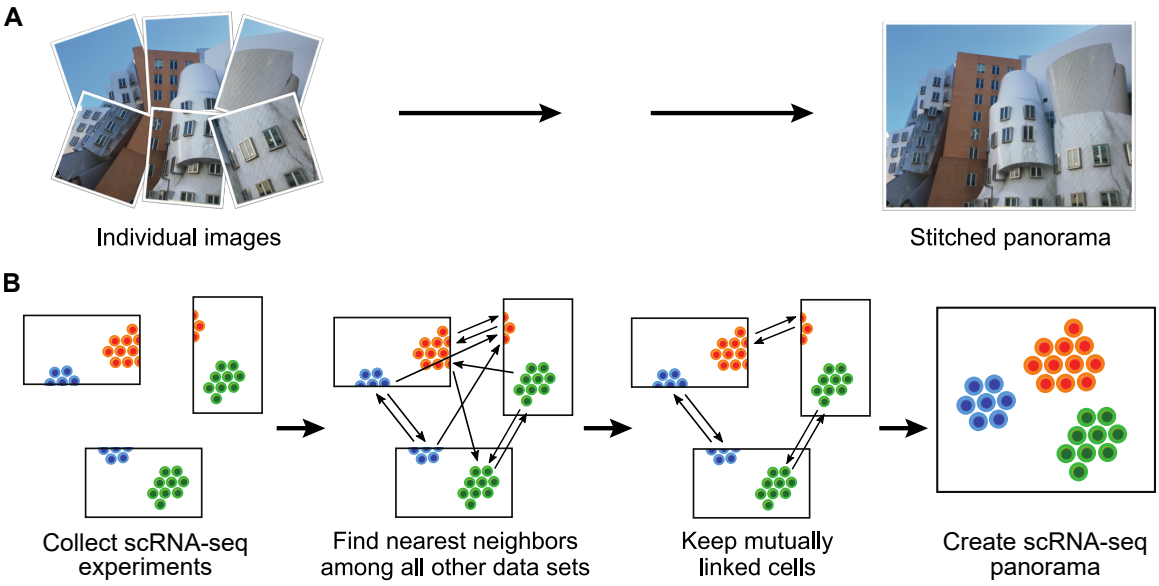

Figure 3-1: Panoramic dataset integration.

(**A**) A panorama stitching algorithm finds and merges overlapping images to create a larger, combined image. (**B**) A similar strategy can also be used to merge heterogeneous scRNA-seq datasets.

datasets containing cells with similar transcriptional profiles and can leverage those matches for batch-correction and integration, without also merging datasets that do not overlap (Figure 3-1). Scanorama is robust to different dataset sizes and sources, preserves dataset-specific populations, and does not require that all datasets share at least one cell population.

Our approach generalizes mutual nearest neighbors matching, a technique which finds similar elements between two datasets, to instead find similar elements among many datasets. Originally developed for pattern matching in images [DOR+15], finding mutual nearest neighbors has also been used to identify common cell types between two scRNA-seq datasets at a time [HLMM18]. However, to align more than two datasets, existing methods [BHS+18, HLMM18] select one dataset as a reference and successively integrate all other datasets into the reference, one at a time, which may lead to suboptimal results depending on the order in which the datasets are considered (Figure A-1). Although Scanorama takes a similar approach when aligning a collection of two datasets, on larger collections of data, it is insensitive to order and less vulnerable to overcorrection, because it finds matches between all pairs of

datasets.

To optimize the process of searching for matching cells among all datasets, we introduce two key procedures. Instead of performing the nearest neighbor search in the high-dimensional gene space, we compress the gene expression profiles of each cell into a low-dimensional embedding using an efficient, randomized singular value decomposition (SVD) [HMT11] of the cell-by-gene expression matrix, which also helps improve the method's robustness to noise. Additionally, we use an approximate nearest neighbor search based on hyperplane locality sensitive hashing [Cha02] and random projection trees [DF08] to greatly reduce the nearest neighbor query time both asymptotically and in practice. We describe these procedures in greater detail in the algorithm overview in the next section.

3.4 Scanorama: Algorithm details

3.4.1 Preprocessing and dimensionality reduction

We are given a collection of single-cell RNA-seq (scRNA-seq) datasets $\mathcal{D} \triangleq \{\mathbf{D}_1, \ldots, \mathbf{D}_d\}$. Each dataset $\mathbf{D}_i$ is represented by a gene expression matrix $\mathbf{X}_i \in \mathbb{R}_{\geq 0}^{N_i \times M_i}$ and a set of genes $\mathcal{G}_i$ where $|\mathcal{G}_i| = M_i$ and N_i is the number of cells in $\mathbf{D}_i$, where $i \in [d]$. Our goal is to identify datasets with similar cell types and optionally apply a batch correction that removes confounding differences in expression between these datasets. The expression values can either be relative expression values (e.g., RPKM or TPM) or absolute transcript counts (e.g., DGE from UMI experiments).

We merge the expression values into a matrix $\mathbf{X} \triangleq \begin{bmatrix} \mathbf{X}_1^{\mathrm{T}} & \ldots & \mathbf{X}_d^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}_{\geq 0}^{N \times M}$ where $N = N_1 + \cdots + N_d$ and $M = |\mathcal{G}_1 \cap \cdots \cap \mathcal{G}_d|$. For scale-invariant comparison between cells, we normalize the expression profiles of each cell to have a unit ℓ_2 norm, i.e.,

$$\mathbf{X}_{i,:} \triangleq \frac{\mathbf{X}_{i,:}}{\|\mathbf{X}_{i,:}\|_2}$$

for all $i \in [N]$. We reduce the dimensionality of the search space for our nearest neighbors queries by computing the SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$ to obtain the lower dimensional

matrix $\tilde{\mathbf{X}} \approx \mathbf{U}_{:,1:\kappa}\mathbf{\Sigma}_{1:\kappa,1:\kappa}$ where $\tilde{\mathbf{X}} \in \mathbf{R}^{N\times\kappa}$. We choose $\kappa = 100$ in our experiments as a conservative cutoff that preserves most of the variation in the data.

Since taking the full SVD is impractical for large values of M and N, we use a randomized SVD [HMT11] that only requires a constant number of linear passes, including a constant number of power iterations to improve approximation accuracy, over the full dataset of size $O(NM)$. Our experiments use only 2 power iterations to obtain the $\kappa+\delta$ most dominant components of the SVD, where δ is a small oversampling parameter also designed to improve the approximation accuracy, which we set to 2 in our experiments. We note that randomized SVD is generally insensitive to these parameters and is very accurate, with regard to the spectral norm approximation error, even after one power iteration and no oversampling [HMT11]. As a result, we obtain dataset-specific matrices with gene expression profiles all in a common low dimensional space, from which we obtain $\tilde{\mathbf{X}}_1, \ldots, \tilde{\mathbf{X}}_d$ where $\tilde{\mathbf{X}} \triangleq \begin{bmatrix}\tilde{\mathbf{X}}_1^{\mathrm{T}} & \ldots & \tilde{\mathbf{X}}_d^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$.

3.4.2 Mutual nearest neighbors search and matching

We identify datasets with shared functional patterns using a "mutual nearest neighbors" strategy originally developed for pattern matching in images, which has been shown to be robust to outliers and even nonlinear geometric distortions [DOR+15]. Mutual nearest neighbors matching has also been successful at aligning two biologically similar datasets from different batches [HLMM18], but we newly generalize this strategy to a large collection of biologically diverse datasets by searching for the nearest neighbors of the cells in one dataset among the cells in the remaining datasets.

More specifically, let $\tilde{\mathbf{X}}_i$ denote the low rank-approximated expression matrix of dataset $\mathbf{D}_i$ and let $\tilde{\mathbf{X}}_{\setminus i} \triangleq \begin{bmatrix}\ldots & \tilde{\mathbf{X}}_{i-1}^{\mathrm{T}} & \tilde{\mathbf{X}}_{i+1}^{\mathrm{T}} & \ldots\end{bmatrix}^{\mathrm{T}}$ be the expression matrix produced by the concatenation of all other expression matrices. We search for the nearest neighbors of cells in $\mathbf{D}_i$ (corresponding to the rows in $\tilde{\mathbf{X}}_i$) among the cells in $\mathcal{D} \setminus \{\mathbf{D}_i\}$ (corresponding to the rows of $\tilde{\mathbf{X}}_{\setminus i}$ by invoking the procedure

$$\mathcal{N}_i \triangleq \mathbf{NearestNeighbors}(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_{\setminus i}, k)$$

where $\mathcal{N}_i$ is a set of directed links $(\mathbf{x}_i, \mathbf{x}_j)$ such that $\mathbf{x}_i \in \mathbf{X}_i$, $\mathbf{x}_j \in \mathbf{X}_j$, $j \neq i$, and $\mathbf{x}_j$ is a k-nearest neighbor of $\mathbf{x}_i$, i.e., $|\mathbf{y}' \in \mathbf{X}_j, \forall j \neq i : |\mathbf{x}_i - \mathbf{y}'\| < \|\mathbf{x}_i - \mathbf{y}\|| < k$. We set k to a default value of 20 in our experiments as a balance between robustness to noise and overly permissive matching.

We repeat this procedure for each $\mathbf{D}_i \in \mathcal{D}$, obtaining sets of nearest neighbor links $\mathcal{N}_1, \ldots, \mathcal{N}_d$. We then match cells between two datasets iff they were mutually linked in the above procedure, i.e., we match $\mathbf{x}_i$ with $\mathbf{x}_j$ iff $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{N}_i$ and $(\mathbf{x}_j, \mathbf{x}_i) \in \mathcal{N}_j$, where we denote such a matching $\{\mathbf{x}_i, \mathbf{x}_j\}$ and the set of all matchings between datasets $\mathbf{D}_i$ and $\mathbf{D}_j$ (where $i \neq j$) as

$$\mathcal{M}_{ij} = \mathcal{M}_{ji} = \{\{\mathbf{x}_i, \mathbf{x}_j\} : (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{N}_i \wedge (\mathbf{x}_j, \mathbf{x}_i) \in \mathcal{N}_j\}$$

noting the symmetry in these matching sets. While computing the value of $\mathcal{M}_{ij}$ would naively take time in $O(k^2 N_i N_j)$, we apply hashing to query for the presence of a pair of cells in $\mathcal{N}_i$ and $\mathcal{N}_j$ in constant time, reducing the time to compute $\mathcal{M}_{ij}$ to $O(k \min\{N_i, N_j\})$, which we do for $O(d^2)$ possible matchings.

Since nearest neighbor queries are naively exponential in the size of the dimension, we improve the efficiency of our algorithm with an approximate nearest neighbors search that combines hyperplane locality sensitive hashing (LSH) [Cha02] and random projection trees [DF08]. The algorithm builds a search index over a reference dataset by randomly choosing two points in the reference and bisecting them with a hyperplane; doing this recursively on divided subsets of points forms a tree with a random hyperplane at each node, where multiple random trees can be constructed to increase the accuracy of a given query. Increasing the number of trees (which we set to 10) or increasing the search radius (which we set to 200 points) further decreases the approximation error. We make $O(N)$ such queries over the entire alignment procedure.

After matching cells between datasets, we put two datasets in the same panorama iff at least one of them has a large percentage of matched cells; specifically, we put $\mathbf{D}_i$

and $\mathbf{D}_j$ in the same panorama iff

$$r_{ij} \triangleq \max\left\{\frac{N_i^{\text{match}}}{N_i}, \frac{N_j^{\text{match}}}{N_j}\right\} \geq \alpha,$$

where $N_i^{\text{match}} \triangleq |\{\mathbf{x}_i : \mathbf{x}_i \in \mathbf{X}_i, \mathbf{x}_i \in \mathcal{M}_{ij}\}|$ and $N_j^{\text{match}} \triangleq |\{\mathbf{x}_j : \mathbf{x}_j \in \mathbf{X}_j, \mathbf{x}_j \in \mathcal{M}_{ij}\}|$ and where we set α to a nominal value of 10% based on observations of alignment scores across a large number of experiments and datasets. We note that α can be varied to be stricter or more permissive when merging panoramas. It may also be possible to learn a value of α from the data if some datasets are known to be similar or disparate. Once datasets have been matched, panoramas are formed by the connected components of the graph where each node is a dataset and an edge between two dataset nodes exists iff the r_{ij} alignment score threshold is met.

3.4.3 Panorama merging and batch correction

Once we identify panoramas, our method can optionally perform batch correction of the gene expression values using the cell matchings to guide the correction by using matched cell types to merge datasets together. Our merging procedure builds upon the technique in reference [HLMM18] that computes a set of Gaussian-smoothed translation vectors that can be added to expression values of one of the datasets that "corrects" for the difference between them.

More specifically, given two datasets $\mathbf{D}_i$ and $\mathbf{D}_j$ and a set of matchings $\mathcal{M}_{ij}$, we denote the expression values as $\mathbf{X}_i^{\text{match}} \in \mathbb{R}_{\geq 0}^{|\mathcal{M}_{ij}| \times M}$ and $\mathbf{X}_j^{\text{match}} \in \mathbb{R}_{\geq 0}^{|\mathcal{M}_{ij}| \times M}$ where the rows of $\mathbf{X}_i^{\text{match}}$ and $\mathbf{X}_j^{\text{match}}$ correspond to pairs of cells in $\mathcal{M}_{ij}$. The matching vectors are therefore the rows of $\mathbf{X}_j^{\text{match}} - \mathbf{X}_i^{\text{match}}$. Let $\mathbf{D}_i$ be the dataset for which we want to correct expression values. We compute weights between the cells in $\mathbf{D}_i$ and the matched cells in $\mathbf{D}_i$ as

$$[\mathbf{\Gamma}_i]_{ab} \triangleq \exp\left\{-\frac{\sigma}{2}\|[\mathbf{X}_i]_{a,:} - [\mathbf{X}_i^{\text{match}}]_{b,:}\|_2^2\right\}$$

where $[\cdot]_{ab}$ indicates the element in the ath row and bth column of a matrix, $[\cdot]_{a:}$

indicates the ath row of a matrix, and $\mathbf{\Gamma}_i \in \mathbb{R}^{N_i \times |\mathcal{M}_{ij}|}$ is a matrix of weights given by a Gaussian kernel function parameterized by σ, which we set to a nominal default value of 15, although we find our algorithm to be generally insensitive to this parameter. Finally, we construct the translation vectors as an average of the matching vectors with Gaussian-smoothed weights, where

$$\mathbf{v}_a \triangleq \frac{[\mathbf{\Gamma}_i]_{a,:}(\mathbf{X}_j^{\text{match}} - \mathbf{X}_i^{\text{match}}))}{(\sum_{b \in [|\mathcal{M}_{ij}|]} [\mathbf{\Gamma}_i]_{a,b}} \quad \text{and} \quad [\mathbf{X}_i']_{a,:} \triangleq [\mathbf{X}_i]_{a,:} + v_a$$

for all $a \in [N_i]$. The $\mathbf{X}_i'$ matrices are then returned by the algorithm as the corrected data.

Intuitively, the translation vector $\mathbf{v}_a$ for a cell a in $\mathbf{D}_i$ is computed as a linear combination of the matching vectors where the Gaussian kernel upweights the matching vectors closest to a. In addition to the batch correction described above, Scanorama also integrates the low dimensional embeddings in $\tilde{\mathbf{X}}$ using the exact same procedure based on the same sets of matched cells $\mathcal{M}_{ij}$ (but where we substitute $\tilde{\mathbf{X}}_i^{\text{match}}$ for $\mathbf{X}_i^{\text{match}}$).

Rather than hold the entire $\mathbf{\Gamma}_i$ matrix in memory, Scanorama can instead calculate the matching vectors $\mathbf{v}_a$ in a batched fashion that reduces a key memory bottleneck when aligning very large datasets. Scanorama can split up the matching matrix $\mathbf{X}_i^{\text{match}}$ into batches of size B so that the new weight matrix has dimension $N_i \times B$. The numerator and denominator of the $\mathbf{v}_a$ weighted average computation are accumulatively summed after each batch and the final normalization takes place only after all batches have been processed. The resulting matching vectors are equivalent in the full and the batched settings. We turn off the batched implementation of the matching vectors by default, but set B to 10,000 in our million-cell dataset experiment.

Each merge requires $O(M_i N_i |\mathcal{M}_{ij}|)$ computation and is therefore the most computationally expensive portion of our procedure. This runtime could be reduced by limiting the number of batch corrected genes, i.e., M_i, to a constant number of highly variable genes or by down-sampling to lower the number of matching vectors involved, i.e., $|\mathcal{M}_{ij}|$. In our experiments, we use all matching vectors, apply batch

correction to the top 10,000 most highly variable genes according to their dispersion (mean-to-variance ratio), and use a vectorized implementation that takes advantage of system parallelism, where we distribute our computation across 10 cores.

Once we have this merging procedure, we can use it to build up a set of panoramas by considering pairs of datasets $\mathbf{D}_i$ and $\mathbf{D}_j$ in decreasing order of the r_{ij} alignment scores. The first $\mathbf{D}_i$ and $\mathbf{D}_j$ are merged together to initialize a panorama and successive pairs are considered. If a successive $\mathbf{D}_i$ and $\mathbf{D}_j$ are not in any panorama, they are merged and placed in a new panorama. If $\mathbf{D}_i$ is in a panorama but $\mathbf{D}_j$ is not, then $\mathbf{D}_j$ is merged into $\mathbf{D}_i$'s panorama, or vice versa. If both $\mathbf{D}_i$ and $\mathbf{D}_j$ are already in panoramas, then their matchings $\mathcal{M}_{ij}$ are used to merge $\mathbf{D}_i$'s panorama with $\mathbf{D}_j$'s panorama (this occurs even if $\mathbf{D}_i$'s panorama is the same as $\mathbf{D}_j$'s panorama). This continues until all pairs of aligned datasets have been considered, after which we terminate and return the batch corrected datasets $\mathbf{D}_1, \ldots, \mathbf{D}_d$.

3.5 Empirical performance of Scanorama

The bulk of our evaluation of the Scanorama algorithm leverages empirical benchmarks on real scRNA-seq datasets. We compare Scanorama to its preceding integration algorithms, Seurat CCA and scran MNN, as well as to unintegrated data. We compare both the ability for Scanorama to integrate similar cell types across studies while preserving biological differences; we also compare runtime and memory usage to the preceding algorithms. We also reference an independent, comprehensive benchmarking study [LBC+20] that compared Scanorama to nine other integration methods (many which appeared after the publication of Scanorama) across seven diverse datasets. In all cases, we demonstrate strong empirical performance, which is described in the remainder of this section.

3.5.1 Simulations and toy datasets

To verify the merit of our approach, we first tested Scanorama on simulated data and a small collection of scRNA-seq datasets. We simulated three datasets with four cell

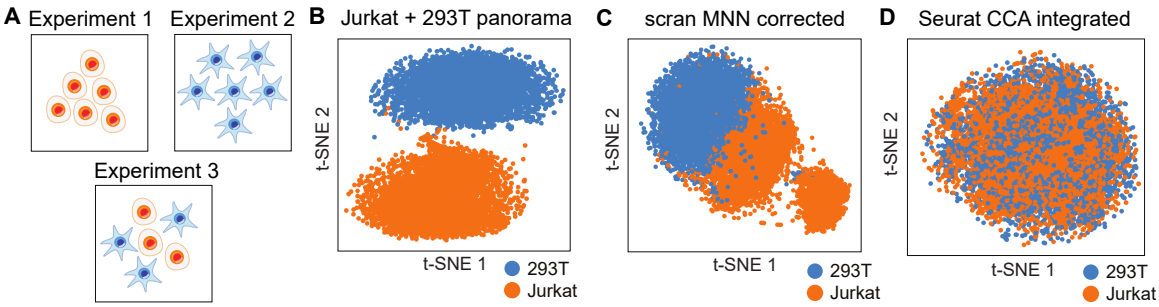

Figure 3-2: Scanorama does not depend on integration order.

(**A**) We apply Scanorama to a collection of three datasets: one entirely of Jurkat cells, one entirely of 293T cells and a 50/50 mixture of Jurkat and 293T cells. (**B**) Our method correctly identifies Jurkat cells (orange) and 293T cells (blue) as two separate clusters. (**C**, **D**) Existing methods for scRNA-seq dataset integration are sensitive to the order in which they consider datasets (Figure A-1) and can incorrectly merge disparate cell types.

types in total but where the first and third datasets had no cell types in common (Figure A-2A,E). We also obtained three previously-generated real datasets: one of 293T cells, one of Jurkat cells, and one with a 50:50 mixture of 293T and Jurkat cells (Figure 3-2A).

In both cases, we were able to merge common cell types across datasets (Figure 3-2B; Figure A-2B,F) without also merging disparate cell types together. In contrast, existing integration methods are either sensitive to the order in which datasets are considered or are highly prone to overcorrection (Figure 3-2C,D; Figure A-2C,D,G,H). Scanorama's improved performance on the simulated datasets and the real 293T/Jurkat collection, while relatively idealized or simple cases, led us to consider if we could also achieve improved performance on larger and more complex collections of scRNA-seq datasets.

3.5.2 105,476 cells across 26 diverse datasets

We then sought to demonstrate the ability of Scanorama to assemble a larger and more diverse set of cell types. In total, we ran our pipeline on 26 scRNA-seq datasets representing nine different technologies and containing a total of 105,476 cells (Figure 3-3A and Table A.1), each dataset coming from a different scRNA-seq experiment from a total of 11 different studies. Scanorama identifies datasets with the same cell

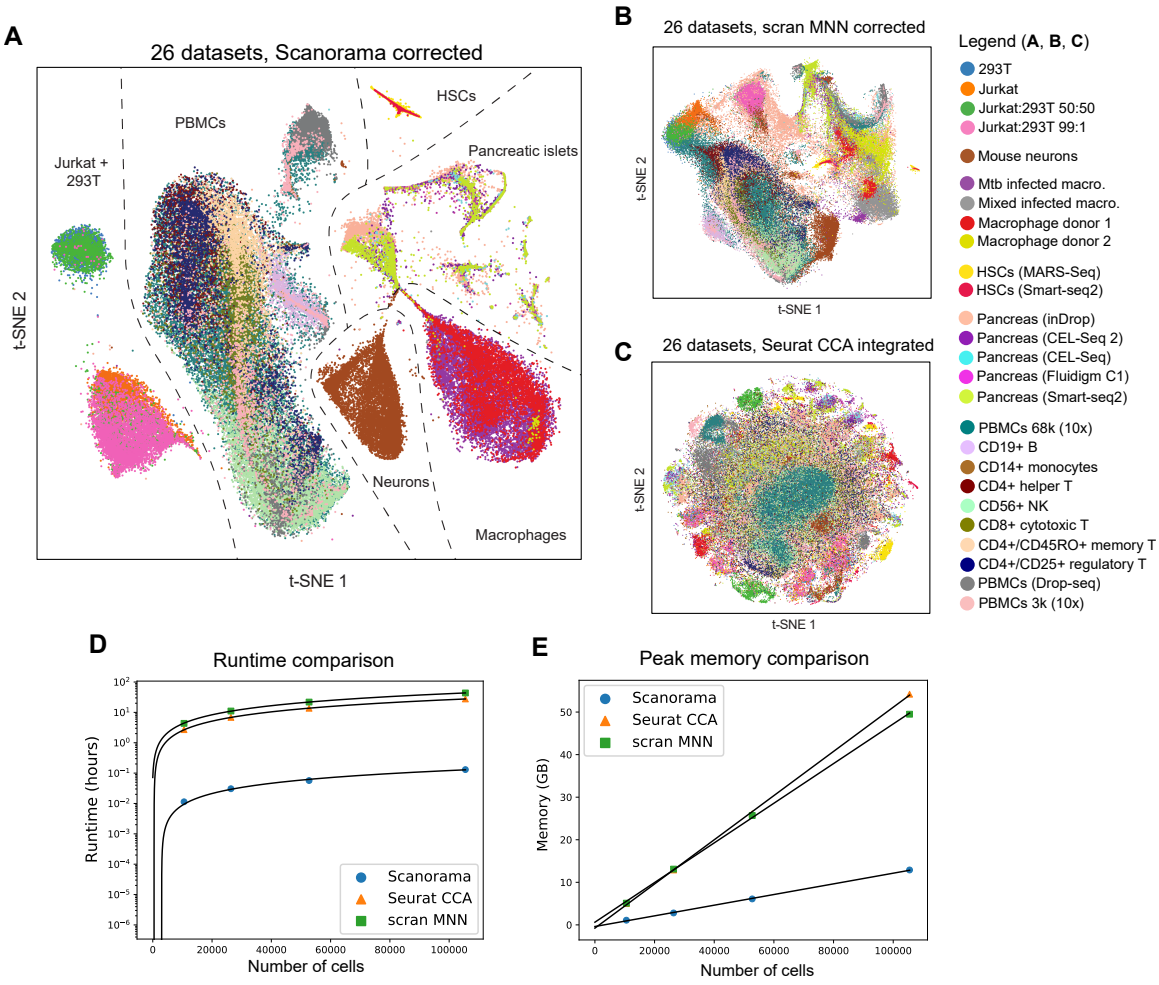

Figure 3-3: Panoramic integration of 26 heterogeneous single-cell datasets.

(**A**) t-SNE visualization of 105,476 cells after batch correction by our method, with cells clustering by cell type instead of by batch. (**B**, **C**) Other methods for scRNA-seq dataset integration are not designed for heterogeneous dataset integration and therefore naively merge all datasets into a single large cluster (Figure A-9). (**D**, **E**) Scanorama efficiently integrates 105,476 cells across 26 datasets in less than 6 min and in under 12 GB of RAM.

types and merges them together such that they cluster by cell type instead of by experimental batch (Figures 3-3A,B,C and A-3). In contrast with existing methods, our algorithm does not merge disparate cell types together (Figure 3-3B,C) and identifies a "negative control" dataset of mouse neurons as distinct from the cell types of all other datasets (Figure 3-3A).

One of the panoramas identified by Scanorama consists of two datasets of hematopoietic stem cells (HSCs) [PAG+15, NHP+16] which, once corrected for batch effects and plotted along the first two principal components, reconstruct the expected HSC differentiation hierarchy (Figure A-4). We also observe cell type-specific clusters within panoramas of pancreatic islet cells (Figures A-5 and A-6) and peripheral blood mononuclear cells (Figure A-7) [ZTB+17] but now have greater power to detect rare cell populations. For example, in the pancreatic islet panorama, we observe a cluster of cells consistent with a previously-reported rare subpopulation of pancreatic beta cells marked by increased expression of endoplasmic reticulum (ER) stress genes *GADD45A* and *HERPUD1* (Figure A-5).

We also note that datasets are aligned according to biological similarity instead of confounding differences in transcriptional quiescence such as dataset-specific dropouts (Figure A-8). Scanorama also aligns biologically similar datasets across experiments that use absolute transcript counts or relative expression values; e.g., the pancreatic islet panorama consists of UMI experiments [BVW+16, MDG+16, GMB+16] and datasets with TPM and RPKM values [LGB+17, SPE+16].

3.5.3 Quantifying integration performance

We sought to quantify the integration performance of our algorithm on the collection of 26 datasets by calculating a Silhouette Coefficient [Rou87] for each cell. Silhouette Coefficient computation first requires cluster label assignments $\mathcal{C} \triangleq \{c_1, \ldots, c_L\}$. The Silhouette Coefficient first makes use of the mean of the distances from cell $\mathbf{x}$ to all

other cells of the same type, i.e., when $\mathbf{x} \in c_i, i \in [L]$,

$$\mu(\mathbf{x}) \triangleq \frac{1}{|c_i| - 1} \sum_{\mathbf{y} \in c_i, \mathbf{x} \neq \mathbf{y}} d(\mathbf{x}, \mathbf{y})$$

for a distance metric d. The Silhouette Coefficient compares $\mu(\mathbf{x})$ to the mean *nearest-cluster* distance to $\mathbf{x}$, more precisely,

$$\nu(\mathbf{x}) \triangleq \min_{c \in \mathcal{C}, c \neq c_i} \frac{1}{|c|} \sum_{\mathbf{y} \in c} d(\mathbf{x}, \mathbf{y}).$$

The Silhouette Coefficient $s(\mathbf{x})$ for $\mathbf{x}$ is then

$$s(\mathbf{x}) \triangleq \frac{\nu(\mathbf{x}) - \mu(\mathbf{x})}{\max\{\mu(\mathbf{x}), \nu(\mathbf{x})\}},$$

taking values between 1 and -1, inclusive, where higher values indicate better clustering performance. Intuitively, the Silhouette Coefficient improves if a cell is close to other cells of the same type and far from all other cells of different types.

On the above collection of 26 datasets, the distribution of Silhouette Coefficients was significantly higher (two-sided, independent t-test $P < 4 \times 10^{-6}$; n = 105,476 cells) after Scanorama integration (median of 0.17) compared to scran MNN (median of -0.03), Seurat CCA (median of -0.18), and no integration (median of 0.14) (Figure A-9). Clustering analyses of Scanorama-integrated data found structure related to cell type and orthogonal to dataset-specific batch (Figure A-6A,B,C), with comparable integration performance to existing methods when all datasets have similar cell type compositions (Figure A-4) and significantly better integration performance than existing methods on collections of datasets with cell type heterogeneity (Figures 3-3, A-1, and A-2).

In addition to integration performance, we can also quantify the batch correction performance of our algorithm by looking at the similarity of the gene expression distributions across datasets before and after batch correction. On five pancreatic islet datasets, for each gene, we calculated the one-way ANOVA F-value testing the null hypothesis that there are equal gene expression means among all five datasets, where

lower F-values indicate more similar means. We computed F-values for each gene in the uncorrected data and after batch correction by Scanorama and scran MNN (we note that this analysis is not applicable to the output of Seurat CCA since it only does integration, not batch correction, and therefore does not modify gene expression values).

We found that 89% of the genes have lower F-values after Scanorama correction (Figure A-6D) compared to only 76% of the genes after scran MNN correction (Figure A-6E), while the variances across genes after Scanorama or scran MNN correction are still very similar to those of the uncorrected data (Scanorama Pearson $\rho = 0.97$; scran MNN Pearson $\rho = 0.99$; $P < 5 \times 10^{-324}$ for both methods; $n = 15{,}369$ genes), indicating that either method is not achieving lower F-values by trivially homogenizing gene expression.

3.5.4 Scalability: Integrating 1 million cells

Due to our algorithmic optimizations, our tool is also substantially more efficient than existing methods for scRNA-seq dataset integration or batch correction. In particular, to integrate our collection of 26 datasets containing 105,476 cells, Scanorama can integrate datasets in roughly five minutes and performs batch-correction of all panoramas in under 20 minutes. In contrast, existing methods require more than 27 hours to integrate the same collection of datasets (Figure 3-3D) using more than three times the amount of memory (Figure 3-3E) yet perform poorly at preserving real biological heterogeneity in the integrated result (Figure 3-3B,C).

We further demonstrate the scalability of our method by applying Scanorama to integrate 1,095,538 cells from two large-scale single-cell transcriptomic studies of the central nervous system (CNS) in mouse, including samples taken from the mouse spinal cord and from different regions of the mouse brain [ZHL$^+$18, SMW$^+$18]. Scanorama aligns functionally similar cells across different regions of the brain, where we can identify cell types using known marker genes (Figure 3-4). Scanorama integrates this collection of 1,095,538 cells in 9.1 hours with a peak memory usage of 95 GB, though additional optimizations may improve the efficiency of our method further. In contrast,

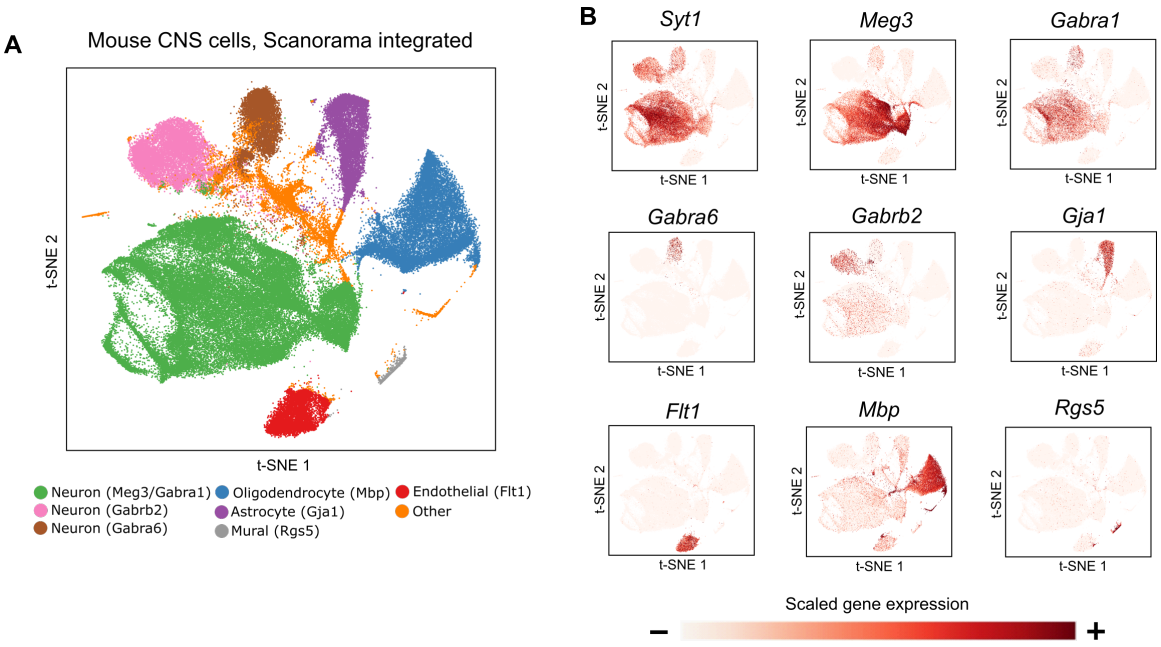

Figure 3-4: Scanorama scales to more than a million cells.

(**A**) Scanorama integrates a collection of 1,095,538 cells from the mouse brain and spinal cord. (**B**) Marker gene expression reveals cell type-specific clusters including *Syt1*, *Meg3*, *Gabra1*, *Gabra6* and *Gabrb2* in neurons, *Gja1* in astrocytes, *Flt1* in endothelial cells, *Mbp* in oligodendrocytes, and *Rgs5* in mural cells.

other methods exceed the maximum memory capacity of our benchmarking hardware when run on this data, illustrating the advantage of our algorithm's computational efficiency when integrating large-scale datasets containing millions of cells.

3.5.5 Robustness to overcorrection

Theoretically, Scanorama relaxes the requirement in scran MNN that all datasets share at least one cell type in common, instead only requiring that each dataset shares at least one cell type with at least one other dataset. However, in practice, we find that even this assumption is often too strict and that Scanorama can avoid overcorrection when a dataset has no overlapping cell types with any other dataset (e.g., mouse neurons among the collection of 26 diverse datasets; Figure 3-3A). Although Scanorama essentially reduces to the algorithm used in scran MNN when aligning a single pair of datasets together (although with much greater computational efficiency), we observe that Scanorama can be robust to overcorrection when integrating a larger collection

of datasets even when none of the datasets being integrated have overlapping cell types (Figure A-10). In principle, forming spurious mutual links between biologically disparate cell types becomes less likely as the number of cells or the number of datasets being integrated increases, so that Scanorama's approach becomes more robust to overcorrection with more data. Some amount of supervision, however, is still recommended when integrating heterogeneous datasets, and further minimizing the likelihood of overcorrection is an important concern for future integrative approaches.

3.5.6 Top performance in a comprehensive benchmark

Comprehensive independent evaluation by Luecken et al. [LBC+20] of Scanorama alongside nine other scRNA-seq integration tools and strategies showed that Scanorama occupies the top tier of integration methods, and is one of three integration methods that they recommend to practitioners. On a benchmark dataset of immune cells, different configurations of Scanorama occupied four out of the seven top method configurations, including the top-ranked position. Overall, across seven different real and simulated scRNA-seq datasets, different configurations of Scanorama occupied the second and fifth positions out of 38 total integration method configurations. This benchmarking study in particular highlighted the ability of Scanorama to preserve dataset-specific, biological signal in addition to removing confounding variation. These results helped highlight the strong empirical performance of Scanorama, particularly as a way of removing technical biases while preserving biological variation, which was an important emphasis of the benchmarking study.

3.6 Application note: Aligning pathogen lifecycles

The empirical benchmarks that establish Scanorama as a state-of-the art integration approach have mostly focused on human or mammalian cells, but since the central dogma of molecular biology extends across all cellular life, Scanorama can just as

easily be applied to better understand cellular pathogens as well. One particularly compelling application of Scanorama integration [XTR+20] looked for commonalities in the asexual reproductive lifecycle of two different protozoan pathogens: *Toxoplasma gondii* (*T. gondii*) and *Plasmodium berghei* (*P. berghei*).

In humans, *T. gondii* infection rarely leads to disease but does establish latent infection in an estimated 30–50% of the global population. However, serious or fatal disease, called toxoplasmosis, does occur particularly in severely immunocompromised individuals, like those with acquired immunodeficiency syndrome (AIDS). Interestingly, *T. gondii* has been shown to alter the behavior of mice so that they are less averse to cat urine, potentially making mice more susceptible to predation. Identifying if *T. gondii* infection in humans also leads to neurological effects is an active area of research [TM17].

P. berghei is part of the genus *Plasmodium* that is the cause of malaria disease, which led to an estimated ~400K human deaths in 2018 primarily in sub-Saharan Africa [Wor19]. While the primary parasite involved in human malaria is *Plasmodium falciparum*, *P. berghei* is a widely used model organism since it causes malaria in laboratory mice. Many drug and vaccine screening platforms therefore also use *P. berghei* as an experimental model [HRA+19].

While *T. gondii* and *P. berghei* are different organisms with different mechanisms of reproduction (*Toxoplasma* undergoes "endodyogeny," a simpler cell division process, whereas *Plasmodium* undergoes "schizogony," a more complex dividing process), they are still both protozoan (more specifically, apicomplexan) human pathogens. Similarities in cellular state between these pathogens could in turn reveal common mechanisms of reproduction and potentially lead to drugs that target shared replication biology in both pathogens.

Xue et al. performed Scanorama integration of scRNA-seq data from *T. gondii* [XTR+20] and *P. berghei* [HRA+19] based on a set of 1,830 orthologous genes, revealing striking similarity in the lifecycles of both pathogens (Figure 3-5). Xue et al. identified correspondences in all stages of cell division: for example, Scanorama aligns the "schizont" stage of *P. berghei* replication with the "M" and "C" stages of *T. gondii*

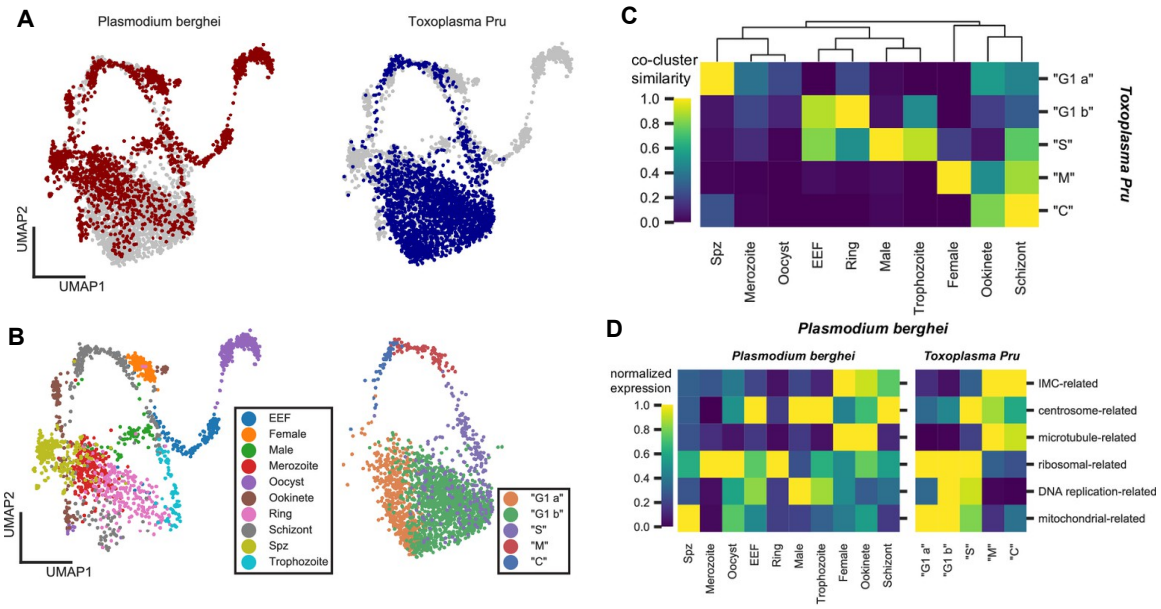

Figure 3-5: Comparative analysis of *Toxoplasma* and *Plasmodium*.

(**A**) Scanorama integration of *Plasmodium berghei* (red, left) and *Toxoplasma gondii* (blue, right). (**B**) Cell cycle of *Toxoplasma* is well-aligned to the erythrocytic cycle of *Plasmodium berghei*, despite fundamental differences in cell cycle progression between these two apicomplexans. Each cell is colored by the original cluster assignment in the corresponding dataset. (**C**) Normalized cluster similarity between the original cluster assignment of *Plasmodium berghei* and *Toxoplasma*. Cluster similarity is calculated by quantifying the fraction of cells that overlap in topological network in each cluster of the corresponding dataset. (**D**) Heatmap of concerted gene sets expression normalized to one within each cluster of cells in *Plasmodium* (left) and *Toxoplasma* (right).

From Xue et al., *eLife* (2020) [XTR+20], used with permission and under a Creative Commons Attribution license.

replication, in which both they show a transcriptomic program in both pathogens consistent with active intracellular remodeling. Correspondences in other states are visualized in Figure 3-5C,D. These results offer an excellent illustration of how Scanorama can help researchers understand common patterns in pathogen biology that could ultimately lead to shared therapeutic strategies.

More generally, Scanorama provides a powerful and efficient integrative framework that is robust to differences in cell type and sensitive to subtle functional changes across a diversity of tissues, organisms, biological conditions, technologies, dataset sizes and different levels of data quality and noise. Many other infectious disease systems could be analyzed using Scanorama, including bacterial pathogens or immune cells at different stages of viral infection. As researchers work to assemble a more complete picture of diverse biological function at a single-cell resolution, the need to integrate heterogeneous experiments also increases. Scanorama provides a robust and efficient solution to this problem.

Chapter 4

Understanding Disease II: Sketching

It took me four years to paint like Raphael, but a lifetime to paint like a child.

—Pablo Picasso, paraphrase (1958)

In the previous chapter, we developed a state-of-the-art method for integrating millions of high-dimensional biological samples (e.g, single-cell transcriptomes). The ability to assemble large single-cell panoramas, however, introduces another problem, namely that complex downstream analysis often has difficulty *scaling* to large dataset sizes and prevents rapid exploratory data analysis. Moreover, large amounts of information redundancy may bias analysis to the abundant information while ignoring less abundant but significant signal in the data. For example, unsupervised clustering of a single-cell dataset that is dominated by a common cell type might fail to identify structure from rare cell types or states. In the host-pathogen setting, for example, most of the immune cells in a blood sample could be inactivated or anti-inflammatory, but a researcher is particularly interested in the rarer populations of activated or inflammatory cells.

In this chapter, we continue to develop algorithms for improving our understanding of complex, heterogeneous biological systems. We are particularly concerned with the

setting in which we want to analyze a large of amount of data, e.g., that generated by a high-throughput technology like scRNA-seq. In such cases, rather than consider all the data, a *sketching* algorithm first selects a representative subset, or a sketch, of the data for downstream analysis. The remaining computational analysis would then need to consider a much smaller number of datapoints, improving computational scalability.

This chapter first uses the problem of rare cell type discovery in scRNA-seq to motivate a diversity-preserving sketching algorithm. Our approach is based on the insight that the volume of a cell type in transcriptomic space approximates its biological diversity. This leads to the *geometric sketching*[1] algorithm [HCD+19], which achieves efficient (near-linear time scalability) by leveraging this insight and combining it with the realization that biological data actually has a low "intrinsic" dimension, i.e., it lies close to a manifold with a dimension much lower than the superficially high original dimension. We analyze our algorithm theoretically, apply it to sketch large transcriptomic datasets, and demonstrate a practical application in rare, inflammatory cell type discovery in a dataset of immune cells from the umbilical cord blood.

4.1 Glossary

- *Scalability.* How a system responds to increasing amounts of work; in computational settings, this is often thought of in terms of the amount of time and memory a program requires, and how time and memory relate to the size of the program's input data.

- *Sketch.* A smaller subset of elements from a larger dataset. Typically used to accelerate a given analysis while preserving the accuracy of the analysis results.

- *Cover, covering.* In the geometric sketching setting, a set of shapes in the transcriptomic space that collectively contains all of the cells in a dataset.

[1] Software available at `http://geosketch.csail.mit.edu` and at `https://github.com/brianhie/geosketch`.

- *Hypercube.* A generalization of a cube (with equal side lengths) to many dimensions.

4.2 Preliminaries

(Preliminaries related to scRNA-seq technologies and standard data analysis, described in Section 3.2, may also be helpful to read alongside the preliminary information below.)

4.2.1 The scalability challenge

scRNA-seq experiments routinely profile hundreds of thousands of cells, with billions of cells likely to be profiled in the near future. Deriving biological insights from single-cell datasets requires computationally intensive operations such as clustering, visualization, and nonlinear data integration. Clustering analyses assign more similar cells to groups, or clusters, that may correspond to biologically meaningful structure. Visualization lets researchers develop an intuition about variation in a dataset by highlighting important variability within an interpretable, usually two-dimensional, plot. Data integration requires searching for similar transcriptomic structure across two or more datasets and removing confounding differences like batch effects. Performing these analyses on very large datasets is already not feasible for many researchers without expensive computational infrastructure, and is still time consuming for researchers with enough compute power. Instead, researchers often perform initial analysis on a random subset of cells chosen with uniform probability for each cell, which is prone to removing rare cell types and negates the advantage of performing large-scale experiments.

4.2.2 A geometric interpretation

In this chapter, we leverage an understanding of a single-cell dataset as a collection of points in a multidimensional "transcriptomic space." Each point in a dataset corresponds to a single cell and its location is determined by measuring gene expression.

The abstraction of points within a multidimensional space enables us to reason about the "geometry" of a scRNA-seq dataset, including the particularly useful concepts of distance and volume. Cells with closer distances in the transcriptomic space have greater transcriptomic similarity. Similarly, a shape that occupies a greater volume of the transcriptomic space represents greater transcriptomic variation.

4.3 Sketching: Motivation and overview

Improvements in the throughput of single-cell profiling experiments, especially droplet-based single-cell RNA-sequencing (scRNA-seq), have resulted in datasets containing hundreds of thousands of cells or even millions of cells [AST+17, ZTB+17, COP+20], with hundreds to thousands of gene expression measurements per cell. As these sequencing pipelines become cheaper and more streamlined, experiments profiling tens of millions of cells may become ubiquitous in the near future [AST+17], and consortium-based efforts like the Human Cell Atlas plan to profile billions of cells [RTL+17].

Leveraging this data to improve our understanding of biology and disease will require merging and integrating many cells across diseases and tissues [HBB19], resulting in reference datasets with massive numbers of cells. Unfortunately, the sheer volume of scRNA-seq data being generated is quickly overwhelming existing analytic procedures, requiring prohibitive runtime or memory usage to produce meaningful insights. This bottleneck limits the utility of these emerging large datasets to researchers with access to expensive computational infrastructure, and makes quick exploratory analyses impossible even for these researchers.

4.3.1 Current approaches

This chapter develops an approach that intelligently selects a small subset of data (referred to as a "sketch") that comprehensively represents the transcriptomic heterogeneity within the full dataset. Because of their vastly reduced computational overhead, our sketches can be efficiently shared among researchers and be used to

quickly identify important patterns in the full dataset to be followed up with in-depth analyses. Currently, by far the most common approach is to uniformly downsample a dataset to obtain a small subset to accelerate the initial data analysis. Although this simple approach could be used to generate sketches of single-cell datasets, it is highly prone to removing rare cell types and negates the advantage of performing large-scale scRNA-seq experiments in the first place.

Alternative sampling approaches that better consider the structure of the data, including k-means++ sampling [AV07] and spatial random sampling (SRS) [RA17], have not yet been applied to the problem of obtaining informative sketches of scRNA-seq data to our knowledge. However, these data-dependent sampling techniques not only lack the ability to efficiently scale to large datasets, but also lack robustness to different experimental settings and produce highly unbalanced sketches that are ill-suited for downstream scRNA-seq analyses as we demonstrate in our experiments.

4.3.2 Geometric insight

The key insight behind our sampling approach is that common cell types form dense clusters in the gene expression space, while rarer subpopulations may still inhabit comparably large regions of the space but with much greater sparsity. Rather than sample cells uniformly at random, we sample evenly across the transcriptomic space, which naturally removes redundant information within the most common cell types and preserves rare transcriptomic structure contained in the original dataset. We refer to our sampling method as "geometric sketching" because it obtains random samples based on the geometry, rather than the density, of the dataset (Figure 4-1).

Geometric sketching is extremely efficient, sampling from datasets with millions of cells in a matter of minutes and with an asymptotic runtime that is close to linear in the size of the dataset. We empirically demonstrate that our algorithm produces sketches that more evenly represent the transcriptomic space covered by the data. We further show that our sketches enhance and accelerate downstream analyses by preserving rare cell types, producing visualizations that broadly capture transcriptomic heterogeneity, facilitating the identification of cell types via clustering, and accelerating

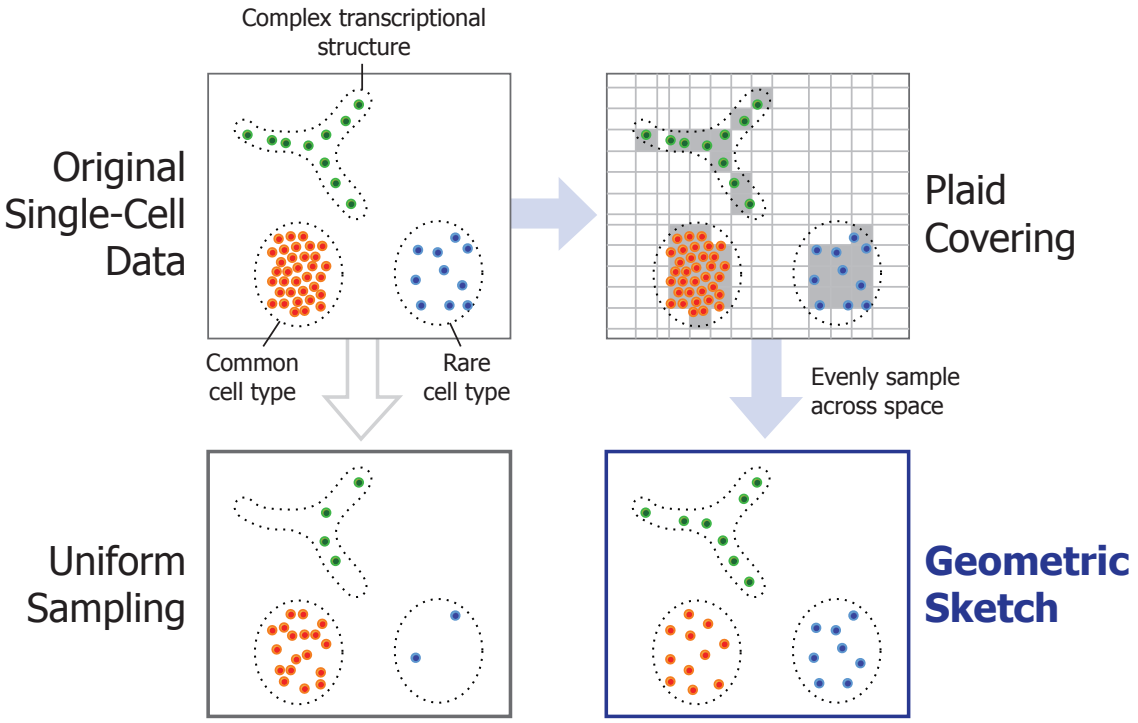

Figure 4-1: Geometric sketching overview.

We first cover the data points with equal-sized boxes (which we refer to as a "plaid covering") to approximate their geometry, then sample data points by first spreading the desired total sample count over the boxes as evenly as possible, followed by choosing the assigned number of samples within each box uniformly at random. The resulting sketch more evenly covers the landscape of the data compared to uniform sampling of points, where the latter is more prone to omitting rare cell types or transcriptomic patterns.

integration of large scRNA-seq datasets. Moreover, we demonstrate how the sensitivity of geometric sketching to rare transcriptomic states allows us to identify a previously unknown rare subpopulation of inflammatory macrophages in a human umbilical cord blood dataset, providing insight into a fundamental immunological process. As the size of single-cell data grows, geometric sketching will become increasingly crucial for the democratization of large-scale single-cell experiments, making key analyses tractable even for researchers without expensive computational resources.

4.4 Geometric sketching: Algorithm details

4.4.1 Problem definition

We first give a mathematical formulation of the sketching problem to elucidate the theoretical insights underlying our approach. Let $\mathcal{X} \triangleq \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a representation of a single-cell dataset, consisting of M-dimensional measurements $\mathbf{x}_i \in \mathbb{R}^M$ from N individual cells. In the case of very large N (e.g., millions of cells) [COP+20], it is often desirable to construct a sketch $\mathcal{S} \subset \mathcal{X}$ (i.e., a downsampled dataset), which can be more easily shared with other researchers and be used to quickly understand the salient characteristics of $\mathcal{X}$ without paying the full computational price of analyzing $\mathcal{X}$.

Drawing insight from computational geometry, we measure the quality of a sketch $\mathcal{S}$ with respect to a dataset $\mathcal{X}$ via the Hausdorff distance d_H [Hau37] defined as

$$d_H(\mathcal{X}, \mathcal{S}) \triangleq \max_{\mathbf{x} \in \mathcal{X}} \{\min_{\mathbf{s} \in \mathcal{S}} d(\mathbf{x}, \mathbf{s})\},$$

where d denotes the distance function of the underlying metric space (i.e., a notion of dissimilarity between two cells). Intuitively, d_H measures the distance of the cell in the original dataset that is farthest away from any of the cells included in the sketch. The lower this distance, the more comprehensively our sketch covers the original dataset.

We are interested in developing an efficient algorithm for obtaining $\mathcal{S}$ of a predetermined size k (i.e., $|\mathcal{S}| = k$) that minimizes $d_H(\mathcal{X}, \mathcal{S})$. A key property of our approach

is that it is agnostic to local density of data points, since only the maximum distance is taken into account. As a result, our sketches more evenly cover the space of gene expression spanned by the original dataset. In contrast, approaches based on uniform sampling or distance-based sampling (e.g., k-means++ [AV07]) are biased toward selecting more cells in densely populated regions at the expense of other regions of interest with fewer data points, as we demonstrate in our experiments.

4.4.2 Theoretical connection to covering problems

Our problem of finding a high-quality sketch $\mathcal{S}$ of size k that minimizes $d_H(\mathcal{X}, \mathcal{S})$ is closely related to the concept of covering numbers in information theory and combinatorics. Informally, internal covering number is defined as the smallest number of equal-sized shapes (e.g., spheres or boxes) centered at individual data points that, together, "cover" all points in a dataset. To relate our covering to the Hausdorff distance, we provide the following lemma:

Lemma 1. *Let $\mathcal{X} \triangleq \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a representation of a single-cell dataset, consisting of M-dimensional measurements $\mathbf{x}_i \in \mathbb{R}^M$ from N individual cells. Let d_H^* be the minimal Hausdorff distance $d_H(\mathcal{X}, \mathcal{S})$ obtained by a sketch $\mathcal{S} \subset \mathcal{X}$ where $|\mathcal{S}| = k$. Then, $d_H^* \triangleq N_{\text{int}}^{-1}(k)$, where $N_{\text{int}}^{-1}(k) \triangleq \min\{r : N_{\text{int}}(\mathcal{X}, r) \leq k\}$.*

Proof. Since d_H bounds the maximum distance of a data point from $\mathcal{S}$, placing a sphere of radius d_H at every point in $\mathcal{S}$ gives a covering of $\mathcal{X}$, which implies $N_{\text{int}}(\mathcal{X}, d_H^*) \leq k$. Thus, $N_{\text{int}}^{-1}(k) \leq d_H^*$. If $N_{\text{int}}^{-1}(k) < d_H^*$, then there exists a cover with k spheres of radius $d' < d_H^*$. Taking the center points of this cover as our sketch $\mathcal{S}'$, we obtain $d_H(\mathcal{X}, S') \leq d_H^*$, a contradiction. Hence, $d_H^* = N_{\text{int}}^{-1}(k)$s. □

Lemma 1 shows that the minimum radius for covering spheres that gives an internal covering number of at most k on a given dataset is in fact equal to the optimal Hausdorff distance achievable by a sketch of size k. An important insight given by this observation is that the problem of finding a high-quality sketch reduces to finding a minimum-cardinality cover of a dataset given a certain radius. In particular, if one were to have access to an oracle that could find the optimal covering of a dataset for

any radius, our problem could be solved by finding the minimum radius that gives the desired number of covering spheres (e.g., via binary search). Unfortunately, finding the minimum-cardinality cover is NP-complete [ANS16], and although algorithms for a variety of simplified settings have been studied [ABD$^+$11, Chv79], none scales to the high-dimensional and large-scale data that we need to handle in single-cell genomics. Given the hardness of the covering problem, we aimed to devise an approximate covering algorithm that readily scales to large-scale single-cell data while maintaining good sketch quality.

4.4.3 Plaid coverings

At the core of our geometric sketching algorithm is a plaid covering, which approximates the geometry of the given single-cell data as a union of equal-sized hypercubes. To enable scalability to large datasets, we restricted our attention to covering the data points with a simple class of covering sets—plaids—whose structure is amenable to fast computation. Formally, we define a length-ℓ plaid cover $\mathcal{C}$ of a dataset $\mathcal{X}$ as a collection of points $\mathbf{c}_1, \ldots, \mathbf{c}_k \in \mathbb{R}^M$ such that two properties hold:

i. Either $c_{ij} = c_{i'j}$ or $|c_{ij} - c_{i'j}| \geq \ell$ for all $i, i' \in [j]$ and $j \in [M]$.

ii. $\mathcal{X} \subset \bigcup_{i=1}^{k} R(\mathbf{c}_i, \ell)$, where $R(\mathbf{c}_i, \ell) \triangleq [c_{i1}, c_{i1} + \ell] \times \cdots \times [c_{im}, c_{im} + \ell]$.

Intuitively, $\mathcal{C}$ represents a collection of M-dimensional square boxes of side length ℓ that cover $\mathcal{X}$ and can be generated by placing a grid (with potentially uneven intervals) over the space and selecting a subset of grid cells. An example plaid cover is illustrated in Figure 4-1. Our greedy algorithm for finding a plaid cover of a given dataset is shown in Algorithm 1.

To see that the plaid cover found by our algorithm uses the smallest number of intervals in each coordinate (although it may not achieve the smallest cardinality overall) consider the following lemma:

Lemma 2. *Algorithm 1 is optimal in each dimension separately.*

Proof. To see this, fix a dimension $d \in [N]$, and consider covering the projection

$$\pi_d(\mathcal{X}) \triangleq \{x_{1d}, x_{2d}, \ldots, x_{Nd}\} \subset \mathbb{R}$$

with a one-dimensional plaid cover of length ℓ. Let $Q \triangleq \{q_1, \ldots, q_k\}$ be any such cover, and let $Y \triangleq \{y_1, \ldots, y_M\}$ denote the cover produced by our algorithm on iteration d. We show that $k \geq m$, i.e., Y has the smallest size of any length-ℓ cover. Assume without loss of generality that $q_1 < q_2 < \cdots < q_k$ and $y_1 < y_2 < \cdots < y_M$. Let z_i denote the ith-smallest element of $\pi_d(\mathcal{X})$. Our algorithm sets $y_1 \triangleq z_1$. We must have $q_1 \leq z_1$, or else z_1 is not covered by Q. Thus, $q_1 \leq y_1$. Proceeding inductively, we see that

$$q_{i+1} \leq \min\{z_i : z_i > q_i + \ell\} \leq \min\{z_i : z_i > y_i + \ell\} = y_{i+1}$$

where the final equality holds because our algorithm defines y_{i+1} exactly this way. Thus, we have $q_i \leq y_i$ for all $i \in \{1, 2, \ldots, \min\{k, M\}\}$. If $|Q| \leq |Y|$, then y_{M-1} and y_M are both greater than all elements in Q. But because Q covers all the points z_i, this implies that y_M covers no points, a contradiction because our algorithm does not construct empty covering sets. Thus, we must have $|Q| \geq |Y|$, and because Q is arbitrary, Y has the smallest possible size. □

Algorithm 1: Greedy Plaid Cover

Data: Data set $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ where $\mathbf{x}_i \in \mathbb{R}^M$, length ℓ
Result: Length-ℓ plaid cover $\mathcal{C}$ of $\mathcal{X}$
$\mathbf{y}_i \leftarrow 0 \in \mathbb{R}^M, \forall i \in [N]$
for $j \in [M]$ **do**
  $z_1, \ldots, z_N \leftarrow \text{Sort}(\{x_{1j}, \ldots, x_{Nj}\})$ `// in ascending order`
  $p \leftarrow 1$
  while $z_p + \ell < z_N$ **do**
    Find smallest $i > p$ where $z_p + \ell < z_i$
    $y_{i'j} \leftarrow z_p, \forall i' \in \{p, \ldots, i-1\}$
    $p \leftarrow i$
  end
  $y_{i'j} \leftarrow z_p, \forall i' \in \{p, \ldots, N\}$
end
return $\{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ `// only unique points are returned`

The main intuition behind our choice of a plaid pattern is that it generalizes grid-based approximation of geometric shapes while maintaining computational efficiency in assigning points to their respective covering box. Note our plaid covering algorithm has time complexity in each dimension of $O(N \log N)$ in general—the main bottleneck being the sorting of each coordinate—and uses $O(N)$ space. In practical scenarios where each coordinate requires only a small constant number of intervals to cover, we achieve $O(N)$ time complexity by taking linear scans to find the next interval without sorting. This is a substantial improvement over other approaches for tackling the covering problem, which typically require $O(N^2)$ time for all pairwise distance calculations. A greedy approach to building a cover could require only $O(kN)$ pairwise distance calculations where k is the number of covering objects [Chv79], yet k is still typically much larger than $\log n$ for our applications in single-cell analysis.

The cardinality of the cover returned by our plaid cover algorithm generally decreases as the length parameter ℓ increases, although pathological cases that deviate from this pattern exist. We empirically confirmed the near-monotonic relationship between number of covering boxes and ℓ on all our single-cell benchmark datasets (Figure A-11). Based on this observation, we perform binary search (with graceful handling of potential exceptions) to find the value of ℓ that approximately produces a desired number of covering boxes. By default, we choose the same number of boxes as the desired sketch size k. A sketch is then constructed by sampling the boxes in a plaid cover and choosing a point at random from each box. The quality of our sketch is given by the following theorem:

Theorem 1. *Given a dataset $\mathcal{X}$ of N points in M dimensions, let $N_{\text{plaid}}(\ell)$ be the number of boxes in the plaid cover returned by our algorithm as a function of length parameter ℓ. Let $N^{-1}_{\text{plaid}}(k) \triangleq \inf\{\ell : N_{\text{plaid}}(\ell) \leq k\}$. Let k be a desired sketch size and assume $k = N_{\text{plaid}}(N^{-1}_{\text{plaid}}(k))$ for simplicity (if not take a nearby k where this holds). Let $S_{\text{plaid}}(k)$ be a sketch of size k obtained by randomly choosing a point from each box in the plaid cover. Let $d^*_H(k) \triangleq \min_{\mathcal{S}:|\mathcal{S}|=k} d_H(\mathcal{X}, \mathcal{S})$. Then, the following holds:*

$$\frac{1}{2} N^{-1}_{\text{plaid}}(2^M k) \leq d^*_H(k) \leq d_H(\mathcal{X}, \mathcal{S}_{\text{plaid}}(k)).$$

Proof. For the first inequality, Let $\mathcal{P} = \{P_1, P_2, \ldots, P_L\}$ be any covering by plaid sets of side length $2d_H^*(k)$, such that all covering sets contain at least one point. We show that $\mathcal{P}$ has cardinality at most $2^M k$. Let $\mathcal{B}$ be a covering of $\mathcal{X}$ by k balls $B_1, B_2, \ldots, B_k$, each with radius $d_H^*(k)$. The definition of d_H^* ensures that such a covering exists. Define

$$I_{\mathcal{P}}(B_i) \triangleq |\{P_j : P_j \cap B_i \neq \varnothing\}|.$$

That is, $I_{\mathcal{P}}(B_i)$ is the number of sets in $\mathcal{P}$ that intersect B_i.

Because $\mathcal{P}$ and $\mathcal{B}$ are both covering sets, each plaid square in $\mathcal{P}$ is intersected by at least one ball in $\mathcal{B}$. Therefore,

$$|\mathcal{P}| \leq \sum_{i=1}^{k} I_{\mathcal{P}}(B_i).$$

On the other hand, we see that $I_{\mathcal{P}}(B_i)$ is bounded above by 2^M, because any ball overlaps at most two plaid intervals in each dimension. Thus,

$$|\mathcal{P}| \leq 2^M k$$

as desired. The second inequality is immediate, because $d_H^*(k)$ is an infimum of Hausdorff distances of all sets of size k with $\mathcal{X}$, and $\mathcal{S}_{\text{plaid}}(k)$ is such a set. □

Theorem 1 gives us a way to bound the optimal Hausdorff distance $d_H^*(k)$ relative to the solution obtained by plaid covering. Although the 2^M factor in the lower bound appears substantial, on real data we expect the exponent to depend on the fractal dimension $d_{\text{frac}} \ll M$ of the data instead, which is typically very small for biological datasets [YDDB15]. We empirically observed that the fractal dimension of single-cell data is around 2 at our working scale. Hence, the performance of a plaid-based sketch makes use of the low intrinsic dimensionality to achieve a highly efficient and empirically well-performing sketching algorithm.

Moreover, we first project the data down to a relatively low-dimensional space (100 dimensions for single-cell data) using a fast randomized PCA [HMT11] before applying our plaid covering algorithm. We note that much work has been done in

obtaining algorithms for computing an approximate PCA of very large datasets with provable bounds on approximation error that are also highly efficient in runtime and memory.

The two parameters needed for our algorithm are the sketch size k and the number of covering boxes $|\mathcal{C}|$. The desired sketch size is chosen depending on the amount of compute resources available and the algorithmic complexity of downstream analyses; smaller sketches omit more cells but will accelerate analysis while preserving much of the transcriptomic heterogeneity. The number of covering boxes converges to uniform sampling as parameter increases; a number of covering boxes less than k may yield a coarser picture of the transcriptomic space, including over-representation of rare cell types, at the cost of an increased Hausdorff distance. We make k a parameter that the user selects and we set $|\mathcal{C}|$ by default to k.

4.5 Empirical performance of geometric sketching

4.5.1 Hausdorff distance benchmarking

We first sought to quantify how well geometric sketching is able to evenly represent the original transcriptomic space by measuring the Hausdorff distance,

$$d_H(\mathcal{X}, \mathcal{S}) \triangleq \max_{\mathbf{x} \in \mathcal{X}} \{\min_{\mathbf{s} \in \mathcal{S}} d(\mathbf{x}, \mathbf{s})\},$$

from the full dataset $\mathcal{X}$ to a geometric sketch $\mathcal{S}$ (Section 4.4.1). Intuitively, a low Hausdorff distance indicates that the points in a sketch are close to all points in the remainder of the dataset within the transcriptomic space, while a high Hausdorff distance indicates that there are some cells in the full dataset that are not well represented within the sketch.

The classical Hausdorff distance (HD), however, is highly sensitive to even a few number of outliers [HKR93]. We therefore use a robust HD measure proposed by Huttenlocher et al. called the partial HD measure, defined as $d_{HK}(\mathcal{X}, \mathcal{S}) \triangleq K^{\text{th}}_{\mathbf{x} \in \mathcal{X}} \{\min_{\mathbf{s} \in \mathcal{S}} d(\mathbf{x}, \mathbf{s})\}$ where $K^{\text{th}}_{\mathbf{x} \in \mathcal{X}}$ denotes the Kth largest value; partial HD re-

quires a parameter $q \triangleq 1 - K/|\mathcal{X}|$ between 0 and 1, inclusive, which is equivalent to classical HD when $q = 0$ [HKR93]. We set $q = 1 \times 10^{-4}$, which obtains a measurement that is very close to the value obtained by classical HD but is robust to the most extreme outliers. We achieved similar results for different values of q (Figure A-12).

We benchmarked geometric sketching against uniform sampling as well as more complex, data-dependent strategies:

- Uniform sampling returns a random sample of the cells, where every cell is given equal probability. We use the random choice function provided by the numpy Python package [Oli07].

- Spatial random sampling (SRS) [RA17] first projects the data points onto the unit hypersphere, then each sample is obtained by uniformly sampling a point on the unit hypersphere and selecting the closest point in the projected dataset according to the cosine distance.

- k-means++ sampling [AV07] randomly chooses an initial sample, then repeatedly samples the next point by giving each point a weight proportional to the minimum distance from previous samples. This procedure continues until the desired number of samples have been obtained. We used the k-means++ implementation from the scikit-learn package [PV11].

Note that, to our knowledge, none of these non-uniform sampling approaches had been previously considered for the problem of downsampling single-cell datasets.

We used four scRNA-seq datasets of varying sizes and complexities to assess our method:

- A 293T/Jurkat mixture with 4,185 cells [ZTB+17] (Table A.2).

- A peripheral blood mononuclear cell (PBMC) dataset with 68,579 cells [ZTB+17] (Table A.3).

- A developing and adolescent mouse central nervous system (CNS) dataset with 465,281 cells [ZHL+18] (Table A.4).

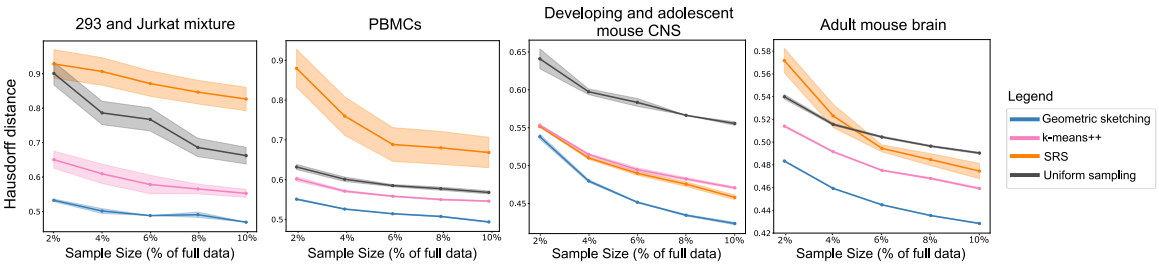

Figure 4-2: Hausdorff distance profiling.

Geometric sketching results in consistently lower Hausdorff distances than other sampling methods across a large number of sketch sizes and datasets. We use a robust Hausdorff distance that is less sensitive to small numbers of outlier observations. Solid lines indicate means and shaded areas indicate standard error across 10 random trials for geometric sketching and uniform sampling.

- An adult mouse brain dataset with 665,858 cells [SMW+18] (Table A.5).

In all cases, we observed that geometric sketching obtains substantially better improvement under the robust Hausdorff distance measure than uniform sampling and the other data-dependent sampling methods, SRS and k-means++ (Figure 4-2). The improvement in Hausdorff distance was consistent across sketch sizes ranging from 2% to 10% of the full dataset, providing quantitative evidence that our algorithm more evenly samples over the geometry of the dataset than do other methods.

4.5.2 Visualizing sketch diversity

We next set out to assess the ability of our geometric sampling approach to improve the low-dimensional visualization of scRNA-seq data, a common exploratory (and often computationally expensive) initial step in single-cell genomic analysis. From our two largest datasets of mouse nervous system, containing 465,281 and 665,858 cells each, we used a 2-dimensional t-SNE [vdMH08] to visualize a sketch containing 2% of the total dataset (sampled without replacement) obtained by geometric sketching.

The results, shown in Figure 4-3, illustrate that the relative representations of cell types in geometric sketches can have striking differences compared to uniformly downsampled datasets. For instance, when obtaining a sketch of 2% of the dataset of adult mouse neurons [SMW+18], clusters of macrophages, endothelial tip cells, and

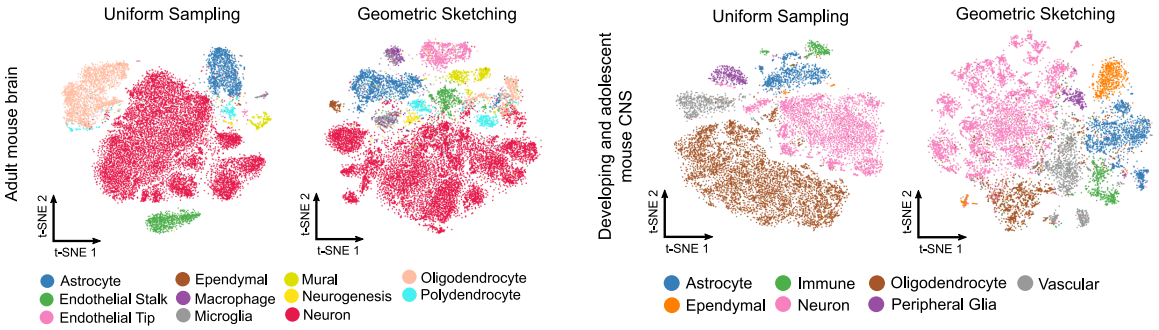

Figure 4-3: Sketch visualizations.

t-SNE visualizations of sketches containing 2% of the cells from the adult mouse brain [SMW+18] and from the developing and adolescent mouse CNS A.5 using uniform random sampling and geometric sketching, with increased representation of rare cell types in the geometric sketch. Visualizations based on other sampling approaches as well as a different visualization method are provided in Figures A-13 and A-14.

mural cells have only 59, 117, and 336 cells, respectively, in the uniform sample out of 1,695, 3,818, and 12,083 cells in the full data, respectively. In contrast, these cell types have 326, 1,022, and 875 cells, respectively, in the geometric sketch of the same size. Although these cell types are rare compared to neurons (428,051 cells in the full dataset), their substantially increased representation in our sketch suggests they inhabit a comparatively large portion of the transcriptomic space. Similarly, on a dataset of 465,281 cells from the developing and adolescent mouse central nervous system (CNS) [ZHL+18], we also observed a more balanced composition of cell types as determined by the original study's authors (Figure 4-3). The rarest cell types are also more consistently represented in a geometric sketch than in sketches obtained by SRS or k-means++ (Figure A-13, Tables A.6 and A.7). We also visualize the data with Uniform Manifold Approximation and Projection (UMAP), an alternative method for computing 2-dimensional visualization embeddings [MH18], with similar results as those produced by our t-SNE experiments (Figure A-14).

We note that our sampling algorithm is completely unsupervised and has no knowledge of the cell type labels, but naturally obtains a balanced composition of cell types by sampling more evenly across the entire transcriptomic space. Indeed, on artificial data in which we controlled the relative volumes and densities of the clusters, geometric sketching samples the clusters proportionally to their relative

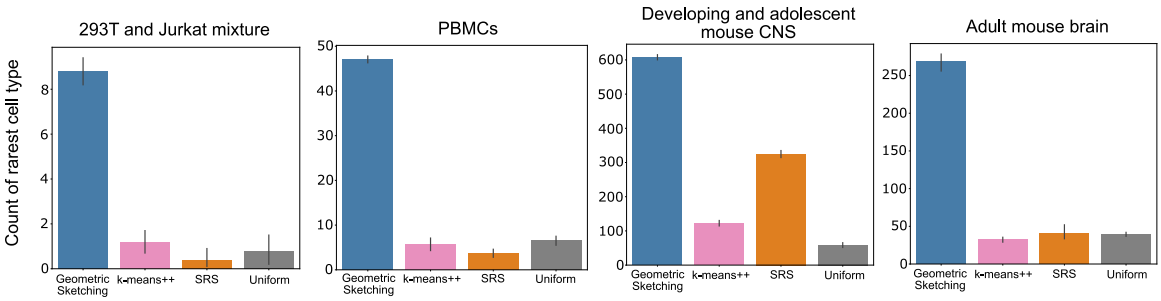

Figure 4-4: Rare cell type preservation.

Geometric sketches preserve rare cell types in the subsampled data. In sketches containing 2% of the total dataset, we counted the number of cells that belong to the rarest cell type in each dataset: 293T cells (0.66% of total cells) in a 293T and Jurkat mixture, dendritic cells (0.38% of total) in a dataset of 68K PBMCs, macrophages (0.25% of total) in a dataset of adult mouse brain cells, and ependymal cells (0.60% of total) in a dataset of developing and adolescent mouse CNS cells. Higher count indicates increased representation of the rare cell type in the sketch. Bar height indicates means and error bars indicate standard error across 10 random trials for geometric sketching and uniform sampling. Comparison of rare cell-type representation over different sketch sizes is shown in Figure A-15B.

volumes rather than their frequencies in the full dataset (Figure A-15A), suggesting that the composition of different cell types in a geometric sketch more closely reflects the transcriptomic variability of individual clusters rather than their frequency in the overall population. Our visualizations therefore reflect a geometric "map" of the transcriptomic variability within a dataset, allowing researchers to more easily gain insight into rarer transcriptomic states.

4.5.3 Rare cell type preservation

As suggested by the above results, one of the key advantages of our algorithm is that it naturally increases the representation of rare cell types with sufficient transcriptomic heterogeneity in the subsampled data. Using the four datasets mentioned above, which include cell type labels provided by the original study authors, we evaluated the ability of our method to preserve the rarest cell type within each dataset. In particular, we focused on 28 293T cells (0.66% of the total number of cells in the dataset) in the 293T/Jurkat mixture, 262 dendritic cells (0.38%) in the PBMC dataset, 1695 macrophages (0.25%) among the adult mouse brain cells, and 2,777 ependymal cells (0.60%) among the mouse CNS cells.

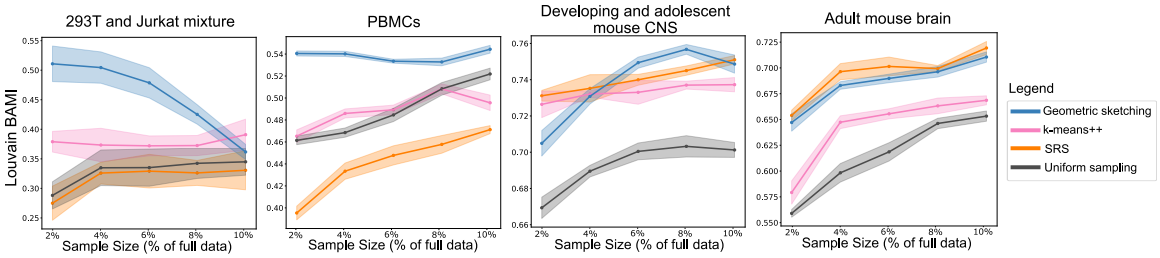

Figure 4-5: Cluster preservation.

Louvain clustering was applied to a subsample of the dataset, cluster labels were transferred to the full dataset using a k-nearest-neighbor classifier fit to the sketch, and the BAMI was measured between the unsupervised cluster labels and the labels corresponding to biological clusters provided by each previous study. Higher score indicates greater agreement between unsupervised clustering and biological cell-type labels. Solid lines indicate means and shaded areas indicate standard error across 10 random trials for geometric sketching and uniform sampling.

In all datasets, the rare cell types are substantially more represented in the sketch obtained by our algorithm compared to other sampling techniques (Figure 4-4). For example, a sketch that is 2% the size of the 665,858 mouse brain cells contains an average of 281 macrophages compared to only 31 cells from uniform sampling. Geometric sketching is able to better preserve rare cell types because the extent of transcriptomic variation among rare cells is similar to that of common cells. To this end, we used the differential entropy of a multivariate Gaussian fit to each cell type as a proxy to its transcriptomic diversity (Tables A.2, A.3, A.4, and A.5). We also note that, within the geometric sketch, almost all of the rare cell types in each dataset have increased representation compared to the full data, where the representation of rare cell types gradually converges to that of uniform sampling as the sketch size increases (Figure A-15B).

4.5.4 Preserving all cell types

Since the samples produced by our algorithm consist of a more balanced composition of cell types, including rare cell types, we also reasoned that clustering analyses should be able to better distinguish these cell types within a geometric sketch compared to uniform downsampling. To assess this capability, we first clustered the sketches

using the standard graph-based Louvain clustering algorithm [BGLL08]. Then, we transferred cluster labels to the rest of the dataset via k-nearest-neighbor classification and assessed the agreement between our unsupervised cluster labels and the biological cell type labels provided by the original studies. We quantified the clustering accuracy via balanced adjusted mutual information (BAMI), our proposed metric for evaluating clustering quality when the ground truth clusters are highly imbalanced, which is often the case for scRNA-seq datasets. BAMI balances the terms in adjusted mutual information [VEB10] to equally weight each of the ground truth clusters, preventing rare cell types from having only negligible contribution to the performance metric. We also provide results for adjusted mutual information, without our balancing technique, which are largely consistent with our comparisons based on BAMI (Figure A-15C).

On a variety of real scRNA-seq datasets, our algorithm's sketches recapitulate the biological cell types consistently better than uniform sampling (Figure 4-5). Although two other data-dependent sampling methods, SRS and k-means++, achieve performance comparable to our method in a few cases, only geometric sketching obtains competitive performance across all datasets, suggesting that our method is reasonably robust to different experimental settings. Notably, because our sketches are drawn without replacement, clustering scores can become closer to those of uniform samples as the size of the sketch increases; this may explain the diminishing performance of our method with increasing sketch size on the mixture of 293T cells and Jurkat cells (Figure 4-5). Still, we note our substantial advantage even on this dataset using very small sketches that select as low as 2% of the full dataset. Moreover, the overall improvement in clustering consistency could become more pronounced as more fine-grain clusters become available as ground truth in light of the enhanced representation of rare transcriptomic states within geometric sketches.

4.5.5 Improved scalability

Not only does geometric sketching lead to more informative sketches of the single-cell data, it is also dramatically faster than other non-uniform sampling methods, which is imperative since researchers stand to gain the most from sketches of very large datasets.

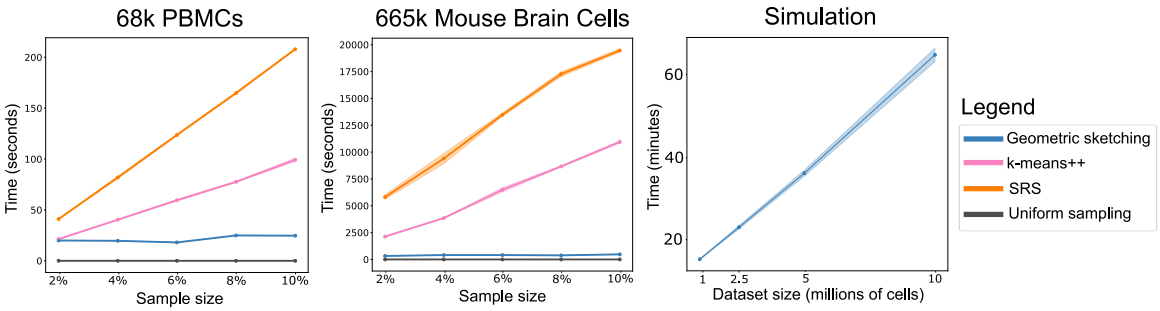

Figure 4-6: Runtime benchmarks.

Geometric sketching is substantially more efficient than other data-dependent subsampling approaches, SRS and k-means++. Solid lines indicate means and shaded areas indicate standard error across 10 random trials for geometric sketching and uniform sampling and 4 random trials for k-means++ and SRS (due to long run times). Geometric sketching has a practical runtime of around 67 min when sampling 20,000 cells from a simulated dataset with 10 million cells, which was obtained by resampling from a dataset of mouse CNS cells [ZHL+18].

Geometric sketching has an asymptotic runtime that is close to linear in the size of the dataset and, when benchmarked on real datasets, is more than an order of magnitude faster than non-uniform methods and has a negligible dependence on the number of samples specified by the user, unlike k-means++ and SRS (Figure 4-6). On our largest dataset of 665,858 cells, our sampling algorithm takes an average of around 5 minutes (Figure 4-6); in contrast, k-means++ takes 3 hours and spatial random sampling (SRS) takes 5.5 hours when subsampling 10% of the cells. On a simulated benchmark dataset of 10 million data points, geometric sketching subsamples 20,000 cells after an average time of 67 minutes, demonstrating practical scalability to datasets with hundreds of millions of cells (Figure 4-6).

Although uniform sampling is trivially the most efficient technique since it does not consider any properties of the underlying dataset, our algorithm is both efficient and produces high quality samples that more accurately represent the underlying transcriptomic space as we demonstrated above. Notably, our runtime comparison does not include the standard preprocessing step of (randomized) principal component analysis (PCA), which we uniformly applied to all methods and whose runtime as well as scalability are comparable to our geometric sketching step (Figure A-16A).

4.5.6 Accelerating data integration

Because of its efficiency, geometric sketching can also accelerate other downstream algorithms for scRNA-seq analysis. One such problem involves integration of multiple scRNA-seq datasets across different batches or conditions [BHS+18, HLMM18, HBB19, KMF+19]. Here, we consider an approach to accelerating scRNA-seq data integration by applying the integration algorithm only to geometric sketches instead of the full datasets.

We assume an integration function that takes in a list of datasets and returns modifications to the datasets that removes differences to due batch effect etc. Let $\mathbf{X} \in \mathbb{R}^{N \times M}$ denote one of the datasets, $\mathbf{X}_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}| \times M}$ denote the subset of $\mathbf{X}$ obtained by geometric sketching, and $\mathbf{X}'_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}| \times M}$ denote the modified version of $\mathbf{X}_{\mathcal{S}}$ returned by the integration function. Our goal is to apply a transformation to $\mathbf{X}$ that puts it into the same integrated space as $\mathbf{X}'_{\mathcal{S}}$. At a high level, we use a nearest-neighbors-based method to compute alignment vectors from $\mathbf{X}$ to $\mathbf{X}_{\mathcal{S}}$, we use Gaussian smoothing to combine these alignment vectors into translation vectors, and then we apply the translation to $\mathbf{X}$ to obtain an "integrated" full dataset $\mathbf{X}'$.

Formally, for each cell in $\mathbf{X}_{\mathcal{S}}$, we find its k nearest neighbors in $\mathbf{X}$ and we denote the set of all matches between a cell in $\mathbf{X}_{\mathcal{S}}$ and $\mathbf{X}$ as $\mathcal{M}$ where $|\mathcal{M}| = k|\mathbf{X}_{\mathcal{S}}|$. Now we define the alignment vectors as the rows of the matrix $\mathbf{X}^{(\text{match})} - \mathbf{X}_{\mathcal{S}}^{(\text{match})}$ where the rows of $\mathbf{X}^{(\text{match})}, \mathbf{X}_{\mathcal{S}}^{(\text{match})} \in \mathbb{R}^{|\mathcal{M}| \times M}$ correspond to the pairs of matching cells in $\mathcal{M}$. We want to combine these alignment vectors to obtain our translation vectors, which we do using Gaussian smoothing. We compute weights via a Gaussian kernel as

$$[\mathbf{\Gamma}]_{a,b} \triangleq \exp\left\{-\frac{\sigma}{2}\left\|[\mathbf{X}]_{a,:} - [\mathbf{X}^{(\text{match})}]_{b,:}\right\|_2^2\right\}$$

where $\mathbf{\Gamma} \in \mathbb{R}^{N \times |\mathcal{M}|}$ and $[\cdot]_{a,b}$ denotes the value in the ath row and bth column of a matrix and $[\cdot]_{a,:}$ denotes the ath row of a matrix. Finally, we construct the translation

vectors as an average of the alignment vectors with Gaussian-smoothed weights, where

$$\mathbf{v}_a \triangleq \frac{[\mathbf{\Gamma}]_{a,:}\left(\mathbf{X}^{(\text{match})} - \mathbf{X}_{\mathcal{S}}^{(\text{match})}\right)}{\sum_{b\in[|\mathcal{M}|]}[\mathbf{\Gamma}]_{a,b}}$$

and we translate

$$[\mathbf{X}']_{a,:} \triangleq [\mathbf{X}]_{a,:} + \mathbf{v}_a$$

for all $a \in [N]$ where $[N]$ denotes the set of all natural numbers up to N. We repeat this for all datasets integrated by the "black-box" integration function; in our study, we used the Scanorama [HBB19] and Harmony [KMF$^+$19] algorithms for integration.

We use geometric sketches of size 4,000 (around 1% of the total data) and parameters $k = 3$ and $\sigma = 15$. We used Harmony version 0.0.0.9000 and Scanorama version 1.0. For all methods, we measured the runtime required for integration and translation, not including the initial PCA step for computing low dimensional embeddings (100 PCs). We quantify dataset mixing by clustering the integrated embeddings using k-means, varying the number of clusters, and computing the average negative Shannon entropy normalized to a maximum value of 1 on the dataset labels averaged across all clusters, an approach taken by recent work [PPY$^+$19].

Since the integration step is more computationally intensive than the latter interpolation step, our geometric sketch-based integration offers a speedup that becomes especially dramatic when integrating large numbers of cells. Moreover, because geometric sketching better preserves rare transcriptomic states, as demonstrated above, rare cell types are also more likely to be integrated during the procedure compared to using sketches from other sampling approaches.

We applied geometric sketch-based acceleration to two existing algorithms, Scanorama [HBB19] and Harmony [KMF$^+$19], for scRNA-seq data integration (Figure 4-7). However, we note that our acceleration procedure is agnostic to the underlying integration method and can easily interface with similar algorithms [BHS$^+$18, HLMM18]. We benchmarked the runtime improvement using geometric sketching on a dataset of 534,253 human immune cells from two different tissues (umbilical cord blood and adult bone marrow). On this data, Scanorama and Harmony require 2.1 and 1.9 hours

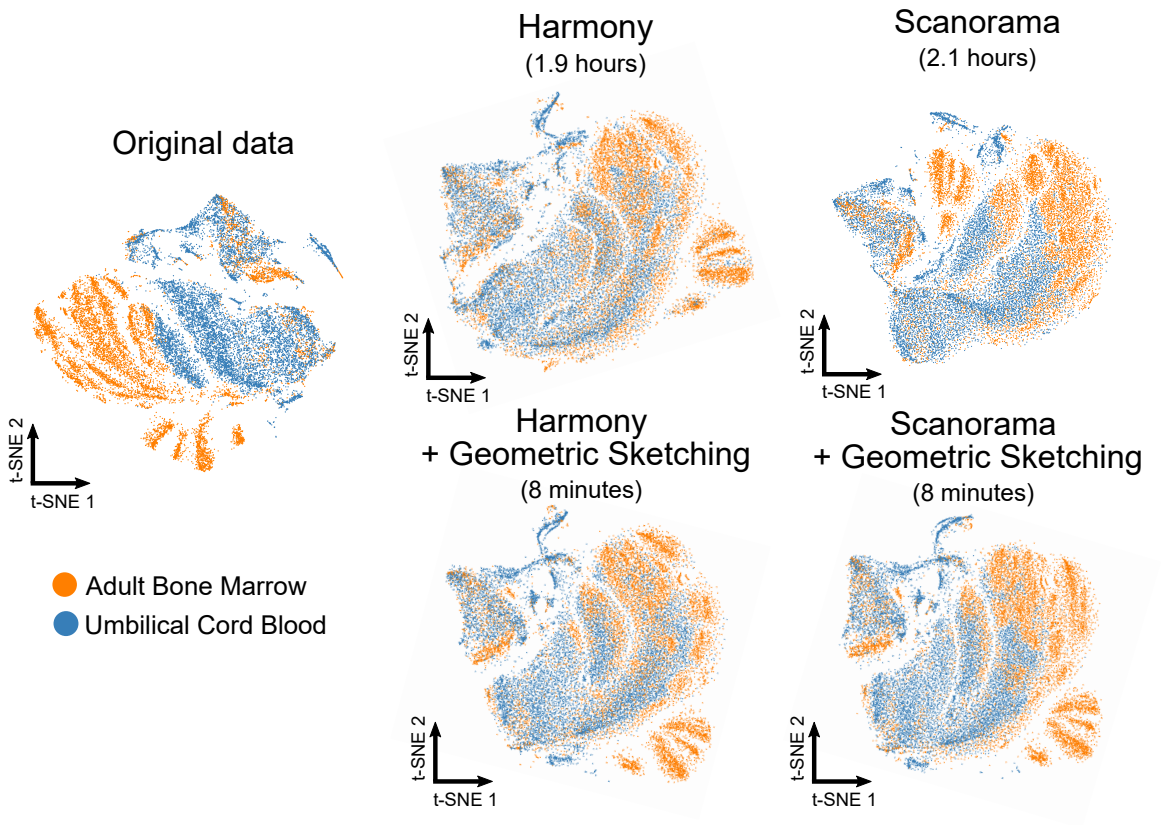

Figure 4-7: Faster data integration with sketching.

Geometric sketching can help accelerate existing tools for scRNA-seq data integration. We use two existing algorithms for scRNA-seq integration, namely Harmony [KMF+19] and Scanorama [HBB19], but note that our approach works for other integrative algorithms as well. Learning alignment vectors among geometric sketches, which are then used to transform the full datasets to remove tissue-specific differences, decreases integration time of 534,253 human immune cells from hours to minutes while achieving comparable integration quality (Figure A-16B).

of computation, respectively, to obtain integrations that remove tissue-specific differences. In contrast, the integration procedure with geometric sketching (which includes finding the geometric sketches, integrating the sketches, and then transforming the full datasets based on the sketches) requires just 8 minutes of computation with either Scanorama or Harmony. Moreover, using geometric sketching-based acceleration has integration performance comparable to the full integration (Figure 4-7) and better than sketch-based integration using other sampling strategies (Figure A-16B), providing yet another example of how geometric sketching can be used to accelerate other algorithms for large-scale scRNA-seq analysis.

4.6 Application note: Discovering rare inflammatory immune states

Because geometric sketching of large datasets highlights rare transcriptomic states, certain subpopulations of cells that are difficult to identify when analyzing the full dataset may become discoverable within a geometric sketch. This property of geometric sketching is particularly relevant in understanding host-pathogen interactions and immunity, since many immune responses to infection are driven by only a small subset of cells, particularly in early or latent infection.

To illustrate this concept, we analyzed a dataset of 254,941 immune cells from human umbilical cord blood. We computed a geometric sketch of 20,000 cells and clustered the sketch via the Louvain community detection algorithm. We were particularly interested in putative macrophage clusters with elevated expression of macrophage-specific marker genes, including *CD14* and *CD68* [KMT+05]. Among these clusters, we found a comparatively rare cluster of macrophages defined by the marker genes *CD74*, *HLA-DRA*, *B2M*, and *JUNB* (AUROC > 0.90) (Figure 4-8).

We hypothesized that this cluster corresponds to inflammatory macrophages since each of its marker genes has been implicated in macrophage activation in response to inflammatory stimuli: *CD74* encodes the receptor for macrophage migration inhibitory

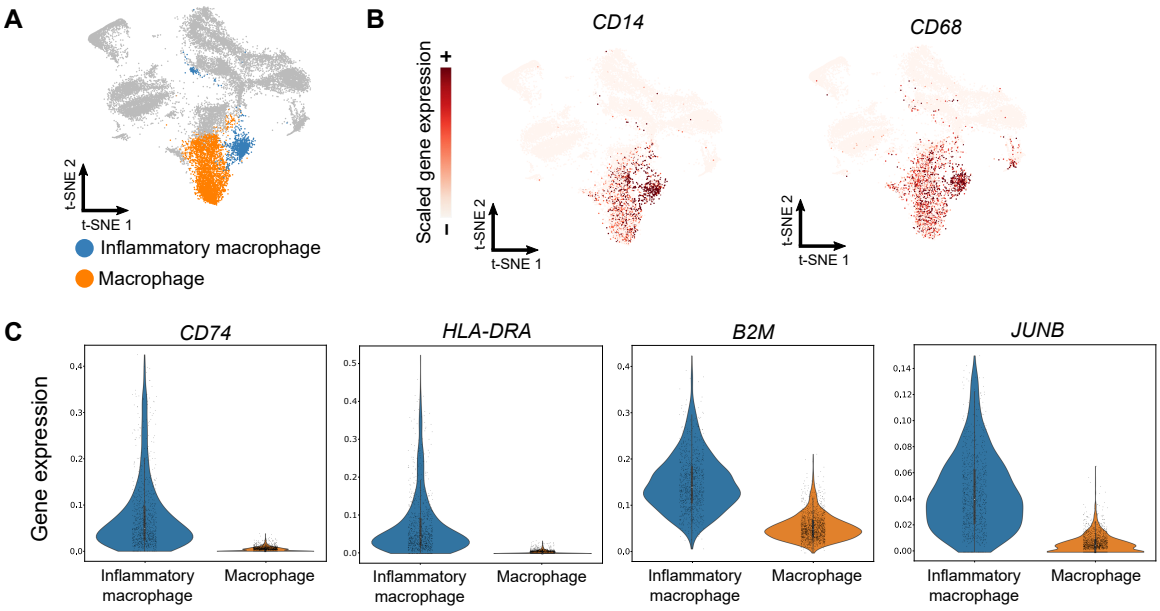

Figure 4-8: Sketching identifies activated macrophage states.

(**A**–**C**) A geometric sketch of 20,000 cells was obtained from a full dataset of 254,941 cells from human umbilical cord blood. Analysis of clusters obtained by the Louvain community detection algorithm reveals multiple clusters of macrophages (**A**), defined by *CD14* and *CD68* marker gene expression (**B**). A rare subpopulation of these macrophages is in turn defined by inflammatory marker gene expression (*CD74*, *HLA-DRA*, *B2M*, and *JUNB*) (**C**), providing insight into an important but comparatively rarer immunological process. Violin plots are generated by fitting a kernel density estimate to the gene expression values across cells. Black dots indicate gene expression values of individual cells with random jitter along the horizontal axis.

factor (MIF) [LMF+03], a pro-inflammatory signal [MLB06]; *HLA-DR* and *B2M* are hallmarks of the antigen presentation in activated macrophages (Section 2.1.4); and *JUNB* has been implicated as a key transcriptional modulator of macrophage activation [FBP+15] and is upregulated by MIF [CR03]. We did not observe major differences in the number of unique genes between this rare cluster and the rest of the macrophages, so these differences in gene expression are most likely not an artifact of variable data sparsity or dropout.

We sought further confirmation of this rare expression signature in macrophages by conducting a separate scRNA-seq study of an in vitro model of macrophage inflammation in which human $CD14^+$ monocytes were polarized with GM-CSF to induce an inflammatory response. We compared this data to a similar scRNA-seq dataset of macrophages but with M-CSF stimulation [HBB19] to induce an anti-inflammatory polarization. Expression of all four marker genes we identified (*CD74*, *HLA-DRA*, *B2M*, and *JUNB*) was significantly elevated in GM-CSF-derived ($n = 354$ cells) macrophages compared to the M-CSF-derived ($n = 1{,}107$ cells) macrophages (one-sided Welch's t-test $P = 4 \times 10^{-34}$ for *CD74*, $P = 1 \times 10^{-29}$ for *HLA-DRA*, $P = 3 \times 10^{-46}$ for *B2M*, and $P = 1 \times 10^{-13}$ for *JUNB*), increasing our confidence in these marker genes as a signature of inflammation.

When we applied the same clustering procedure to either the full dataset or a uniform subsample, the clustering algorithm did not assign a separate cluster to inflammatory macrophages but rather placed all macrophages into a single cluster, likely because of the relative scarcity of this cell type compared to the large cluster of inactive macrophages.

These results are important since macrophages activated during inflammation play a crucial role in the immune system: macrophages ingest or "phagocytose" foreign invaders, present parts of those invaders on the cell surface so that other components of the immune system (like T cells) can recognize those foreign parts in the future, and stimulate other immune cells to respond to infection via signaling molecules called "cytokines" [MW16]. While researchers do understand macrophage activation in broad terms, inflammatory macrophages have been shown to be highly heterogeneous in

their response to infection and much is still unknown about this heterogeneity or how it is controlled. Thus, scRNA-seq coupled with tools for like geometric sketching can play a role in revealing rare macrophage types. More in-depth study of macrophage subpopulations, in part through intelligent computational analyses that highlight cellular diversity, will help reveal insight into inflammation and ways to modulate inflammatory processes in response to disease.

Chapter 5

Understanding Disease III: Synthesis

Tena hi bhaṇe jaccandhānaṁ hatthiṁ dassehī' ti.
(Very well then, I say, show the blind people an elephant.)
—*Udāna* 6.4 (c. 1st century B.C.)

The past two chapters have described algorithmic contributions to the analysis of modern, high-dimensional biological data, in particular single-cell transcriptomics. Biological systems, however, are more complex than what gene expression alone can describe. Increasingly, high-throughput assays measure multiple data *modalities* within the same experiment [SS19]. For example, a multimodal assay of bacterial infection could not only measure the human mRNA within a single cell but also the intracellular bacterial mRNA, the expression of human protein cell surface markers, and the time elapsed since initial infection; in this experiment, human mRNA, bacterial mRNA, human surface proteins, and time can each be thought of as a separate data modality.

Two fundamental (and related) questions when analyzing multimodal data are:

i. How do we weigh the contribution of different modalities when performing data analysis? While each modality could be given equal weight, in some instances a researcher may have good reason to trust some modalities over others (e.g.,

due to different levels of noise), or a researcher may be more interested in one modality over the others.

ii. How do we reconcile different information across modalities? For example, a transcriptomic modality might indicate that two cells are similar, while a proteomic modality might indicate that two cells are different.

Fundamentally, these questions underlie a fundamental problem of synthesizing information from multiple data streams into a coherent whole. We therefore refer to this problem as *multimodal synthesis*.

This chapter develops a unifying conceptual approach, which we call Schema[1] [SHNB20], to multimodal synthesis based on ideas from metric learning and implemented with a quadratic programming formulation. Schema is designed to be highly general—it works for any modality on which one can define a distance metric and it extends to arbitrarily many modalities. Since our method is general, we highlight a breadth of applications, with a special attention to multimodal single-cell analysis. This chapter ends with an application note describing how multimodal data enables new insight into the variability in T-cell recognition of foreign antigens, which is a fundamental immunological problem. More broadly, because the study of host-pathogen interactions is inherently multimodal, we anticipate these techniques will serve as a powerful analytic tool for better understanding the immune system and infectious disease.

5.1 Glossary

- *Data modality*: A distinct type of (potentially multidimensional) data; for example, RNA expression, protein expression, and time are different modalities.

- *Multimodal synthesis*: Data analysis that weighs and reconciles the information contribution of each modality.

[1]Software available at `http://schema.csail.mit.edu` and at `https://github.com/rs239/schema`

- *Quadratic programming*: A mathematical optimization problem in which a quadratic function involving potentially many variables is optimized with respect to linear constraints on the variables.

- *Metric learning*: Using a set of training examples indicating data points that are similar or different, a metric learning algorithm learns general patterns of data similarity or difference.

5.2 Preliminaries

(Preliminaries related to scRNA-seq technologies and standard data analysis, described in Section 3.2, may also be helpful to read alongside the preliminary information below.)

5.2.1 Multimodal biological assays

While a detailed primer on modern, multimodal biological assays is left to reference [SS19], introductory detailis on a few important modalities will be useful for immediately appreciating the work in this chapter. In the past two chapters, we have been concerned primarily with a high-dimensional transcriptomic readout of a given cell, but there are many aspects of biological systems beyond mRNA.

Modern techniques also enable high-dimensional readouts of different aspects of the central dogma, in particular, proteomics and epigenomics. Many of these techniques leverage the same general nucleic acid barcoding strategy described in our discussion of scRNA-seq (Section 3.2). For example, surface protein expression can be measured with a collection of antibodies, where each antibody is designed to bind a specific surface protein and where a unique DNA barcode attached to the antibody indicates which protein it binds [SHS$^+$17]. A popular epigenomic readout is "chromatin accessibility," which identifies regions of the genome that are "open" and those that are "closed"; genes in open regions are accessible to transcription factors and are more likely to be expressed, while closed regions are generally not expressed. A popular high-throughput

method called single-cell ATAC-seq, or assay for transposase-accessible chromatin using sequencing, also uses barcodes to tag DNA fragments that come from open regions, with a unique barcode for each single cell [CCR+18]. Multimodal data would then look like, for example, simultaneous measurements of single-cell transcriptomes, proteomes, and accessibility-defined epigenomes.

Other data modalities of note are space and time. A common strategy that enables spatial single-cell analysis is to associate a barcode with its two-dimensional (or, increasingly, three-dimensional) coordinates; e.g., in spatial transcriptomics, barcodes define both the cellular identity of an mRNA molecule and its location in space [RSG+19]. Time often corresponds to the physical time elapsed relative to a start time, e.g., hours since stimulation with an inflammatory molecule. In some instances, some studies also try to assign a continuous order to a set of samples based on some property (e.g., based on increasing average expression of a set of genes) and thereby order the cells according to "pseudotime" [TCG+14]; however, because pseudotime is often inferred based on the information from a different modality, care must be taken when reasoning about pseudotime in a multimodal setting.

5.2.2 Metric learning

(A primer on distance metrics, a fundamental concept in metric learning and in this chapter, is provided in Section 3.2.3.)

The goal of metric learning is to learn an appropriate distance metric for a problem of interest using information from a set of training examples. Through weak supervision, these training examples provide information on how the distance metric behaves. Broadly, consider a distance metric d defined according to a set of parameters Θ. Most metric learning approaches uses a set of paired or tripleted training points of the form

$$\mathcal{S} \triangleq \{(x, y) : x \text{ and } y \text{ should be similar}\},$$

$$\mathcal{D} \triangleq \{(x, y) : x \text{ and } y \text{ should be dissimilar}\}, \quad \text{and}$$

$$\mathcal{R} \triangleq \{(x, y, z) : x \text{ should be more similar to } y \text{ than to } z\}.$$

A metric learning algorithm can then be formulated as an optimization algorithm that finds

$$\Theta^* \triangleq \arg\min_{\Theta} \mathcal{L}(\Theta; \mathcal{S}, \mathcal{D}, \mathcal{R})$$

where $\mathcal{L}$ is a loss function that penalizes values of Θ that violate the constraints in $\mathcal{S}$, $\mathcal{D}$, and $\mathcal{R}$. Intuitively, rather than use a traditional distance metric like Euclidean distance directly on a dataset, an algorithm learns a distance metric with respect to a set of training examples. Metric learning approaches are reviewed in further depth in reference [BHS14].

5.3 The rise of multimodal biology

5.3.1 Challenges and opportunities

High-throughput assays can now measure diverse cellular properties, including transcriptomic [ZTB+17, HPN+20], epigenomic [CCR+18], proteomic [SHS+17], functional [Gen19], and spatial [RSG+19] data modalities. Excitingly, single-cell experiments increasingly profile multiple modalities simultaneously within the same experiment [SS19], 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, we do not need to integrate modalities across different populations of cells, which was the problem we focused on in Chapter 3 [HBB19].

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 transcriptomically but are different epigenetically? While some multimodal analysis accommodates only specific modalities (e.g., special tools for spatial transcriptomics), is aimed at just gene-set estimation [AAB+20], or is limited only to a pair of modalities [DY20], a *general* multimodal analysis paradigm that applies and extends to any data modality holds the promise

of unifying these observations to inform biological discovery. Importantly, given the rapid biotechnological progress that continues to enable novel measurement modalities and easier simultaneous multimodal profiling, such a paradigm should scale to massive single-cell datasets and be robust to noise and sparsity in the data. Furthermore, the ability to synthesize more than just two modalities provides deeper insights and more robust (accurate) inferences, as we demonstrate.

5.3.2 A key insight from metric learning

Before the advent of multimodal single-cell experiments, computational analysis has focused on variation within a single modality. A critical problem brought about by the advent of multimodal single-cell experiments is how to reason about information across modalities in a mutually consistent way. The key insight underlying our approach to multimodal synthesis, which we call Schema, is that 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 [BHS14]. Given a reference modality, Schema transforms this modality such that its Euclidean distances agree with a set of supplementary distance metrics from the other modalities, while also limiting the distortion from the original reference modality. Analyses on the transformed data will thus incorporate information from all modalities (Figure 5-1).

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 (i.e., reference), with RNA-seq being a frequent choice. The coordinates of points in the primary modality are then transformed using information from secondary modalities. Importantly, the

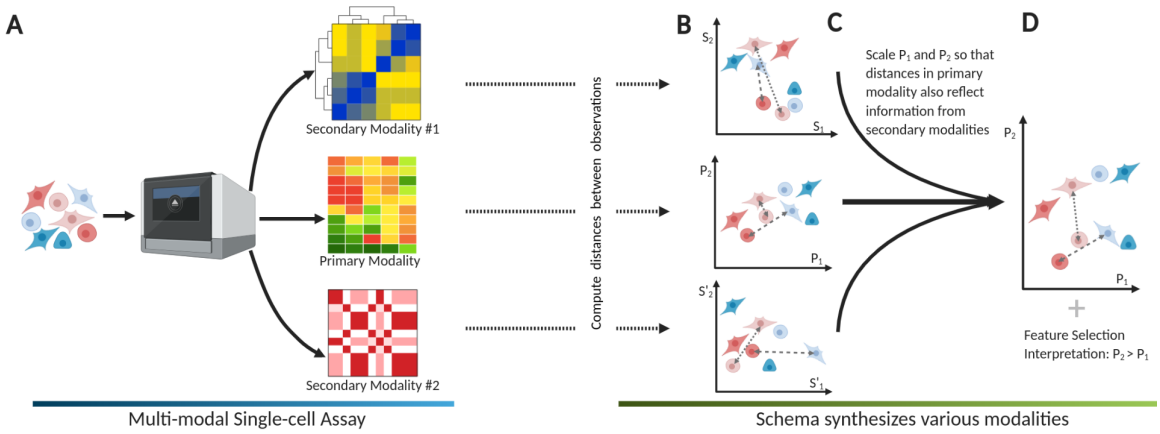

Figure 5-1: Schema overview and intuitions.

(**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 lines indicate distances between the same pairs of observations. (**C**) Schema transforms the primary-modality space by scaling each dimension so that the Euclidean distances in the transformed space have a higher correlation with corresponding distances in the secondary modalities; arbitrary distance metrics are allowed for the latter. Importantly, the transformation is guaranteed to limit the distortion of the original space. (**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.

transformation is constrained to limit the distortion to the primary modality below a researcher-specified threshold. This acts as 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 (a common approach in statistics for inferring information from cross-covariance matrices), is prone to overfitting to sample-specific noise, as we show in our results.

To see how Schema's transformation synthesizes modalities, consider the case where the primary dataset is gene expression data. While the points close in Euclidean space are likely to be biologically similar cells with shared expression profiles, longer Euclidean distances are less informative. 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 the following section. 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). In this study, we demonstrate this interpretability in our applications of Schema.

5.4 Schema: Algorithm details

5.4.1 Problem formulation

Suppose we have N observations across r datasets $\mathbf{X}_j$, $1 \leq j \leq r$, where $\mathbf{X}_j = \{\mathbf{x}_i^{(j)} : 1 \leq i \leq N\}$ contains data (categorical or continuous) for each observation. We will refer to $\mathbf{X}_1$ as the *primary* dataset and the rest as secondary. Each dataset's dimensionality and domain may vary. In particular, we assume $\mathbf{X}_1$ is k-dimensional. Each dataset $\mathbf{X}_j$ should also have some notion of distance between observations attached to it, which we will denote ρ_j, so $\rho_j(\mathbf{x}_n^{(j)}, \mathbf{x}_m^{(j)})$ is the distance between the nth and mth observations in $\mathbf{X}_j$. Since our entire framework below deals in *squared* distances, for notational convenience we will let ρ_j be the *squared* distances between points in D_j; also, we drop the superscript in $\mathbf{x}_j^{(1)}$ when referring to the primary dataset D_1 and its data.

The goal is to find a transformation Ω such that $\Omega(\mathbf{X})$ generates a dataset $\mathbf{X}^*$ such that the Euclidean metric ρ^* on $\mathbf{X}^*$ "mediates" between the various metrics ρ_j, each informed by its respective modality. Note that none of the ρ_js need to be Euclidean. The above setup is quite general, and we now specify the form of the transformation Ω and the criteria for balancing information from the various metrics. Here, we constrain Ω to a *scaling transform*. That is, $\Omega(\mathbf{X}) \triangleq \mathbf{X} \cdot \mathrm{diag}(\boldsymbol{\omega})$ for some $\boldsymbol{\omega} \in \mathbb{R}^k$, where $\mathrm{diag}(\boldsymbol{\omega})$ denotes a $k \times k$ diagonal matrix with $\boldsymbol{\omega}$ as its diagonal entries. The scaling transform $\boldsymbol{\omega}$ acts as a feature-weighting mechanism: it chooses the *features* of $\mathbf{X}_1$ that align the datasets best (i.e., ω_i being large means that the ith coordinate of $\mathbf{X}_1$ is important). 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.

Here, our approach to integration between the metrics ρ_j is to learn a metric ρ^* that aligns well with all of them. 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:

$$\text{Corr}(\rho^*, \rho_j) = \frac{\text{Cov}\left[\rho^*, \rho_j\right]}{\left(\text{Var}\left[\rho^*\right]\text{Var}\left[\rho_j\right]\right)^{1/2}} \tag{5.1}$$

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. We also allow hard constraints, whereby the correlation between the transformed data and some $\mathbf{X}_j$ has to be at least some value. Our goal is thus to find

$$\begin{aligned} &\max_{\boldsymbol{\omega}\in\mathbb{R}^k} \sum_{j=1}^{r} \gamma_j \text{Corr}(\rho^*(\boldsymbol{\omega}), \rho_j) \\ \text{subject to} \qquad &\text{Corr}(\rho^*(\boldsymbol{\omega}), \rho_j) \geq \phi_j \text{ for } j \in \{1, \ldots, r\} \end{aligned} \tag{5.2}$$

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 $\boldsymbol{\omega}$ and is determined entirely by the solution to Equation (5.2). In the rest of our discussion, we will primarily refer to $\boldsymbol{\omega}$, rather than ρ^*. In order to make this optimization feasible, we use the machinery of *quadratic programming*.

5.4.2 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:

$$\min_{\mathbf{v}\in\mathbb{R}^s} \mathbf{v}^{\mathrm{T}}\mathbf{Q}\mathbf{v} + \mathbf{q}^{\mathrm{T}}\mathbf{v} \tag{5.3}$$

subject to

$$\mathbf{G}\mathbf{v} \preceq \mathbf{h}$$

$$\mathbf{A}\mathbf{v} = \mathbf{b}$$

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

To put our optimization in Equation (5.2) in a QP formulation, we expand the covariance and variance terms in Equation (5.1), and show that the covariance is *linear* in the transformation and variance is *quadratic*, i.e., we show

$$\mathrm{Cov}(\mathbf{w}, \rho_\ell) = \left(\tfrac{1}{|\mathcal{P}|}\mathbf{a}_\ell - \tfrac{1}{|\mathcal{P}|^2}\mathbf{b}_\ell\right)\mathbf{w} \quad \text{and} \quad \mathrm{Var}(\mathbf{w}) = \mathbf{w}^{\mathrm{T}}\left(\tfrac{1}{|\mathcal{P}|}\mathbf{S} - \tfrac{1}{|\mathcal{P}|^2}\mathbf{T}\right)\mathbf{w} \tag{5.4}$$

where $\mathbf{w} \in \mathbb{R}^k$ such that $w_i \triangleq \omega_i^2$, $\mathbf{a}_\ell$ and $\mathbf{b}_\ell$ are k-dimensional vectors that depend only on $\mathbf{X}_\ell$, $\mathbf{S}$ and $\mathbf{T}$ are $N \times k$ matrices that depend only on $\mathbf{X}_1$, and $\mathcal{P}$ is the set of pairs of observations. It is also not hard to show that $(|\mathcal{P}|^{-1}\mathbf{S} - |\mathcal{P}|^{-2}\mathbf{T})$ is psd, as required. For details of the derivation, see Section B.1.

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 reframed as 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:

$$\max_{\mathbf{w}\in\mathbb{R}^k} \sum_{j=1}^{r} \gamma_r \mathrm{Cov}(\mathbf{w}, \rho_j^2) - \alpha \mathrm{Var}(\rho^*) - \lambda\|\mathbf{w} - \mathbf{1}\|^2 \tag{5.5}$$

subject to

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

$$\mathbf{w} \succeq \mathbf{0}$$

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 $\mathbf{w}$, 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 $\mathbf{w}$ alone (i.e. incorporate $-\lambda\|\mathbf{w}\|^2$) which would encourage $\mathbf{w}$ to be small; we have found that empirically the choices yield similar results.

This program can be solved by standard QP solvers (see Section B.1 for the full details of how to put the above program in canonical form for a solver), and the solution $\mathbf{w}^*$ can be used to transform unseen input data, using $\boldsymbol{\omega}^* \in \mathbb{R}^k$, where $\omega_i^* = \sqrt{w_i^*}$.

5.4.3 Implementation details

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 $\mathbf{X}^*$ and each of the $\mathbf{X}_i$; and also the ratio of the largest value of $\mathbf{w}$ to its average value, Formally, the user can specify s_i such that

$$\text{Corr}(\rho^*, \rho_i) \geq s_i \text{ for } 1 \leq i \leq r$$

and q such that

$$\frac{\|\mathbf{w}\|_\infty}{\|\mathbf{w}\|_1} \leq \frac{q}{k}$$

The quantity q thus controls the maximum weight that any one feature can take. While these quantities are not directly optimizable in our QP formulation, we can access them by varying the hyperparameters λ, α, and β.

Intuitively, we note that the choice of λ controls whether $\mathbf{w}$ satisfies q and that α and β control whether the correlation constraint s is satisfied. To satisfy these constraints, we simply grid search across feasible values of λ, α, and β. We solve the QP for fixed values of λ, α, and β, keeping only the solutions for which the s and q 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 more easily interpretable and natural hyperparameters.

We recommend that only s (minimum correlation) and not q (maximum feature weight) be used to control Schema's optimization. The default value of q in our implementation is set to be very high (10^3) so that it is not a binding constraint in

most cases. We recommend not changing it and in future versions of Schema we may reformulate the QP so that q is entirely removed. To limit the distortions in the primary modality, we recommend that s be set close to 1: the default setting of s is 0.99 and we recommended values ≥ 0.9. When Schema is used for feature selection, we recommend aggregating results across an ensemble of runs over a range of s values (a wide range is recommended here) to increase the robustness of the results.

Standard linear decompositions, like PCA or NMF are useful as preprocessing steps for Schema. 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) [SOAC+18]. The transform $\boldsymbol{\omega}$ 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) or those generated by another single-cell analysis method like MOFA+.

5.4.4 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. An expanded version of arguments in this paragraph is available in the appendices. 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 argue intuitively that optimizing the correlation aims to preserve local neighborhood structure (Appendix B.2). Using correlation in the objective also affords the flexibility to broaden $\text{Corr}(\mathbf{w}, \rho_j)$ in Equation (5.2) to any function f_j of the metric: i.e., $\text{Corr}(\mathbf{w}, f_j \circ \rho_j)$; this allows us to invert the direction of alignment or more heavily weigh local distances.

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 (Section 5.4.6). This enables Schema to continue scaling to more massive scRNA-seq datasets.

5.4.5 Connections to linear decomposition methods

Like popular linear decomposition techniques, e.g., principal component analysis (PCA) or canonical correlation analysis (CCA) [Hot36], Schema runs quickly and performs only linear transformations of the data. However, there is a gap between methods like PCA, which explain the variance in a *single* dataset, and those like CCA, which focus on explaining the covariance *between* datasets. The former cannot integrate information across datasets; the latter de-emphasize structure within a dataset. The assumption underlying Schema is that both the variance *within* a dataset and covariance *across* datasets carry important biological information. Specifically, our focus on the importance of preserving *neighborhood structure* in the primary dataset explicitly emphasizes the within-dataset relationships we care about.

Schema effectively complements these existing decomposition techniques. As discussed earlier, one can use a change-of-basis transform as a preprocessing step, so that the QP-derived scaling transform puts weights on the transformed features in a way that preserves neighborhood structure. This allows, for example, principled dimensionality reduction by choosing the heaviest weighted features instead of just an arbitrary number of principal components.

Our correlation-based alignment approach does parallel kernel CCA [Aka01], a generalization of CCA where arbitrary distance metrics can be specified when correlating two datasets. While Schema offers similar flexibility for 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. 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.

5.4.6 Efficiency and approximation

Schema's efficiency stems from our 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. Additionally, while the full Schema algorithm has quadratic scalability 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, enabling *sublinear* scalability in the number of cells that will be crucial as multimodal datasets increase in size.

Our approach is to show that, given a $\hat{\boldsymbol{\omega}}$ 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 Equation (5.1). 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 $\mathcal{P}$ be a random subset of all possible interactions. For now, we assume that interactions are chosen uniformly at random. Solving the optimization problem in Equation (5.2) with our sample $\mathcal{P}$ yields $\hat{\boldsymbol{\omega}}$, an estimator for the true optimal transform $\boldsymbol{\omega}$. We show that $\hat{\boldsymbol{\omega}}$ approximates $\boldsymbol{\omega}$ well by showing that the pairwise distances among cells in $\mathbf{X}$ followed by a transformation by $\hat{\boldsymbol{\omega}}$ have high correlations with the secondary datasets as long as $\hat{\boldsymbol{\omega}}$ has high correlations on the subsample.

Formally, we will guarantee, for any $\alpha, \delta > 0$ and sample size at least $|\mathcal{P}| =$

$O\left(\frac{\log(1/\alpha)}{\delta^2}\right)$,

$$\left|\mathrm{Corr}(\hat{\mathbf{w}}, \rho_j) - \widehat{\mathrm{Corr}}(\hat{\mathbf{w}}, \rho_j)\right| < \delta \text{ with probability at least } 1 - \alpha, \tag{5.6}$$

where $\widehat{\mathrm{Corr}}(\,\cdot\,,\,\cdot\,)$ 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 $\mathbf{W} \succeq 0$ be a $k \times k$ psd matrix. We also assume randomly draw pairwise distances $\boldsymbol{\delta}$ 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

$$\mathrm{Corr}(\mathbf{W}, \rho_1) \triangleq \frac{\overbrace{\mathbb{E}[\boldsymbol{\delta}^{\mathrm{T}}\mathbf{W}\boldsymbol{\delta}\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta}]}^{A} - \overbrace{\mathbb{E}[\boldsymbol{\delta}^{\mathrm{T}}\mathbf{W}\boldsymbol{\delta}]}^{B}\overbrace{\mathbb{E}[\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta}]}^{C}}{\underbrace{\mathrm{Var}^{1/2}(\mathbf{W})}_{D}\underbrace{\mathrm{Var}^{1/2}(\rho_1)}_{E}}. \tag{5.7}$$

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

$$\begin{aligned}
\widehat{\mathrm{Corr}}(\mathbf{W}, \rho_1) &\geq \frac{(A-\epsilon)-(B+\epsilon)(C+\epsilon)}{(D+\epsilon)(E+\epsilon)} \\
&\approx \frac{A-BC-(1+B+C)\epsilon}{DE(1+\epsilon/D)(1+\epsilon/E)} \\
&\approx \left(\frac{A-BC}{DE} - \frac{B+C+1}{DE}\epsilon\right)(1-\epsilon/D)(1-\epsilon/E) \\
&\approx \left(\frac{A-BC}{DE} - \frac{B+C+1}{DE}\epsilon\right)(1-\epsilon/D-\epsilon/E) \\
&\approx \left(\frac{A-BC}{DE}\right)\left(1+\frac{D+E}{DE}\epsilon\right) - \frac{B+C+1}{DE}\epsilon \\
&= \mathrm{Corr}(\mathbf{W}, \rho_1)\left(1 - \frac{\mathrm{Var}^{1/2}(\mathbf{W}) + \mathrm{Var}^{1/2}(\rho_1)}{\mathrm{Var}^{1/2}(\mathbf{W})\mathrm{Var}^{1/2}(\rho_1)}\epsilon\right) - \frac{1+\mathbb{E}[\boldsymbol{\delta}^{\mathrm{T}}\mathbf{W}\boldsymbol{\delta}] + \mathbb{E}[\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta}]}{\mathrm{Var}^{1/2}(\mathbf{W})\mathrm{Var}^{1/2}(\rho_1)}\epsilon.
\end{aligned}$$

Thus, for a desired overall confidence level η,

$$\epsilon = \left(\frac{\mathrm{Var}^{1/2}(\mathbf{W})\mathrm{Var}^{1/2}(\rho_1)}{\max\left\{\mathrm{Var}^{1/2}(\mathbf{W}) + \mathrm{Var}^{1/2}(\rho_1), 1 + \mathbb{E}[\boldsymbol{\delta}^{\mathrm{T}}\mathbf{W}\boldsymbol{\delta}] + \mathbb{E}[\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta}]\right\}}\right)\eta.$$

To show that we can bound each of the terms A, B, C, D, E we use *Hoeffding's inequality* [Hoe63] 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 $[a, b]$, let $s \triangleq b - a$, and let $\bar{x} \triangleq \frac{1}{n}\sum_{i=1}^{n} x_i$. 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

$$\mathbb{E}[x] \geq \bar{x} - t \text{ with confidence } 1-\alpha \text{ for } n \geq \frac{s^2 \log(1/\alpha)}{t^2}. \tag{5.8}$$

We begin by applying the inequality on term $A \triangleq \mathbb{E}[\boldsymbol{\delta}^{\mathrm{T}}\mathbf{W}\boldsymbol{\delta}\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta}]$ by bounding $\boldsymbol{\delta}^{\mathrm{T}}\mathbf{W}\boldsymbol{\delta}\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta}$. It is clear that $\left|\boldsymbol{\delta}^{\mathrm{T}}\mathbf{W}\boldsymbol{\delta}\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta}\right| \leq \left|\boldsymbol{\delta}^{\mathrm{T}}\mathbf{W}\boldsymbol{\delta}\right|\left|\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta}\right|$, so we can bound each individually. Note that we can assume without loss of generality that $\mathbf{W}$ is diagonal

here, because otherwise (since it is psd), we could write $\mathbf{W} = \mathbf{U}\mathbf{D}\mathbf{U}^{\mathrm{T}}$, where $\mathbf{D}$ is diagonal and $\mathbf{U}$ is unitary; setting $\mathbf{y} \triangleq \mathbf{U}\boldsymbol{\delta}$ yields $\left|\boldsymbol{\delta}^{\mathrm{T}}\mathbf{W}\boldsymbol{\delta}\right| = \left|\mathbf{y}^{\mathrm{T}}\mathbf{D}\mathbf{y}\right|$, and, by unitarity, $\|\boldsymbol{\delta}\| = \|\mathbf{y}\|$.

Then, by Cauchy-Schwarz,

$$\left|\boldsymbol{\delta}^{\mathrm{T}}\mathbf{W}\boldsymbol{\delta}\right| \leq \left|\sum \delta_i W_{ii} \delta_i\right| \leq \|\mathbf{W}\|\|\boldsymbol{\delta}\|^2 \tag{5.9}$$

where $\|\mathbf{W}\|$ is the matrix-norm, i.e., $\|\mathbf{W}\| \triangleq \sqrt{\mathrm{Tr}(\mathbf{W}^{\mathrm{T}}\mathbf{W})}$. So for a diagonal matrix, $\|\mathbf{W}\|^2 = \sum W_{ii}^2$. We can bound $\|\boldsymbol{\delta}\| \leq \max_{x_i, x_j \in \mathbf{X}}\{\|x_i - x_j\|\} \triangleq \mathrm{diam}(\mathbf{X})$.

Thus, $\left|\boldsymbol{\delta}^{\mathrm{T}}\mathbf{W}\boldsymbol{\delta}\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta}\right| \leq \|\mathbf{W}\|\mathrm{diam}^4(\mathbf{X})$.

To get a confidence interval of size ϵ, we plug into Equation (5.8), so we require

$$N \geq \frac{\|\mathbf{W}\|\mathrm{diam}^8(\mathbf{X})\log(1/\alpha)}{\epsilon^2}.$$

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 $\|\mathbf{W}\|$ by choice of hyperparameters.

The same analysis can be used for terms B and C in Equation (5.7), 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 that

$$\mathrm{Var}(\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta}) = \mathbb{E}[(\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta} - \mathbb{E}[\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta}])^2].$$

Again, $\left|\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta} - \mathbb{E}[\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta}]\right|$ is bounded by the maximum squared distance in the dataset, i.e., $\mathrm{diam}^2(\mathbf{X})$, so we can use the Hoeffding inequality from above in the same way.

And term D takes the same form as above, but with $\boldsymbol{\delta}^{\mathrm{T}}\mathbf{W}\boldsymbol{\delta}$ instead of $\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta}$. As noted in Equation (5.9), this is a bounded random variable as well, but here with bound $\|\mathbf{W}\|^2\mathrm{diam}^4(\mathbf{X})$.

Thus, in order to get a uniform confidence interval across all the terms, we require

$$N \geq \frac{\|\mathbf{W}\|^2\mathrm{diam}^8(\mathbf{X})\log(1/\alpha)}{\epsilon^2}. \tag{5.10}$$

5.5 Empirical performance and generality

When evaluating the performance of Schema, we wanted to demonstrate not only its ability to reveal biological insights but also its generality across many modalities and analytic scenarios. In this section, we apply Schema across to a host of problems in single-cell genomics and then, in our application note, we apply it in a particularly creative way to immune sequence variation data in the following section.

5.5.1 Inferring cell types

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 of single cells [CCR+18]. Focusing on cell type inference, a key analytic step in many single-cell studies, 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 5-2F).

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 ($\geq 99\%$). We then clustered the cells by performing Leiden community detection [TWvE19] (a recently introduced modification of the Louvain algorithm [BGLL08]) on the transformed dataset and compared these clustering assignments to the Leiden clusters obtained without Schema transformation. We used the expertly defined-labels from Cao et al. [CCR+18] as the ground truth cluster labels and quantified clustering agreement with the adjusted Rand index (ARI) [Ran71], which has a higher value if there is 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

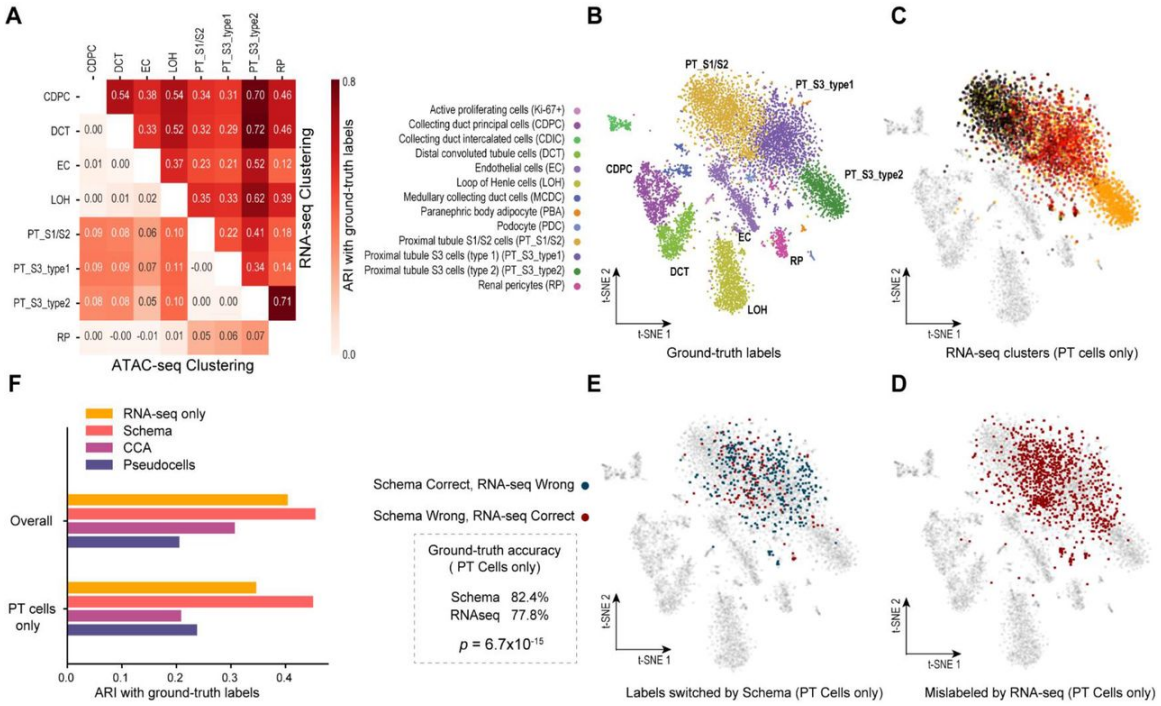

Figure 5-2: Clustering of synthesis of RNA-seq and ATAC-seq.

(**A**) Leiden clustering of cellular profiles results in greater ARI agreement 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. (**C**, **D**) 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. (**E**) Schema synthesis of RNA- and ATAC-seq features, followed by Leiden clustering, results in significantly greater concordance with ground-truth 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 synthesized data are higher, especially for PT cells. Synthesizing the modalities using canonical correlation analysis (CCA) or a "pseudocell" approach described in the original study results in lower ARI scores.

labels using just RNA-seq or ATAC-seq datasets individually (ARIs of 0.40 and 0.04, respectively, Figure 5-2F). 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, 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. We note that an unconstrained synthesis of the modalities using canonical correlation analysis (CCA) resulted in an ARI of 0.31, lower than what is achieved by using just the RNA-seq modality (Figure 5-2F). However, since Schema constrains the ATAC-seq modality's influence when synthesizing it with RNA-seq data, we could extract additional signal provided by ATAC-seq while preserving the rich information provided by the transcriptomic modality. We also evaluated a heuristic approach described in the original study: group cells into small clusters ("pseudocells") by RNA-seq similarity and compute an average ATAC-seq profile per pseudocell, using these profiles for the final clustering. This approach also underperformed Schema (ARI of 0.20).

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 5-2A). PT cells are the most numerous cells in the kidney dataset and many of the misclassifications in the RNA-seq based clustering relate to these cells (Figure 5-2B,C,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 (one-sided binomial $P = 6.7 \times 10^{-15}$), leading to an overall improvement in clustering quality (Figure 5-2E).

5.5.2 Developmental differential expression

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 [PSGG$^+$19], 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 5-3A). 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 $\pm 25\%$ variations in these weights. We used RNA-seq data as the primary dataset, representing it by its top ten principal components.

We evaluated Schema alongside MOFA+, a recently introduced single-cell multimodal analysis technique [AAB$^+$20]. 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 Argelaguet et al.'s MOFA+ analysis of this dataset, the authors identified 10 factors that capture similar information to the top principal components. To identify differentially expressed genes with MOFA+, we selected the top genes from two factors (MOFA1 and MOFA4) reported by Argelaguet et al. 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

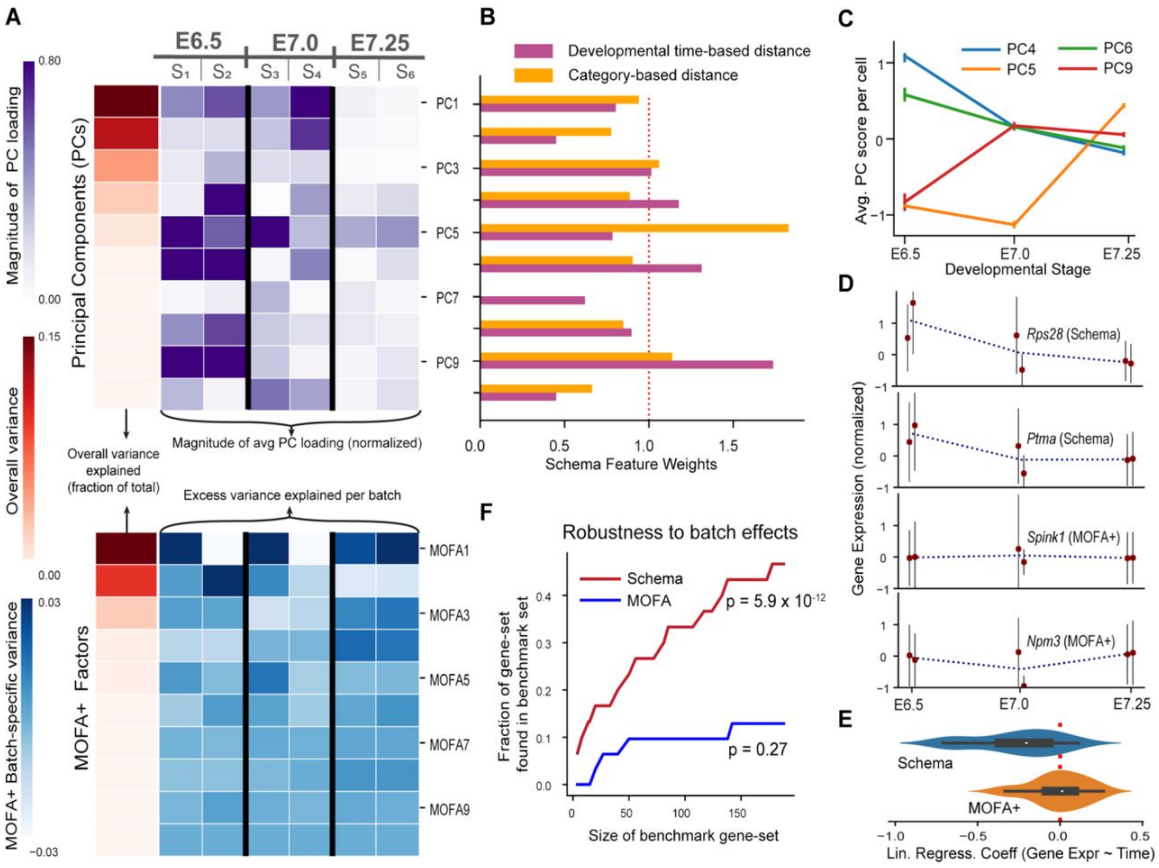

Figure 5-3: Developmental differential expression.

(**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, $P = 3.8 \times 10^{-6}$); 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).

monotonically changing expression along the time course (Figure 5-3B,C,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 5-3B,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 found that the Schema genes corresponded to regression coefficients significantly different from zero (Figure 5-3D,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.

We then measured the overlap of Schema and MOFA+ gene sets with these benchmarks (Figure 5-3F) and found that, compared to MOFA+, the Schema gene set shows a markedly higher overlap with the benchmarks that is statistically significant

(Bonferroni-corrected hypergeometric $P = 5.9 \times 10^{-12}$ 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.

5.5.3 Spatial differential expression

We performed some preliminary analysis with Schema of spatial transcriptomics data [RSG+19], another increasingly important multimodal scenario, here encompassing gene expression, cell-type labels, and spatial location. 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 (noting that these transcriptomes are not guaranteed to be single-cell), and thus cell types are located throughout the tissue [RSG+19, SMW+18]. Interestingly, we observed spatial density variation for certain cell types; specifically, transcriptomes corresponding to granule cell types are observed in regions of both high and low spatial density (Figure 5-4B). We therefore reasoned that we could use Schema to 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 want to 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, with spatial density controlled by a distance metric that scores two cells as similar if their spatial neighborhoods have matching densities. The densely-packed granule cell genes identified by Schema are strongly enriched for GO terms and REACTOME pathways [FJM+18] related to signal transmission, e.g., 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 $q = 2.87 \times 10^{-11}$). This finding suggests potentially greater neurotransmission activity within these cells.

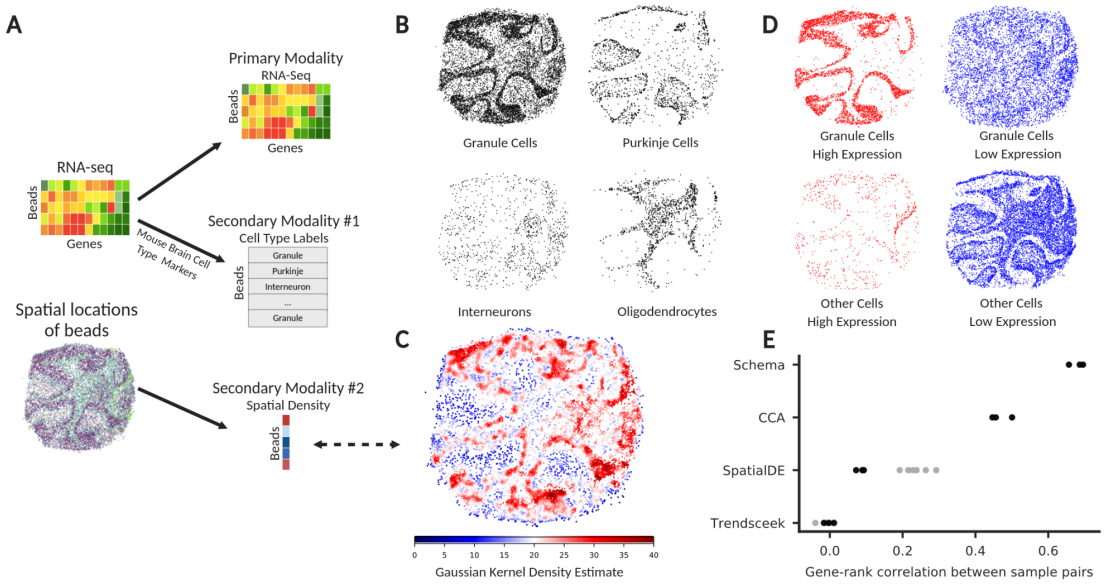

Figure 5-4: Spatial differential expression.

(**A**) Rodriques et al. [RSG+19] simultaneously assayed spatial and transcriptomic modalities of cells in mouse cerebellum tissue (here, data from puck `180430_1` is shown). In addition, they labeled beads (each corresponding to a cell) with 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 section: 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 ranked 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**) Evaluation of the stability of gene rankings computed by Schema, canonical correlation analysis (CCA), SpatialDE and Trendsceek on three replicates sourced from mouse cerebellum tissue. The black points indicate the Spearman rank correlation of gene scores across pairs of replicates. The grey points show the cross-replicate gene-rank correlation of the intermediate results.

We sought to benchmark our method on spatial transcriptomics data by comparing the robustness of Schema's results with those based on canonical correlation analysis (CCA) [Hot36] and with two methods specifically intended for spatial transcriptomics, namely SpatialDE [STS18] and Trendsceek [EJS18].

An important point is that CCA, SpatialDE, and Trendsceek are less general 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 (we focus on spatial density variation). 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 to incorporate any number of additional data modalities.

To evaluate the stability and quality of spatial transcriptomic analysis across different techniques, we analyzed three replicate samples of mouse cerebellum tissue (coronal sections prepared on the same day; pucks `180430_1`, `180430_5`, `180430_6`) and compared the results returned separately for each replicate. While both Schema and CCA identify a gene set that ostensibly corresponds to granule cells in dense regions (Figure 5-4D), 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 (FDR $q < 0.001$ in all cases). We therefore find that Schema's results are substantially more robust across the three replicates. We also find that Schema, in not seeking an unconstrained optimum, is more robust to overfitting to sample-specific noise than CCA (Figure 5-4E).

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 (gray points in Figure 5-4E), we found that SpatialDE 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 5.5.6). These results demonstrate the robustness and efficiency of Schema's supervised approach.

5.5.4 Epigenomics informs expression

We next sought to demonstrate the flexibility of Schema to analyses beyond cell type clustering and differential expression analysis. We turned to a study that simultaneously profiled gene expression and chromatin accessibility from 3,260 human A549 cells [CCR$^+$18]. Using Schema, we characterized the genomic regions (relative to a gene's locus) where chromatin accessibility strongly correlates with the gene's expression variability, i.e., regions whose accessibility is differentially important for highly variable genes. Schema assigned the highest weight to features associated with chromatin accessibility over long ranges, i.e., ~10 megabase (Mb) regions (Figure 5-5C). Searching for an explanation, we investigated if highly variable genes share genomic neighborhoods and mapped gene loci to topologically associated domains32 (TAD) of this cell type. We found strong statistical evidence that highly variable genes are indeed clustered together in TAD compartments (Figure 5-5D), supporting our findings from Schema and suggesting an epigenetic role in controlling gene expression variability.

This demonstration illustrates how Schema's generality facilitates innovative explorations of multimodal data beyond, for example, cell type clustering or differential gene expression analysis. Here we chose genes to be the unit of observation, allowing us to design a primary dataset that links each gene's RNA-seq measurements to ATAC-seq peak counts in its neighborhood. Specifically, each primary feature corresponds to a

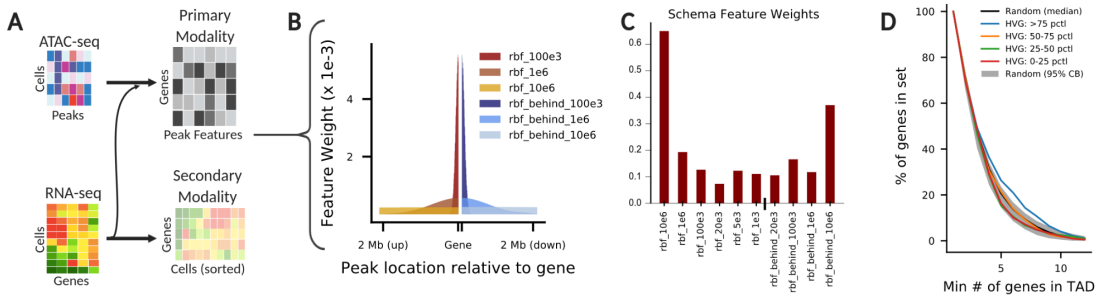

Figure 5-5: Highly variable genes are related to genome topology.

(**A**) We investigated expression variability in the context of chromatin accessibility, evaluating if highly variable genes display differential accessibility around their genomic loci. With genes as the units of observation, we used Schema to analyze simultaneously assayed ATAC-seq and RNA-seq data [CCR+18]. The primary modality captured peak counts while the secondary modality corresponded to gene expression profiles, normalized and sorted so that short Euclidean distances correspond to similar levels of expression variability. (**B**) Each feature of the primary modality corresponded to a genomic region near the gene, scoring how its expression covaries with peak counts in the region. The feature's range is defined by a Gaussian radial basis function, of the form $\exp\{-(d/\lambda)^2\}$, that weighs the contribution of a peak by its distance d from the gene. We defined features upstream and downstream of the gene's transcription start and end sites, respectively. (**C**) Schema identifies long-range features (~10 Mb) as being the most relevant in correlating chromatin accessibility with expression variability. (**D**) To further explore this result, we investigated the organization of highly variable genes in topologically associating domains (TADs). We divided genes into quartiles by expression variability. For each quartile, we plotted the fraction of genes that are in TADs containing k or more genes from the set (k is on the x-axis). The gray region and the black line represent the 95% range and median, respectively, of random baselines generated by shuffling genes between TADs.

genomic window relative to the gene's locus and is scored as the sum over all cells of peak counts in the window, each cell weighted by the gene's expression. We created multiple features, each corresponding to a specific window size and placement (Figure 5-5B). Reusing RNA-seq as the secondary modality, we designed a distance metric that captures similarity in expression variability: for each gene, we normalize and sort the vector of its expression values across cells so that identical vectors imply an identical pattern of gene expression variation. We used Schema as a feature-selection mechanism, where up-weighted features correspond to genomic windows important for explaining gene expression variability.

We then benchmarked this feature selection approach against a ridge regression where features of the primary modality were specified as the explanatory variables and the standard deviation of each gene's expression (summarized from the secondary modality) was the response variable. Both analyses agreed in assigning the highest weights to features corresponding to long-range (~10 Mb) genomic regions upstream and downstream of a gene. However, Schema's regularization mechanism helps produce more consistent and stable feature weights, as evaluated on subsets of genes grouped by strand orientation or chromosome (Figure 5-5C).

To further investigate the connection between chromatin accessibility and gene expression variability, we analyzed gene membership in TADs, hypothesizing that gene expression variability is likely to be influenced by the organization of TADs in the genome. We analyzed the clustering of highly variable genes (HVGs) on TADs within A549 cells (inferred from Hi-C data, ENCODE accession `ENCFF336WPU`) and found that HVGs are indeed more likely to be clustered together in TADs (Figure 5-5D). By two independent permutation-based tests, we were able to reject the null hypothesis that genes in the top quartile of variability are dispersed randomly across TADs ($P < 0.004$ in both cases, with Bonferroni correction). Schema-based feature weighting therefore revealed an association between genomic architecture and gene variability. Notably, these results show how synthesis of multimodal RNA- and ATAC-seq data not only benefits standard analyses like cell type inference, but also enables creative and diverse exploratory data analysis.

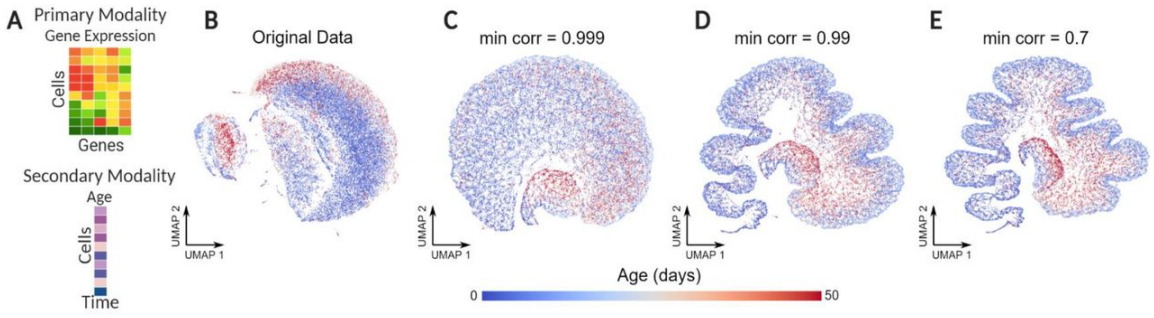

Figure 5-6: Incorporating temporal metadata into visualizations.

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. (**A**) 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. (**B**) UMAP visualization of the original transcriptomic data. (**C**, **D**, **E**) 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.

5.5.5 Visualization

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. Having demonstrated this capability for cell type inference, we now explore another use case. Since widely-used visualization methods such as UMAP [MH18] do not allow a researcher to specify aspects of the underlying data that they wish to highlight in the visualization, 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 [DJK$^+$18] 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 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.

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 5-6). Importantly, accessing this structure required only a limited distortion of the data, corresponding to relatively high values (≥ 0.99) of the minimum correlation constraint (Figure 5-6C). To verify that Schema was able to infuse 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. In contrast, an unconstrained synthesis by CCA leads to a lower correlation (0.059) than seen in the original RNA-seq dataset. Schema thus enables visualizations that synthesize biological metadata, while preserving much of the distance-related correlation structure of the original primary dataset. With Schema, researchers can therefore investigate single-cell datasets that exhibit strong latent structure (e.g., due to secondary metadata like age or spatial location), needing only a small transformation to make that structure visible.

5.5.6 Scalability

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 [RSG+19] (three modalities, 20,823 transcriptomes and 17,607 genes) in 34 minutes, requiring less than 5 GB of RAM in the process. 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. Additionally, while the full Schema algorithm has quadratic scalability 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 (Section 5.4.6), enabling *sublinear* scalability in

the number of cells (with small amounts of error) that will be crucial as multimodal datasets increase in size.

5.6 Application note: Synthesizing immune sensing

To showcase how Schema can reveal insight into a fundamental immunological problem, as well as a domain beyond gene expression, we turn to an exciting multimodal setting in which the sequence of a T-cell receptor (TCR) can be linked to its antigens at massively parallel, single-cell resolution. T cells are an crucial part of adaptive, cell-mediated immunity (Section 2.1.4). When foreign proteins, like those from an infecting virus or bacterium, enter a human cell, those proteins are digested into smaller pieces called peptides; then, the cell has machinery that can the expose those peptides on its surface using a set of proteins called the major histocompatibility complex (MHC). Ideally, T cells can recognize a peptide-MHC (pMHC) antigen that is not from the host and become "activated," which signals to the immune system that the cell expressing the foreign protein should be killed, thereby preventing pathogens from using that cell to replicate. Each T cell has a unique TCR sequence that recognizes a specific pMHC. To sense different foreign antigens, TCR sequences vary immensely across T cells; however, much is still unknown about how the TCR sequence affects antigen binding [MW16].

We therefore integrated multimodal proteomic and functional data with Schema to better understand how sequence diversity in the hypervariable CDR3 segments of TCRs relates to pMHC binding. We analyzed a single-cell dataset [Gen19] that recorded clonotype data for 62,858 T cells and their binding specificities against a panel of 44 pMHCs that come from different viruses [cytomegalovirus (CMV), Epstein-Barr virus (EBV), influenza, human T-cell lymphotropic virus (HTLV), human papillomavirus (HPV), and HIV] and known cancer antigens.

We used Schema's feature-selection capabilities to estimate the sequence locations

and residues in the complementarity-determining region (CDR) 3 segment of α and β chains important to binding specificity. To do so, 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 [Ham50] between observations (i.e., the number of locations at which two sequences differ). In the secondary modalities, we used the Euclidean distance between the binary vector indicating the antigen(s) that bound a given sequence, with 1 indicating binding and 0 otherwise.

Schema assigned low feature weights to the location segments 3–9 in α-chain CDR3 and to and 5–12 in β-chain CDR3. This suggests that those regions can tolerate greater sequence variability while preserving binding specificity. To evaluate these results, we compared them to estimates based on CDR3 sequence motifs sourced from VDJdb [SBZ$^+$18], a curated database of TCRs with known antigen specificities. In VDJdb, TCR motifs are scored using an adaptation of the relative-entropy algorithm [MMWC12] 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 5-7). 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 (Spearman rank correlations of 0.38 and 0.92 for the α and β chains, respectively; Figure 5-7C-D). 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.

Next, we used Schema to investigate the selection pressure on the actual amino acid values at the variability-prone locations identified above. We first selected a sequence location (e.g., location 4 in α chain CDR3) and constructed a primary modality where each cell was represented by a one-hot encoding of the amino acid at the location (i.e., a 20-dimensional boolean vector). The secondary modality was binding specificity information, as before. We performed separate Schema runs for each such location of

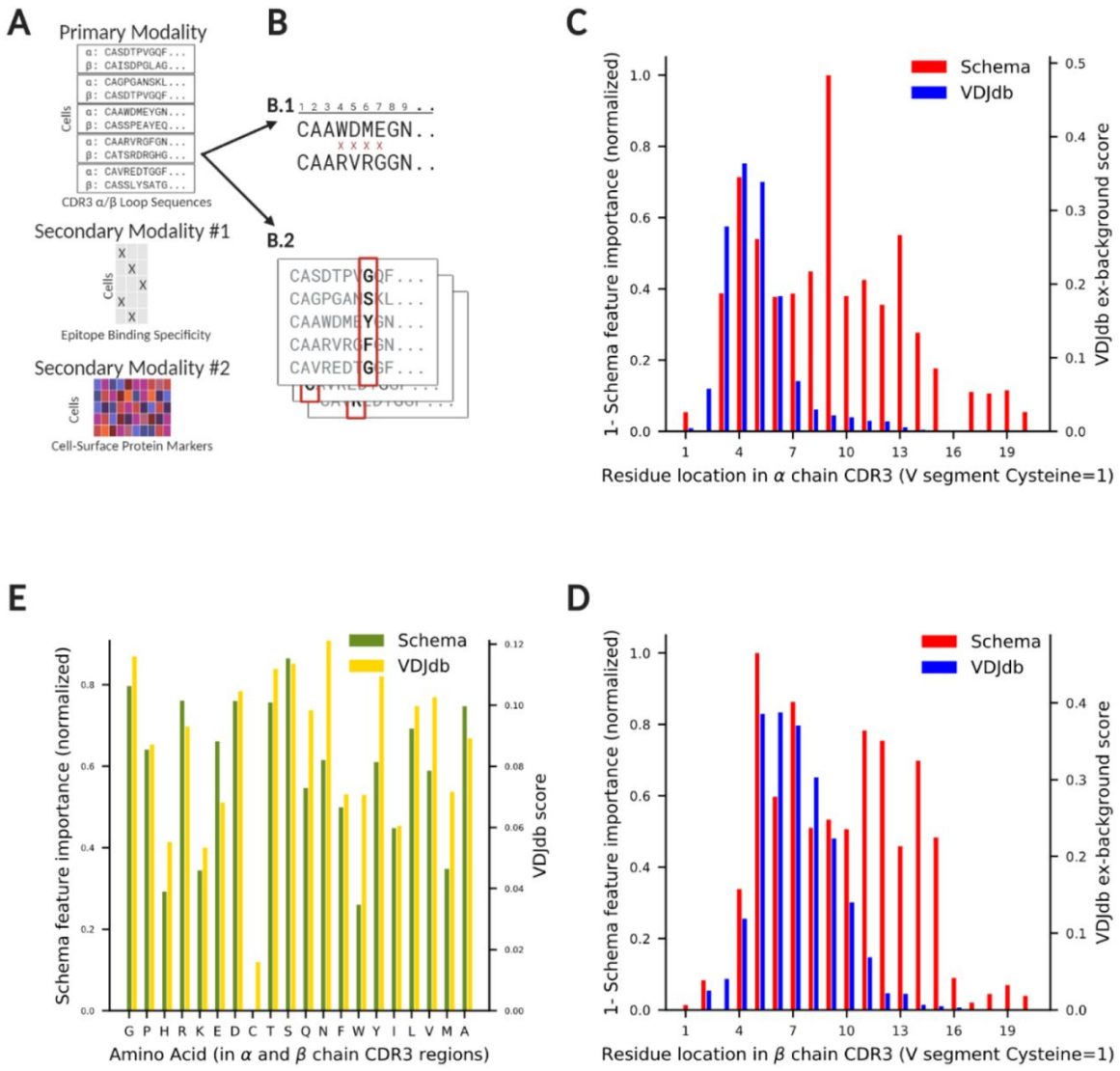

Figure 5-7: Sequence affect on TCR binding specificity.

(**A**) We analyzed a multi-modal dataset to understand how TCR 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 antigens. (**B**) We performed two Schema analyses: (**B.1**) To infer location-wise selection pressure, the feature vector of the primary modality was the CDR3 sequence, with the Hamming distance between two sequences as the metric. (**B.2**) The second analysis aimed to understand amino acid selection pressure at locations that showed high variability. For each such location, a one-hot encoding of the amino acid at the location was used as the feature vector. (**C**, **D**) Schema identifies sequence locations 3–9 (α chain) and 5–12 (β chain) as regions where sequences can vary with a modest impact on binding specificity. We compared Schema's scores to statistics computed from motifs in VDJdb [SBZ+18]. (**E**) Schema and VDJdb agree on the relative importance of amino acids in preserving binding specificity (Spearman correlation of 0.74, two-sided t-test $P = 2 \times 10^{-4}$).

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 (Spearman rank correlation $= 0.74$, two-sided t-test $P = 2 \times 10^{-4}$).

As T-cell engineering improves and designing a custom TCR for an antigen of interest becomes an increasingly viable therapeutic strategy [DFGH$^+$17], it is also important to focus on the the key sequence locations and amino acids that govern antigen binding. The location and amino acid preferences estimated by Schema in this application note can be used directly in any algorithm for computational design of epitope-specific CDR3 sequences to bias search towards more functionally plausible candidate sequences. Our analysis shows that, with Schema, it is possible to recover from a *single* experiment fundamental patterns of T-cell sensing that had only previously been available by combining information across many studies and labs.

Chapter 6

Fighting Disease I: Discovery

There is a balm in Gilead
To make the wounded whole;
There is a balm in Gilead
To heal the sin-sick soul!

—African-American spiritual (19th century)

Ultimately, to end disease, we need the tools to fight it. Antivirals and antibiotics can help manage or cure disease, whereas, through a miracle of modern medicine, vaccines enable disease prevention. Perhaps the most direct contribution that algorithms can make to therapeutic discovery is to actually propose new drugs or to repurpose existing drugs for new problems.

We now shift from trying to understand infectious disease to leveraging this knowledge to fight disease. In this chapter and the next, we focus on the problem of drug-target interaction (DTI) prediction [YAG+08, HCB18]. The concept is simple: a supervised machine learning algorithm, given information about a drug (e.g., a small-molecule antibiotic) and a biological target (e.g., a protein critical for bacterial replication), is trained to predict if the drug-target pair interacts. Traditionally, DTI prediction refers to a binary classification problem (i.e., the algorithm predicts a positive or negative interaction), but more general variants involving multiclass

classification (e.g., predicting different kinds of interaction, like inhibition or induction) or involving regression (e.g., predicting a continuous binding score) also exist.

DTI prediction can be used to repurpose existing drugs for a target of interest or to find new targets (or side effects) of an existing drug. A DTI prediction algorithm can also be used to evaluate the properties of a new drug. The most immediate application of DTI prediction therefore is *drug discovery*.

In this chapter, we first review some of the initial approaches to DTI prediction and lay out their problems, including poor scalability to large pharmacological datasets. We then describe a more efficient approach to DTI prediction based on neural networks that not only outperforms these initial approaches but also enables efficient training and prediction [HCB18]. Our approach is so efficient that it has a practical runtime even when implemented in a privacy-preserving secure computation framework, which reveals no information about the underlying data to the computing parties but also incurs a large cryptographic computational overhead[1].

However, when we evaluated this approach in a practical setting to suggest new interactions that we then experimentally validated, we observed a much higher false positive rate than anticipated. The experiments and results presented in this chapter, therefore, set up the work in the subsequent chapter that also looks for DTIs with machine learning, but which does so in a way that improves the certainty and the quality of new predictions. These two chapters are therefore meant to be read together, with the background and the methods described in this chapter primarily meant to motivate the next. We also defer a practical infectious disease-related application note to the end of the next chapter.

6.1 Glossary

- *Drug-target interaction (DTI) prediction*: Given a drug and a target, predict whether they interact or not (and, optionally, what kind of interaction and how

[1]Software for the work described in this chapter is available at `https://github.com/brianhie/secure-dti`.

strong the interaction is).

- *Recommender system*: An algorithm that recommends products to users (e.g., recommending Netflix movies to viewers).

- *Sparsity*: Describes the amount of zero values in a matrix, where more zero values indicates greater sparsity.

- *Matrix factorization*: An algorithm that decomposes a matrix into the product of two lower dimensionality matrices; useful to uncover potential low-dimensional structure in the initial matrix.

- *Network diffusion*: An algorithm that intuitively simulates the flow or "diffusion" of information through a network, where edge weights and the network topology dictate how much information flows to each node.

6.2 Preliminaries

In this section, we summarize some of the earliest machine learning approaches to DTI prediction. Given a dataset of drugs $\mathcal{X}_D$ and a dataset of targets $\mathcal{X}_T$, the goal of DTI prediction learns a function $f : \mathcal{X}_D \times \mathcal{X}_T \to \mathcal{Y}$ where the $\mathcal{Y}$ describes the interactivity between a drug-target pair. In this chapter, we are primarily concerned with the binary classification setting, i.e., $\mathcal{Y} = \{0, 1\}$, though a regression setting where $\mathcal{Y} = \mathbb{R}$ is also possible.

6.2.1 Recommender systems

The earliest attempts at DTI prediction borrowed from the large machine learning literature on *recommender systems* [KBV09]. These systems were built to address a common commercial need: given a set of users and a set of products (e.g., books on Amazon or movies on Netflix), how do you recommend new products to a user based on that user's consumption history?

A first challenge is that of *sparsity*. A user might provide explicit ratings (e.g., "like" or "disklike") for a few products, but leave many products unrated. There is thus incomplete information as to user preferences, so the recommender system is tasked with imputing this information from existing patterns.

A second challenge is when a new user arrives who has not provided feedback on any products, which is referred to as the *cold start* problem [KBV09]. The simplest solution is to simply recommend a random set of products to the user, or the most popular products across the general population. A more intelligent solution is to also collect *side information* about all users (e.g., demographic information) that can then be used to compare users even with no previous consumption history and recommend items to users in this cold start setting. For example, a new 18-year-old user can be recommended a different set of products than a new 68-year-old user.

The connection to the DTI setting is straightforward. Rather than recommend products to potential users, the same algorithms can recommend drugs to potential targets. Moreover, similar challenges involving sparsity and cold start also apply; typically, a drug has positive or negative interaction information for just a few targets and new drugs start with no interaction information at all.

6.2.2 Matrix factorization

A common approach to recommender systems and DTI prediction leverages a linear algebra technique called collaborative matrix factorization (CMF) [SG08, ZDMZ13, CLH+13]. The input to the algorithm is a matrix $\mathbf{X} \in \{0,1\}^{N \times M}$, in which the rows correspond to unique drugs and the columns correspond to unique targets. The entry in the ith row and jth column is 1 if drug i positively interacts with target j, or 0 otherwise (i.e., a negative or unknown interaction). In most settings, $\mathbf{X}$ is a sparse matrix.

The idea in CMF is to impute missing values in $\mathbf{X}$ by exploiting low-rank structure. By decomposing $\mathbf{X}$ into the matrix product of two low-dimensional matrices $\mathbf{Z}_D \in \mathbb{R}^{N \times k}$ and $\mathbf{Z}_T \in \mathbb{R}^{M \times k}$, CMF learns a more general notion of interactivity that it can

use to predict new interactions via the objective function

$$\min_{\mathbf{Z}_D, \mathbf{Z}_T} \left\{ \|\mathbf{X} - \mathbf{Z}_D \mathbf{Z}_T^{\mathrm{T}}\|_{\mathrm{F}}^2 + \lambda_m (\|\mathbf{Z}_D\|_{\mathrm{F}}^2 + \|\mathbf{Z}_T\|_{\mathrm{F}}^2) \right\}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm of a matrix and λ_m is a constant controlling a regularization term. Predicted DTIs are those with high imputed values in $\tilde{\mathbf{X}} \triangleq \mathbf{Z}_D \mathbf{Z}_T^{\mathrm{T}}$.

To deal with the cold start problem, CMF introduces side information in the form of similarity matrices $\mathbf{S}_D \in \mathbb{R}^{N \times N}$ and $\mathbf{S}_T \in \mathbb{R}^{M \times M}$ such that, e.g., an entry s_{ij} in $\mathbf{S}_D$ encodes a measure of chemical structure similarity between the ith and jth drugs. Side information can be incorporated into the CMF objective as

$$\min_{\mathbf{Z}_D, \mathbf{Z}_T} \left\{ \|\mathbf{X} - \mathbf{Z}_D \mathbf{Z}_T^{\mathrm{T}}\|_{\mathrm{F}}^2 + \lambda_m (\|\mathbf{Z}_D\|_{\mathrm{F}}^2 + \|\mathbf{Z}_T\|_{\mathrm{F}}^2) \right\} + \lambda_D \|\mathbf{S}_D - \mathbf{Z}_D \mathbf{Z}_D^{\mathrm{T}}\|_{\mathrm{F}}^2 + \lambda_T \|\mathbf{S}_T - \mathbf{Z}_T \mathbf{Z}_T^{\mathrm{T}}\|_{\mathrm{F}}^2$$

where λ_D and λ_T are also regularization constants [CLH+13]. Intuitively, CMF learns embeddings of drugs and targets that reconstruct known interaction information while also sharing information across similar drugs and similar targets.

6.2.3 Network diffusion

CMF is based on the more general intuition that we observe a sparse subset of the full drug-target space $\mathcal{X}_D \times \mathcal{X}_T$ and that the goal of DTI prediction is to impute values that "fill in" the information corresponding to unobserved interactions. This intuition is not limited to a linear algebraic formulation but also has connections to graph theory. In a graph theoretic approach to DTI prediction based on "network diffusion" [MKY+13, WYZL14], each drug and each target forms a node in a network. Edges in the network can be drawn between known drug-target pairs.

The goal of network diffusion is "transfer" information from known DTIs to unknown drug-target pairs. A common algorithm for doing this information transfer is called random-walk-with-restart (RWR) [CBP16]. RWR starts off with an adjacency matrix $\mathbf{A} \in \mathbb{R}_{\geq 0}^{d \times d}$ that is used to construct a transition probability matrix $\mathbf{P} \in [0, 1]^{d \times d}$

such that

$$p_{i,j} \triangleq \frac{a_{i,j}}{\sum_{j'} a_{i,j'}}$$

where $a_{i,j}$ is the element in the ith row and jth column of $\mathbf{A}$ and $p_{i,j}$ is defined similarly. Let $\mathbf{s}_i \in [0,1]^d$ store in each element the probability of a node being visited from the ith node. RWR relies on iterative updates to $\mathbf{s}_i$ in the form

$$\mathbf{s}_i^{(t+1)} \triangleq (1-\theta_r)\mathbf{P}\mathbf{s}_i^{(t)} + \theta_r \mathbf{e}_i,$$

where $\theta_r \in [0,1]$ is a "restart probability" parameter and $\mathbf{e}_i \in \{0,1\}^d, \|\mathbf{e}_i\|_2 = 1$ is a standard basis vector. RWR returns a fixed point $\mathbf{s}_i^{(\infty)}$. Intuitively, RWR starts off with a unit of information on each node that "diffuses" to other nodes based on the adjacency matrix $\mathbf{A}$.

In one RWR-inspired DTI prediction method [WYZL14], an initial sparse drug-target adjacency matrix $\mathbf{X}^{(0)} \in \{0,1\}^{N \times M}$ is created. To incorporate side information, edges can also be drawn between drugs where the weight on each edge is related to the similarity of those drugs, with higher weight placed on more similar drugs, and edges are drawn between targets based on target similarity (essentially forming the $\mathbf{S}_D$ and $\mathbf{S}_T$ matrices described for CMF above). These go into an iterative update rule

$$\mathbf{X}^{(t+1)} \triangleq \alpha \mathbf{P}\mathbf{X}^{(t)} + (1-\alpha)\mathbf{X}^{(0)}$$

where $\mathbf{P} \in [0,1]^{N \times N}$ is the normalized matrix computed based on $\mathbf{X}^{(t)}\mathbf{S}_T\mathbf{X}^{(t)\mathrm{T}}\mathbf{S}_D$ and $\alpha \in [0,1]$ is the restart parameter. DTI predictions are obtained in $\mathbf{X}^{(\infty)}$ after convergence.

6.2.4 DTI prediction challenges

As is probably apparent from the above discussion, there are many similarities between approaches based on linear algebra and those based on graph theory. More complex approaches for DTI prediction are built off of similar concepts. One of the most influential approaches is DTINet [LZZ+17], which uses RWR combined with matrix

factorization and optimization approaches to synthesize drug-drug and target-target similarity matrices across multiple modalities (e.g., structural similarity and functional similarity).

An important limitation of DTI prediction based on matrix factorization, network diffusion, or some combination of these approaches is that they are infeasible on modern datasets primarily because their computations scale quadratically with the number of drugs N and the number of targets M in the dataset (e.g., $O(N^2)$ or $O(NM)$), which is prohibitive for realistic datasets with millions of compounds.

A different approach is to use a supervised machine learning model to directly learn the function $f : \mathcal{X}_D \times \mathcal{X}_T \rightarrow \mathcal{Y}$ using, e.g., a support vector machine or a neural network. Side information is provided to the model as features and the model therefore implicitly learns the similarity information during training. Importantly, this approach enables linear time scalability in the number of observed drug-target training pairs, which is typically much less than the full space of all possible interactions. It is this approach that we describe in the sections below.

6.3 Neural network for DTI prediction

6.3.1 Motivation

Scalability to large DTI datasets and improving prediction performance are the primary motivating reasons for a neural approach to DTI prediction Scalability is especially important as collaborative efforts to develop new, life-saving drug therapies have recently begun to take shape among pharmaceutical companies and academic labs, despite the highly competitive nature of the industry [Rea14, Wil17]. Driving this transformation is the stalled or declining productivity of existing drug development pipelines amidst growing financial and regulatory pressures. Many in industry and academia are realizing that the difficult task of identifying novel drug candidates would be more successful if they leveraged pooled experimental datasets and knowledge that go beyond any single organization, resulting in datasets with millions of DTIs.

6.3.2 Neural network model

To achieve scalable computation while maintaining high accuracy, we draw from recent advances in deep learning [GBC16] to train a neural network model for DTI prediction. Our neural network takes feature representations of a compound and a target as input, and predicts the interactivity of the given pair. Although we used chemical structure fingerprints and protein domain annotations as input features in our computational experiments, our framework readily generalizes to alternative features. We circumvent the quadratic complexity of existing methods by training our neural network over a dataset consisting of only the observed DTIs and a comparable number of putatively non-interacting drug-target pairs, which is typically vastly smaller than the full drug-by-target matrix.

We now define our neural network model. We are given a feature matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$, where each row corresponds to a single training example and each column corresponds to a single data feature. We are also given a label vector $\mathbf{y} \in \{-1, +1\}^M$ where $y_i = +1$ if $\mathbf{X}_{:,i}$ is a positive training example and $y_i = -1$ otherwise. While we assume binary labels in our work, our framework easily generalizes to continuous interaction scores.

Our neural network model is a standard multilayer perceptron [GBC16], consisting of real-valued weight matrices $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L+1)}$ and column vector biases $\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(L+1)}$, where L is the number of hidden layers and where each hidden layer consists of neurons $\mathbf{Z}^{(1)}, \ldots, \mathbf{Z}^{(L)}$. During the forward-propagation phase, certain neurons are "activated" according to

$$\mathbf{Z}^{(1)}_{:,i} \triangleq f_{\text{act}}(\mathbf{W}^{(1)}\mathbf{X}_{:,i} + \mathbf{b}^{(1)}) \quad \text{and} \tag{6.1}$$

$$\mathbf{Z}^{(l)}_{:,i} \triangleq f_{\text{act}}(\mathbf{W}^{(l)}\mathbf{Z}^{(l-1)}_{:,i} + \mathbf{b}^{(l-1)}) \tag{6.2}$$

for $l = 2, \ldots, L$ and $i = 1, \ldots, M$. For our purposes, we assume each hidden layer has the same number of neurons, which we denote H, where $\mathbf{Z}^{(l)} \in \mathbb{R}^{H \times M}$. The function f_{act} is known as an activation function, which in our neural network is the rectified

linear unit (ReLU) [GBB11], which takes the form $f_{\text{act}}(x) \triangleq \max\{0, x\}$. After the final hidden layer, our model outputs scores $\mathbf{s} \in \mathbb{R}^M$ where

$$\mathbf{s}_i = \mathbf{W}^{(L+1)}\mathbf{Z}^{(L)}_{:,i} + \mathbf{b}^{(L+1)}$$

for $i = 1, \ldots, M$. Note that in our single output setting, $\mathbf{W}^{(L+1)} \in \mathbb{R}^{1\times H}$ and $\mathbf{b}^{(L+1)} \in \mathbb{R}$. We evaluate the predictive performance of the model using the hinge loss function,

$$\mathcal{J}(\mathbf{s}, \mathbf{y}) \triangleq \frac{1}{M}\sum_{i=1}^{M} \max\{0, 1 - s_i y_i\}.$$

Next, we use these errors to compute derivative updates to the weights and biases, starting with the output layer, where

$$\boldsymbol{\delta}^{(L+1)\mathrm{T}} = \mathbf{y} \odot \frac{1}{M}\mathbb{1}\{\mathbf{1} - \mathbf{s} \odot \mathbf{y} > \mathbf{0}\}, \tag{6.3}$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(L+1)}} = \boldsymbol{\delta}^{(L+1)}\mathbf{Z}^{(L)\mathrm{T}} + \lambda\mathbf{W}^{(L+1)}, \quad \text{and} \tag{6.4}$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{b}^{(L+1)}} = \boldsymbol{\delta}^{(L+1)}\mathbf{1} \tag{6.5}$$

where $\odot$ is the component-wise, or Hadamard, product. Note that we add a regularization term to the weight updates parameterized by the constant λ. Following the standard back-propagation algorithm for training neural networks, these derivatives are recursively propagated through each hidden layer using

$$\boldsymbol{\delta}^{(l)} = \left(\mathbf{W}^{(l+1)\mathrm{T}}\boldsymbol{\delta}^{(l+1)}\right) \odot \mathbb{1}\{\mathbf{Z}^{(l)} > \mathbf{0}\}, \tag{6.6}$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(l)}} = \boldsymbol{\delta}^{(l)}\mathbf{Z}^{(l-1)\mathrm{T}} + \lambda\mathbf{W}^{(l)}, \quad \text{and} \tag{6.7}$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{b}^{(l)}} = \boldsymbol{\delta}^{(l)}\mathbf{1} \tag{6.8}$$

for hidden layers $l = 2, \ldots, L$. For the input layer $l = 1$, we compute

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}} = \boldsymbol{\delta}^{(1)}\mathbf{X}^{\mathrm{T}} + \lambda\mathbf{W}^{(1)} \quad \text{and} \tag{6.9}$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{b}^{(1)}} = \boldsymbol{\delta}^{(1)}\mathbf{1}. \tag{6.10}$$

Finally, we update the model weights and biases using Nesterov momentum updates [BBLP13]

$$\boldsymbol{\Theta}_v^{(t+1)} \triangleq \mu \boldsymbol{\Theta}_v^{(t)} - \alpha \frac{\partial \mathcal{J}}{\partial \boldsymbol{\Theta}^{(t)}}, \tag{6.11}$$

$$\boldsymbol{\Theta}^{(t+1)} \triangleq \boldsymbol{\Theta}^{(t)} - \mu \boldsymbol{\Theta}_v^{(t)} + (1+\mu)\boldsymbol{\Theta}_v^{(t+1)} \tag{6.12}$$

for parameters $\boldsymbol{\Theta} \in \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L+1)}, \mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(L+1)}\}$, successive time steps $t = 1, \ldots, T$, and constants μ and α which are the momentum and learning rates, respectively.

Once these parameters are learned and the neural network is used to infer a prediction value from some observed data, only the forward-propagation steps are taken and the parameters are no longer updated.

In practice, we do not use all training examples for each parameter update iteration or even during inference, but rather a random subset referred to as a "mini-batch," which we denote $\mathbf{X}_{\text{batch}} \in \mathbb{R}^{N \times M_{\text{batch}}}$ and the corresponding labels $\mathbf{y}_{\text{batch}} \in \{-1, +1\}^{M_{\text{batch}}}$. In our protocol, $\mathbf{X}_{\text{batch}}$ and $\mathbf{y}_{\text{batch}}$ are sampled randomly without replacement until all training examples have been considered, after which we restore all training data and repeat. We use these randomly sampled mini-batches to iteratively compute the unbiased estimate of the gradient and update our model parameters, a procedure referred to as stochastic gradient descent (SGD).

6.4 Interlude: Adding security

One of the foremost advantages of our neural network is that it can be efficiently implemented in a "secure computation" framework which enables the training data information to be completely hidden from the computing parties that perform the training protocol. While security is not the main point of this chapter, it is still an important aspect of our DTI prediction efforts and we therefore devote a brief mention of it in this section and leave additional details to Appendix C.

Secure multiparty computation (MPC) protocols [CDN15] from modern cryptog-

raphy allow multiple entities to compute over their private datasets without revealing any information about the underlying raw data, except for the final computational output. Unfortunately, existing secure computation frameworks typically have trouble scaling to complex computations over large datasets, e.g., training a complex model over a large amount of experimental data to predict new therapeutic interactions.

The scalability of our neural network approach to DTI prediction enables a secure MPC version of the algorithm. Conceptually, our protocol divides computation across collaborating entities while ensuring that none of the entities has any knowledge about the private data (Figure A-17). We achieve this using a cryptographic framework known as secret sharing [BOGW88] in which a private value ("secret") is collectively represented by multiple entities. Each entity is given a random number ("share") in a finite field (i.e., integers modulo some prime number p) such that the sum of all shares modulo p equals the secret. Any strict subset of entities cannot extract any information about the underlying secret using their shares. Various protocols have been developed for performing elementary operations (e.g., addition or multiplication) over secret-shared inputs [BOGW88, CS10], which taken together form the building blocks for a general purpose MPC framework.

Although secret sharing-based MPC typically requires overwhelming amounts of data communication between entities for complex and large-scale computations, very recent optimizations have leveraged techniques such as generalized Beaver triples and shared pseudorandom number generators to significantly reduce communication cost, enabling practical secure computation for challenging problems such as genome-wide association studies for a million individuals [CWB18]. Even with these advances, however, secure MPC is still infeasible for existing DTI prediction methods [NIW+13] primarily because their computations scale quadratically with the number of drugs and the number of targets in the dataset, which is prohibitive for pooled datasets with millions of compounds.

We circumvent the quadratic complexity of existing methods by training our neural network over a dataset consisting of only the observed DTIs and a comparable number of putatively non-interacting drug-target pairs, which is typically vastly smaller than

the full drug-by-target matrix. Furthermore, we greatly reduce the cryptographic overhead of secure neural network training by optimizing our architectural choices for efficient MPC, such as using the rectifier [GBB11] as our activation function and hinge loss as our loss function, both of which require only a single data-oblivious comparison to evaluate. These operations can be more efficiently implemented in MPC than alternatives such as the sigmoid function, which requires many such comparisons to accurately approximate. Full MPC training details are provided in Appendix C.1. Taken together, our efficient protocol allows our neural network to securely train over a wide area network (WAN) in under four days on a dataset with more than a million training instances. In contrast, a recently proposed protocol for privacy-preserving neural network training [MZ17] requires months of communication time over a WAN to train on an image dataset of smaller scale (60K examples, 784 features).

We wanted to develop a secure version of our DTI neural network since, until now, collaboration among pharmaceutical companies and academic labs, including open-access data sharing partnerships like the Structural Genomics Consortium, have been of limited scope because pharmacological data sharing is fundamentally restricted by concerns about intellectual property and other financial interests. Currently, entities have to moderate the amount of data they share in order to maintain the confidentiality of drugs under development or the set of potential targets being tested, both of which may contain sensitive information about underlying research or business strategies. Instead, our approach enables data sharing that mitigates the risk of leaking confidential information.

Because the secure version of our neural network introduces just a tolerable amount of cryptographic overhead that is not too different from the plaintext version of the neural network, our results are reported for the secure version and we refer to our neural network DTI prediction method as Secure-DTI.

6.5 Cross-validation: Advancing the state-of-the-art

6.5.1 DrugBank dataset

We wanted to compare the accuracy of our securely trained neural network for DTI prediction (Secure-DTI) to state-of-the-art DTI prediction techniques, including those based on matrix factorization with side information (CMF) [ZDMZ13], network diffusion (NetLapRLS, BLMNII, HNM) [XWZW10, MKY+13, WYZL14], and heterogeneous data integration (DTINet) [LZZ+17] on DrugBank 3.0, a standard benchmark dataset [KLJ+11] with 708 drugs, 1,512 targets and 1,923 interactions.

DrugBank includes structure information for its chemicals, including a representation specified by the simplified molecular-input line-entry system (SMILES) [Wei88]. We used JChem Base (version 17.28.0, 2017, ChemAxon, http://www.chemaxon.com) to convert SMILES to an extended connectivity fingerprint with diameter 4 (ECFP4) [RH10], which hashes SMILES to a bit vector in $\{0,1\}^{1,024}$. Each protein in DrugBank had an associated Ensembl [YAA+16] protein identifier, which we used to query the Ensembl protein dataset via the Ensembl REST server for the Pfam domain families [FCE+16] associated with each protein. For DrugBank 3.0, we observed 1,129 unique Pfam families, resulting in a feature vector in $\{0,1\}^{2,153}$. Our method can easily incorporate alternative feature representations of drugs and targets. We expect the precise feature set to be determined by the collaborating entities given a specific study setting.

On the DrugBank 3.0 dataset, we compared our secure neural network model to existing DTI prediction methods as reported in Luo et al. [LZZ+17], which introduces the DTINet model and compares it to previous state-of-the-art methods BLMNII, NetLapRLS, HNM, MF, and CMF. The authors of the study tested their methods using 10-fold cross validation (CV) on a balanced test set, an imbalanced test set with a 1:10 ratio of positive to negative examples, and the entire drug-target interaction space. As in Luo et al., we used two metrics to evaluate classification accuracy. The

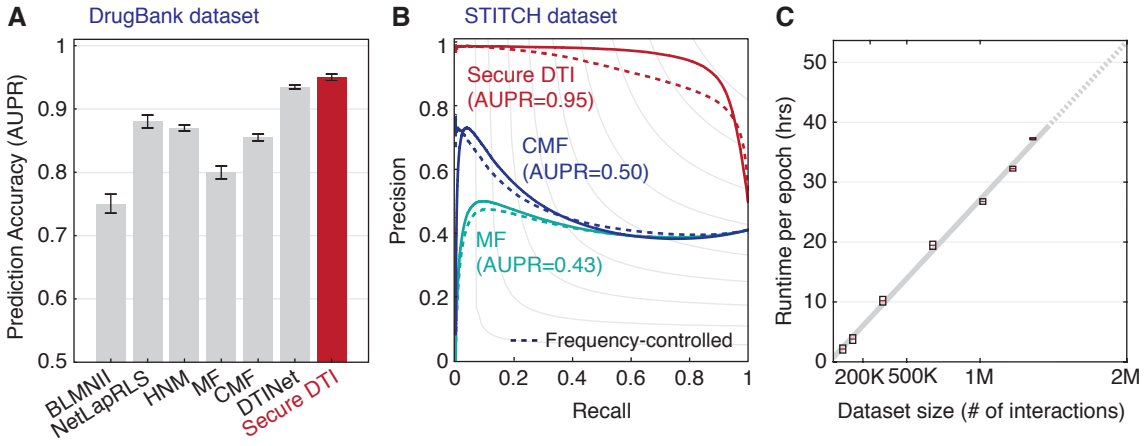

Figure 6-1: Prediction of DTIs.

(**A**) Predictions from the DrugBank 3.0 dataset. Bar height corresponds to mean AUPR (area under the precision-recall curve), and error bars indicate SD. We compared Secure DTI to the plaintext methods BLMNII, NetLapRLS, HNM, MF, CMF, and DTINet, as reported in Luo et al. (15), by means of 10-fold cross-validation on balanced training and test sets. (**B**) Predictions form the STITCH 5 dataset with more than 1 million drug–target pairs. Secure DTI is compared with matrix factorization with (CMF) and without (MF) side information (see Figure A-18 for other evaluation settings). Solid lines, sampling negative examples randomly; dashed lines, sampling negative examples while matching the relative frequencies of drugs and targets to those in the positive examples, representing a more challenging test case. Reported AUPRs are for the solid curves. (**C**) Runtime of our training protocol, over a local area network (LAN), for different dataset sizes. Box height represents SD.

first is the receiver-operating characteristic (ROC) curve, which plots the true positive rate and the false positive rate at various thresholds for the predicted scores. To summarize the ROC curve in a single statistic, we take the area under the ROC curve (AUROC) using the trapezoidal rule. We also used the precision-recall curve, which plots the precision and recall at various scoring thresholds. We use the area under the precision-recall curve (AUPR) as our precision-recall summary statistic [BEP13].

Secure-DTI surpasses the performance of all baseline methods in cross validation accuracy (Figure 6-1A), a surprising result in light of the optimizations we made to achieve practical scalability. Our improvement over the best-performing baseline (DTINet) is statistically significant (one-sided Wilcoxon rank-sum P-value of 0.006).

6.5.2 STITCH dataset

We next set out to demonstrate the scalability and predictive performance of Secure-DTI on a much larger dataset that more accurately represents the scale of cross-institutional collaboration. We obtained 969,817 interactions from the STITCH 5 human dataset [SSV+16], to our knowledge the largest publicly available DTI dataset. Chemicals in STITCH were featurized based on ECFP4s using the same procedure for DrugBank. For the STITCH target features, we observed 5,879 unique Pfam families, which we encoded as a bit vector in $\{0,1\}^{5,879}$ where a 1 at a position in the vector indicates that a given protein sequence contained the Pfam family associated with that position. The chemical and protein features were concatenated to produce a feature vector in $\{0,1\}^{6,903}$.

As described in the main text, out of the 969,817 total interactions we randomly chose 290,946 interactions (~30%) as a held-out test set, and further divided the remaining interactions into a training set of 484,908 interactions (~50%) and a validation set of 193,963 interactions (~20%), which we used to tune model parameters. We compared the performance of our secure neural network model to collaborative matrix factorization (CMF) and matrix factorization without side information (MF), both of which were implemented in plaintext.

Other baseline methods that we considered for DrugBank could not be applied to the STITCH dataset due to their lack of scalability. In particular, with hundreds of thousands of chemicals represented in STITCH, the drug-by-drug similarity matrix required by these methods creates a significant computational bottleneck. Furthermore, most of the drug data used in DTINet is unavailable for a much larger set of chemicals and proteins in STITCH 5. Although CMF also requires the similarity matrix, we were able to train the model in plaintext (albeit inefficiently) by recomputing only small portions of the similarity matrix, rather than storing the entire matrix in memory or on disk.

We trained the secure neural network, CMF, and MF on training and validation sets where the number of negative interaction examples was the same as the number of

positive interactions. We assessed cross-validated model performance in four different evaluation settings. First, we chose a random subset of interactions as a positive test set and sampled an equal number of negative pairs (a balanced test set). Second, we sampled ten-times the number of negative pairs compared to positive DTIs (an imbalanced test set). Third, we separated out a group of 131,767 chemicals involved in ~30% of all interactions, which we used as positive test examples, and used the remaining interactions as training data. We again used an equal number of random pairs as negative examples but enforced the division of chemicals between training and testing sets (a divided-chemical test set). Fourth, we repeated the divided-chemical experiment but forcing the negative training examples to have an equal representation of drugs and targets as the positive training examples by using random edge-swaps of the drug-target interaction graph.

Model training was done using secure MPC and the resulting model parameters were used to make predictions in the plaintext setting, equivalent to the setting when collaborating entities reveal the model as the final output of the secure protocol. Classification accuracy was measured using the AUROC and AUPR statistics.

We evaluated the cross validation performance of Secure-DTI on STITCH 5. Even on the challenging task of predicting DTIs of previously unseen compounds, Secure-DTI achieved high accuracy (AUPR of 0.95), which substantially outperforms matrix factorization methods (AUPRs of 0.50 and 0.43; Figure 6-1B and A-18). Other baseline methods could not be reasonably applied to a dataset of this size (even in plaintext) due to their quadratic scalability.

In contrast, Secure-DTI took less than four days to train on millions of interactions over a WAN and efficiently scaled with a linear dependence on the number of interactions in the dataset (Figure 6-1C). Even training on two million interactions, we extrapolate the total runtime for one epoch (one linear pass over the full, shuffled training set) to be around 2.2 days.

6.6 Experimental validation: Room for improvement

We wanted to go beyond cross validation and demonstrate the potential for novel discoveries that can result from our machine learning algorithm. We therefore trained Secure-DTI on all STITCH 5 interactions and scored the remaining possible drug-target pairs for interactivity, which is closer to how our pipeline would be used in a real-world setting. We controlled for bias toward highly represented drugs and targets in the dataset by either (i) filtering out any prediction involving both a drug and target highly represented in the original dataset (Secure-DTI-A) or (ii) sampling negative examples (i.e., non-interactions) during model training such that each drug or target was seen at the same relative frequency in the negative examples as in the positive examples (Secure-DTI-B) (Figure 6-1B).

In both cases, many of our top predictions (5/12 for Secure-DTI-A and 9/12 for Secure-DTI-B) were validated by our own targeted assay experiments (see Appendix C.2 for experimental validation details) or by published experimental studies that have not yet been deposited into the STITCH database (Table 6.1). Our validation experiments suggest a novel interaction between imatinib and ErbB4, for which we could not find any existing experimental support. It will be interesting to see if this interaction is confirmed by other studies.

The top prediction from both methods was an interaction between the estrogen receptor (ER) and droloxifene, which had reached phase III clinical trials as an ER modulator for advanced breast cancer [BHEK+02]. Similarly, the predicted interaction between the vitamin D receptor (VDR) and seocalcitol has been clinically well-established [TVRWM04]. Furthermore, some predictions without direct activity have strong evidence for an indirect functional interaction; for example, nutlin-3 has been shown to inhibit PARP1 protein levels through p53-dependent proteasomal degradation in mouse fibroblasts [MOO+11].

Rank	Drug	Target	Validation	Drug	Target	Validation
1	Droloxifene	ERα	Active*	Droloxifene	ERβ	Active*
2	Droloxifene	ERβ	Active*	CHEMBL601690	p110α	Active
3	Imatinib	ErbB3	Inconclusive*	Droloxifene	ERα	Active*
4	Imatinib	ErbB4	Active*	Seocalcitol	VDR	Active
5	Nutlin-3	PARP1	Inactive*	AGN-PC-0A9TBG	PPARγ	Active
6	Droloxifene	PgR	Inactive*	CHEMBL589864	p110α	Active
7	Actinomycin D	PARP1	Weakly active*	T5958429	PARP1	Active
8	Hoechst 33258	PARP1	Inactive*	AGN-PC-0N7PYE	Factor Xa	Active
9	GW-501516	GR	Inactive*	AGN-PC-00DJ3O	PPARγ	Active
10	AGN-PC-0BFP0W	Lck	Active	AGN-PC-0NA8NJ	PTPRZ1	N/A
11	CHEMBL2332055	mGluR1	Inconclusive*	AGN-PC-0NA8NJ	PTPRG	N/A
12	CHEMBL2332055	mGluR5	Inconclusive*	AGN-PC-088DZ9	PROC	N/A

Table 6.1: Predicted out-of-dataset drug-target interactions.

We trained Secure DTI on all human drug-target interactions from STITCH 5, which we used to score and rank all pairs of drugs and targets that are not in the STITCH database. We implemented two methods to control for model bias toward overrepresented drugs and targets, either (left) filtering out predictions involving a drug and target that are both highly represented in STITCH (columns 2 through 4) or (right) retraining Secure DTI such that the negative training examples had an equal representation of drugs and targets as the positive training examples (columns 5 through 7). An asterisk (*) indicates predicted interactions that were experimentally validated, including all testable interactions without existing literature support. Interactions labeled N/A involve commercially unavailable compounds and so could not be tested. We labeled an interaction as "active" if its IC50 was less than 100 μM and "inconclusive" if activity was observed but only at one or two high concentration levels, a potential artifact of compound aggregation. We labeled the interaction between Actinomycin D and PARP1 as "weakly active" as consistent activity was observed over a wide range of concentrations, including close to 50% inhibition at 100 μM (our highest tested concentration), but it should be noted its dose-response curve does not follow a typical sigmoidal shape.

The ability to discover new DTIs was an exciting result, especially given the efficiency (even with cryptographic overhead) of our learning algorithm. Still, we realized that there was room for improvement when applying machine learning to DTI prediction in a drug discovery context. First, the false positive rate (10 out of 24 pairs tested, or ~42%) was higher than that suggested by our cross-validation experiments (~10%). Second, many of the validated interactions were relatively weak; for example, the interaction between imatinib and ErbB4 had a dissociation constant in the micromolar range, whereas kinase inhibitors are typically considered successful when their activity in the the *nano*molar range, up to three orders of magnitude more potent than what we observed.

Reflecting on these results, we realized there were computational remedies for these issues. We hypothesized that the increase in false positives may be a result of the test dataset being different from the training dataset, leading the model to be confused on test examples that were unlike anything it had seen before. Our feature representations based on ECFP4s and one-hot-encoded Pfam domains were also relatively simplistic and discarded a lot information about both the chemical structure and the target protein. And because we only trained a binary classifier, our model does not distinguish a weak interaction from a potent one.

Therefore, while Secure-DTI was a good first step to learning-based DTI prediction, we knew that there were many ways to improve our ability to discover new biology. Those efforts are described in the following chapter.

Chapter 7

Fighting Disease II: Uncertainty

There are known knowns: there are things we know we know. We also know there are known unknowns, that is to say we know there are some things we do not know. But there are also unknown unknowns: the ones we don't know we don't know.

—Donald Rumsfeld, press conference (2002)

In the last chapter, we laid out an initial attempt to use machine learning algorithms to predict how well a drug and target interact, a common problem in the drug discovery process. In this chapter, we again focus on this discovery process, but also on improving the quality of our algorithm's predictions where the ultimate test of performance is not cross validation but actually discovering novel interactions.

A very common problem when using a machine learning model to make new discoveries is that the additional data that you provide to the model may be very different from the training data, i.e., the test data comes from a different distribution. When encountering data unlike anything it has seen before, a standard machine learning model could become confused and output a nonsensical prediction. Moreover, because a standard algorithm does not report how certain or uncertain it is about a given prediction, we have no way of knowing if the model is operating in the well defined in-distribution setting or in the poorly defined out-of-distribution setting.

This chapter describes our attempts at improving biochemical activity prediction, with a particular focus on designing a machine learning algorithm that knows when it is *uncertain*. We test different ways of quantifying uncertainty and find good uncertainty prediction from a class of Bayesian machine learning algorithms called *Gaussian processes*. Not only do we achieve state-of-the-art cross validation performance, but we also achieve strong experimental validation (and discover new nanomolar-range kinase inhibitors) even when the test data comes from a different distribution than the training data. We end with an application note where we build on the discoveries made by our algorithm and find new growth inhibitors of *Mycobacterium tuberculosis*[1].

7.1 Glossary

- *Epistemic uncertainty*: Uncertainty that arises from a lack of knowledge and therefore challenging to quantify; often contrasted with aleatoric uncertainty due to statistical variability, which is easier to quantify.

- *Sample efficiency*: In machine learning, the ease with which a model adapts its beliefs based on new data.

- *Unsupervised pretraining*: A machine learning strategy in which, first, an unsupervised learning model extracts features from a large dataset and, then, those features are used as input to a supervised model; often, the unsupervised model leverages large neural architectures.

- *Gaussian process (GP) regressor*: A supervised learning algorithm that, given a test example x, outputs a prediction in the form of a Gaussian distribution that resembles the distribution of the training labels of points close to x or, if x is far away from any training points, outputs a Gaussian that resembles a prior distribution; both the notion of distance to the training set and the prior distribution are GP parameters (specified by a user or learned from data).

[1]Software for the work described in this chapter is available at `https://github.com/brianhie/uncertainty`.

- *Dissociation constant (Kd)*: Measures the binding strength between two molecules in a chemical reaction in units of concentration, where smaller values indicates higher binding affinity.

7.2 Preliminaries

7.2.1 Uncertainty

When leveraging existing data to test new hypotheses, biologists often find themselves in what machine learning researchers refer to as an "out-of-distribution" paradigm. For example, a researcher may be interested in finding a small molecule that inhibits a kinase, a problem of biochemical and pharmacological importance. The researcher may have existing data (for example, from a high throughput screen) or have domain expertise (for example, knowledge of previously successful inhibitors). When searching for new inhibitors, some chemical structures might be similar to well-studied structures, and therefore might also have similar behavior. However, there is an enormous space of chemical structures with uncertain or unknown biochemistry. While notions of biochemical "similarity" or "uncertainty" might be obvious to a human expert, a standard machine learning algorithm has no corresponding notion of uncertainty. Similar problems exist in many other biological settings as well.

In the "in-distribution" paradigm, which is what most machine learning methods assume, the test data comes from the same data distribution as the training data. However, when the test data does not come from the same distribution, i.e., it is "out-of-distribution," a fundamental assumption underlying most methods breaks (Figure 7-1A,B). Most modern models, including state-of-the-art "deep learning" methods, are susceptible to bias, overconfidence, and other pathologies when making out-of-distribution predictions [NYC15, AOS$^+$16, AOS$^+$16, CJS18, GPSW17, LPB17]. As a result, some researchers manually or heuristically remove out-of-distribution prediction examples, but these approaches fail to address the root of the problem, which is a fundamental limitation of the learning algorithms themselves.

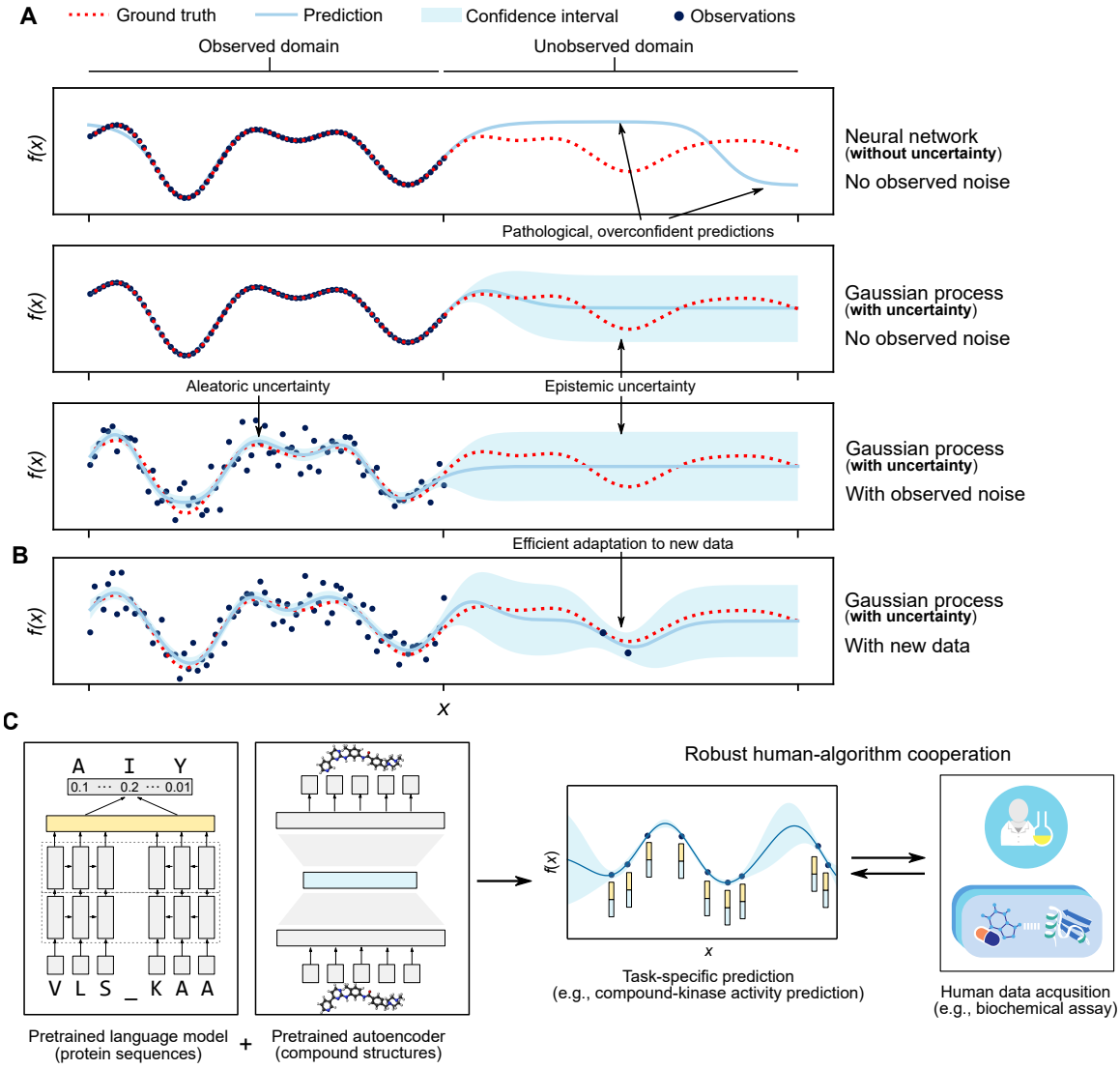

Figure 7-1: Robust uncertainty prediction for machine-guided discovery.

(**A**) When a machine learning model encounters an example like nothing in its training set, its behavior is usually undefined. A way to improve robustness is for the model to report high uncertainty on such examples. Rather than output a single point prediction for each example in a given domain, more robust methods, such as a Gaussian process (GP), model the aleatoric (or statistical) uncertainty of observations and the epistemic (or systematic) uncertainty that comes from a lack of data. In a GP, the epistemic uncertainty of unexplored regions of the domain is explicitly encoded as a prior probability. (**B**) GPs can readily update their beliefs with just a handful of new data points. (**C**) Using modern, neural pre-trained feature representations, a GP can achieve state-of-the-art prediction performance even with limited data. Knowing uncertainty helps guide a researcher when prioritizing experiments and, when combined with sample efficiency, enables a tight feedback loop between human data acquisition and algorithmic prediction.

Instead, it is also possible for a machine learning model to output both a prediction and an associated confidence score, where in-distribution predictions are assigned high confidence and out-of-distribution predictions are assigned high uncertainty (Figure 7-1A,B) [BS09, GPSW17, Nea12]. A principled approach to uncertainty prediction is based on Bayesian statistics. A Bayesian machine learning model will output a probability distribution for each prediction (for example, a Gaussian distribution), rather than a single point estimate (Figure 7-1C). A location-related summary statistic of the distribution, like the mean or the median, can then be used as the prediction value; a dispersion-related summary statistic, like the variance or standard deviation, can be used as the uncertainty score. Importantly, a user can specify a *prior distribution* with high uncertainty such that, if a model has little knowledge about a given training example, the model prediction will be close to the prior.

Types of uncertainty

The kind of uncertainty associated with novel biological discovery is typically called *epistemic*, or systematic, uncertainty (Figure 7-1A). Epistemic uncertainty, as its Greek root suggests, is due to a lack of knowledge; in the machine learning setting, epistemic uncertainty arises due to a lack of training data. This is the kind of uncertainty captured by the prior distribution in the Bayesian machine learning setting mentioned above. Another kind of uncertainty is called *aleatoric*, or statistical, uncertainty (Figure 7-1B). Aleatoric uncertainty occurs when repeated experiments are run, each producing different results. This kind of uncertainty is also an important consideration and can be learned from data using standard statistical approaches.

7.2.2 Sample efficiency

Another important concept in this chapter, "sample efficiency" (Figure 7-1B) [GWH14, MRDJ17], is the ability to make use of and quickly adapt to new data. In contrast, providing a small amount of new data to a sample inefficient algorithm would not substantially change its predictions. Sample efficiency is especially critical in domains

where new data collection is limited or slow (for example, synthesizing and testing novel customized small-molecule drugs [LBB18]). A typical criticism of modern deep learning methods is that they are not sample efficient, but instead require many training examples to achieve reasonable performance [Ng, GBC16]. In contrast, humans can learn from only a few examples; for example, after a single instance of touching a hot stove, a human typically does not need any additional information to exercise greater caution in the future.

7.2.3 Pretraining

Another import concept is the notion of "pretraining" [EBC+10]. Pretraining automatically extracts relevant, general features in a task-agnostic, or an unsupervised, way; machine learning models can subsequently leverage pretrained features on a variety of more specific downstream tasks. In the kinase activity prediction setting, pretrained features might be extracted from training an unsupervised algorithm on a large database of small molecules or on a large corpus of protein sequences (Figure 7-1C). These compound and protein features can then be used as input to a supervised machine learning algorithm that learns the specific task of predicting compound-kinase interactions.

7.2.4 Review of Gaussian process regression

GPs are prime candidates for machine learning-based hypothesis generation since they naturally quantify prediction uncertainty [RW05], are highly sample-efficient [GWH14], and can readily incorporate a rich set of features like those obtained by pretraining. GPs allow a researcher to specify a prior distribution encoding high epistemic uncertainty when the training distribution provides little information on unseen test examples (Figure 7-1) [BS09, MRDJ17]. As datapoints become more distal to the training set, GP uncertainty also grows to approach the prior uncertainty, analogous to human uncertainty increasing on examples that deviate from existing knowledge (Figure 7-1).

Below, we go into more depth into how Gaussian process regression works, since we rely heavily on it throughout this chapter. Readers who already have a good appreciation for how GP regression works, or those comfortable with the high-level intuition, can feel free to skip ahead.

To begin, first consider an M-dimensional multivariate Gaussian $\mathbf{g} \sim \text{Normal}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where g_i is the ith element in $\mathbf{g}$, with μ_i and σ_{ij} defined similarly for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, respectively. We will forgo a deeper review of all the (beautiful) theoretical properties of multivariate Gaussians and assume some amount of familiarity with them; they are also characterized extensively in [RW05].

We can interpret a multivariate Gaussian as a distribution over functions. Let $\mathcal{X} \triangleq \{x_1, \ldots, x_M\}$ and let $f : \mathcal{X} \to \mathbb{R}$ be a random function such that

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_M) \end{bmatrix} \sim \text{Normal}\left(\begin{bmatrix} \mu_1 \\ \vdots \\ \mu_M \end{bmatrix}, \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1M} \\ \vdots & \ddots & \vdots \\ \sigma_{M1} & \cdots & \sigma_{MM} \end{bmatrix} \right).$$

Note that $g_i = f(x_i)$ is a random variable and there is a one-to-one mapping between the values of $\mathbf{g}$ and the values of $f(\mathbf{x})$.

Gaussian processes are often thought of as an extension of the random function interpretation of a multivariate Gaussian to *infinitely* many functions. To see how, now let $\mathcal{X}$ be infinite (and potentially uncountably so, e.g., $\mathbb{R}^N$). Now, we define two important functions

$$m : \mathcal{X} \to \mathbb{R} \quad \text{and} \quad K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}.$$

If, for *any* nonempty subset $\{x_1, \ldots, x_M\} \subseteq \mathcal{X}$,

$$\begin{bmatrix} g_1 \\ \vdots \\ g_M \end{bmatrix} = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_M) \end{bmatrix} \sim \text{Normal}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_M) \end{bmatrix}, \begin{bmatrix} K(x_1, x_1) & \cdots & k(x_1, x_M) \\ \vdots & \ddots & \vdots \\ K(x_M, x_1) & \cdots & K(x_M, x_M) \end{bmatrix} \right)$$

then we can define

$$\mathbf{g} = f(\mathbf{x}) \sim \mathrm{GP}(m, K),$$

i.e., $\mathbf{g}$ is a Gaussian process (intuitively, an infinite-dimensional multivariate Gaussian).

A Gaussian process is therefore fully described by a mean function m and a covariance function K. For example, a popular covariance function is the Gaussian, or squared exponential, kernel scaled by a constant k_{prior} related to the prior uncertainty

$$K(\mathbf{x}_i, \mathbf{x}_j) = k_{\text{prior}}^2 \exp\left\{-\frac{1}{2}\gamma \|\mathbf{x}_i - \mathbf{x}_j\|_2^2\right\}$$

where $\|\cdot\|_2$ denotes the ℓ_2-distance between feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$.

To see how GPs can be used in supervised learning, consider their functional interpretation. In regression, we want to learn a function that predict labels from some input data, i.e.,

$$y^{(i)} = f(x^{(i)}) + \boldsymbol{\epsilon}^{(i)}.$$

In GP regression, f has a GP prior and $\boldsymbol{\epsilon}^{(i)} \sim \mathrm{Normal}(\mathbf{0}, \sigma^2\mathbf{I})$ is some noise perturbation.

Now, we can put this all together to derive the inference equations for GP regression. Assuming we have some training data $\mathbf{x} \triangleq \left(x^{(1)}, \ldots, x^{(M')}\right)$ and some test data $\tilde{\mathbf{x}} \triangleq \left(\tilde{x}^{(1)}, \ldots, \tilde{x}^{(M'')}\right)$. Also we define $\mathbf{y}$ as the training labels (that we know) and $\tilde{\mathbf{y}}$ as the test labels (that we do not know). We therefore want to obtain

$$p_{\tilde{\mathbf{y}}|\mathbf{y},\mathbf{x},\tilde{\mathbf{x}}}(\tilde{\mathbf{y}}|\mathbf{y}, \mathbf{x}, \tilde{\mathbf{x}}).$$

Using block matrix form to express the multivariate Gaussian defined by the GP, we know that

$$\begin{bmatrix}\mathbf{y} \\ \tilde{\mathbf{y}}\end{bmatrix}\Bigg| \mathbf{x}, \tilde{\mathbf{x}} = \begin{bmatrix}f(\mathbf{x}) \\ f(\tilde{\mathbf{x}})\end{bmatrix} + \begin{bmatrix}\boldsymbol{\epsilon} \\ \tilde{\boldsymbol{\epsilon}}\end{bmatrix}$$

is distributed as

$$\text{Normal}\left(\begin{bmatrix} m(\mathbf{x}) \\ m(\tilde{\mathbf{x}}) \end{bmatrix}, \begin{bmatrix} K(\mathbf{x},\mathbf{x}) + \sigma^2\mathbf{I} & K(\tilde{\mathbf{x}},\mathbf{x}) \\ K(\mathbf{x},\tilde{\mathbf{x}}) & K(\tilde{\mathbf{x}},\tilde{\mathbf{x}}) + \sigma^2\mathbf{I} \end{bmatrix}\right).$$

By the rules for conditioning on Gaussians, we get

$$\tilde{\mathbf{y}}|\mathbf{y},\mathbf{x},\tilde{\mathbf{x}} \sim \text{Normal}\left(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}\right)$$

where

$$\tilde{\boldsymbol{\mu}} \triangleq m(\tilde{\mathbf{x}}) + K(\tilde{\mathbf{x}},\mathbf{x})\left(K(\mathbf{x},\mathbf{x}) + \sigma^2\mathbf{I}\right)^{-1}(\mathbf{y} - m(\mathbf{x})) \quad \text{and} \tag{7.1}$$

$$\tilde{\boldsymbol{\Sigma}} \triangleq K(\tilde{\mathbf{x}},\tilde{\mathbf{x}}) + \sigma^2\mathbf{I} - K(\tilde{\mathbf{x}},\mathbf{x})(K(\mathbf{x},\mathbf{x}) + \sigma^2\mathbf{I})^{-1}K(\mathbf{x},\tilde{\mathbf{x}}). \tag{7.2}$$

This result has an elegant interpretation: the predicted mean is equal to the mean function evaluated on the test data $m(\tilde{\mathbf{x}})$ modified by additional information on how "different" the test data is from the training data. An analogous argument is also the case for the predicted covariance.

While inference is often dominated by the matrix inversion step, i.e., calculating $(K(\mathbf{x},\mathbf{x}) + \sigma^2\mathbf{I})^{-1}$, which scales cubically in the size of the training data, in practice, there has been a large amount of work on exploiting sparsity to accelerate exact inference or on doing approximate inference with good empirical performance. These approaches have enabled GPs to scale to datasets with billions of training examples [LOSC20].

7.3 Cross validation: Uncertainty redux

7.3.1 Setup

As a test case for machine-guided discovery, we predicted binding affinities between small molecule compounds and protein kinases. We select this particular application

since kinases have diverse pharmacological implications that include infectious disease therapeutics [ASP+14, LOS+11, WWS+09, WID09] and comprehensive compound-kinase affinity training data exists for a limited number of compounds [DHH+11].

We first set up an *in silico* simulation of the prediction and discovery process. We obtained a publicly-available dataset [DHH+11] containing binding affinity measurements, within a 0.1 to 10,000 nanomolar (nM) range, of the complete set of kinase-compound pairs among 72 compounds and 442 unique kinase proteins (the dataset contained 379 unique kinase genes with multiple mutational variants for some of the genes). We set up a cross-validation-based simulation by separating the known data into training and test data (Figure A-19A). To simulate out-of-distribution prediction, we ensured that approximately one-third of the test data contained interactions involving compounds not in the training data, one-third contained interactions involving kinase genes not in the training data, and one-third contained interactions involving compounds and kinase genes not in the training data (Figure A-19A).

Our main set of benchmarking methods leverages unsupervised pretraining via state-of-the-art neural graph convolutional-based compound features (pretrained by the original study authors on ~250K small molecule structures) [JBJ18] and neural language model-based protein sequence features (pretrained by the original study authors on ~21M protein sequences) [BB19]. Subsequent regression models use a concatenation of these features to predict Kd binding affinities (Figure 7-2A).

7.3.2 Benchmark methods

We benchmark three baseline methods without uncertainty:

Multilayer perceptron (MLP)

Also known as a densely-connected neural network [HCB18, ÖÖO18], we test an MLP architecture similar to that described in Chapter 6, but trained with mean square error loss since we are performing regression rather than classification.

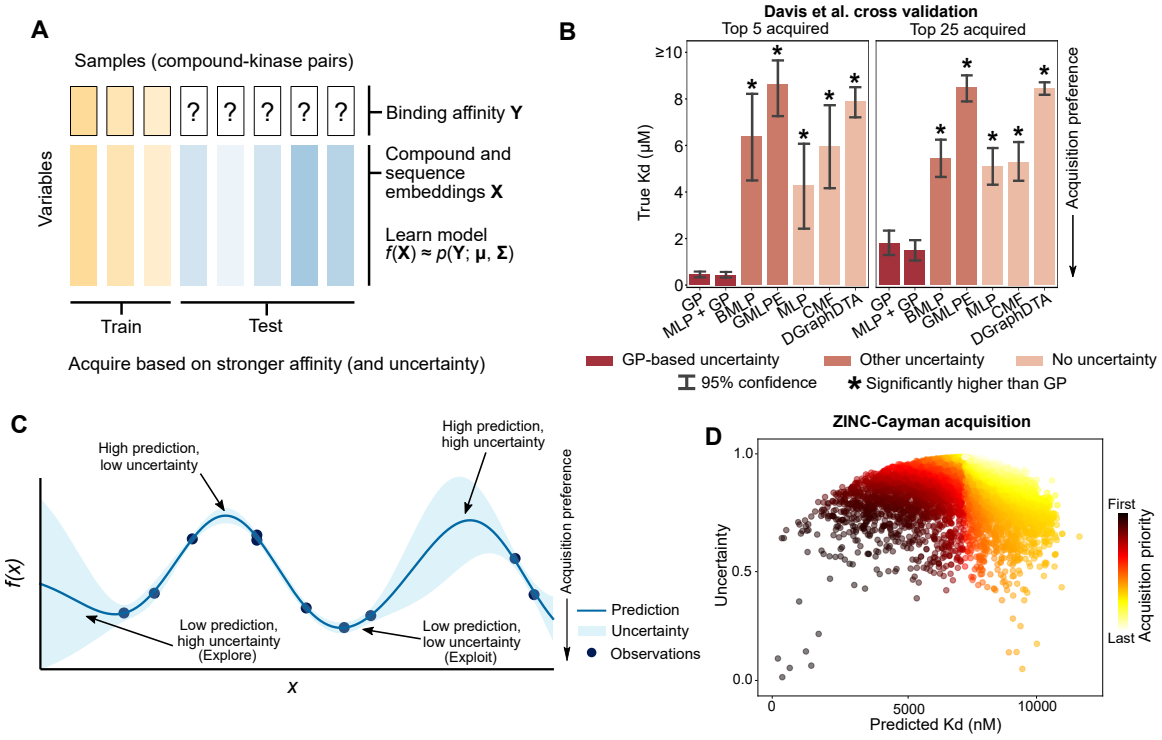

Figure 7-2: Computational prediction of compound-kinase affinity.

(**A**) We desire to predict compound-kinase affinity based on features derived from compound structure and kinase sequence and use these predictions to acquire new interactions. Incorporating uncertainty into predictions is especially useful when the data distributions of the training and test sets are not guaranteed to be the same. (**B**) True Kds of the top five and twenty-five prioritized compound-kinase pairs for each model over five model initialization random seeds. Bar height indicates mean Kd; statistical significance was assessed with a one-sided Welch's t-test P-value at FDR < 0.05. (**C**) Predictions augmented with uncertainty scores enable a researcher to perform experiments in high confidence, high desirability regions ("exploitation") or to probe potentially high desirability regions with less model confidence ("exploration"). (**D**) Each point represents a compound in the ZINC-Cayman library (Table A.8) with an associated predicted Kd (with PknB) and uncertainty score outputted by a GP (normalized by the prior uncertainty), colored by the order the compound appears according to our acquisition function. We use an acquisition function that prioritizes high confidence, low Kd predictions.

Collective matrix factorization (CMF)

We performed CMF, which we review in Section 7.3.2 in the previous chapter, using the compound-kinase Kds as the explicit data matrix and the neural-encoded compound and kinase features as side-information [SG08, ZDMZ13, CLH+13]. Briefly, the CMF loss function used here is

$$\mathcal{L}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}; \mathbf{M}, \mathbf{X}_1, \mathbf{X}_2, \lambda_1, \lambda_2) \triangleq \|\mathbf{M} - \mathbf{A}\mathbf{B}^{\mathrm{T}}\|_{\mathrm{F}}^2 + \lambda_1 \|\mathbf{X}_1 - \mathbf{A}\mathbf{C}^{\mathrm{T}}\|_{\mathrm{F}}^2 + \lambda_2 \|\mathbf{X}_2 - \mathbf{B}\mathbf{D}^{\mathrm{T}}\|_{\mathrm{F}}^2$$

with respect to latent variable matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$. $\mathbf{M}$ is the compound-by-kinase binding affinity matrix; $\mathbf{X}_1$ is a side-information matrix where each row contains compound features; $\mathbf{X}_2$ is a side-information matrix where each row contains kinase features; $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm of a matrix; and λ_1 and λ_2 are user-specified optimization constants (we set these values to the default value of 1, but observed that cross-validated performance metrics were robust to changes in this parameter). The number of components (i.e., the number of columns in $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$) was set to the default value of 30, but we also noticed robustness of cross-validated metrics to changes in this parameter. The CMF objective was fit using the limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS) via the cmfrec Python package version 0.5.3 (Cortes, 2018) (`https://cmfrec.readthedocs.io/en/latest/`).

DGraphDTA

We also used DGraphDTA [JLZ+20] to predict compound-kinase Kds. DGraphDTA leverages a graph neural network based on the compound molecular structure and the protein residue contact map. We wanted to benchmark against DGraphDTA to assess the benefit of our unsupervised pretraining-based features, since DGraphDTA trains a graph convolutional neural network designed specifically to train = on a simpler set of features. We used the implementation provided at `https://github.com/595693085/DGraphDTA` with default model architecture hyperparameters. For compound features, we provided the model with chemical SMILE strings [Wei88] that

the model transforms into a graph convolutional representation; for kinase features, we use the protein contact maps provided by the original study.

We also benchmark methods that learn some notion of prediction uncertainty:

Gaussian process (GP)

Our first uncertainty model fits a GP regressor [RW05] to the training set. The GP provides a Kd prediction in the form of a Gaussian distribution, where we use the mean of the Gaussian as the prediction value and the standard deviation as the measure of uncertainty. For the kinase experiments, k_{prior} is set to 10,000 nM; for the protein fluorescence experiments. Each prediction takes the form of a (scalar) Gaussian distribution; we use the mean as the prediction value and the variance as the uncertainty estimate. We use the Gaussian process regressor implementation provided by the scikit-learn Python package.

Gaussian process fit to residuals of a multilayer perceptron (MLP + GP)

Since much of the interest in machine learning has been on improving the performance of neural network models, a simple way to augment neural networks with uncertainty is to combine the predictions made by a neural network and predictions made by a GP [QMM20]. We use an MLP regressor with the same architecture and hyperparameters as the standalone MLP model described above. The GP fit to the residuals of the MLP regressor has the same form as described for the regular GP above but where the regression problem is formulated as

$$y_i - \text{MLP}(\mathbf{x}_i) \sim \text{GP}(\mathbf{x}_i)$$

for training example $\mathbf{x}_i$ and training label y_i. To calculate the prediction value, we evaluate both the MLP and the GP and sum the MLP prediction and the GP mean [QMM20], i.e.,

$$y_{\text{pred}}^{(i)} = \text{MLP}(\tilde{\mathbf{x}}_i) + \mathbb{E}\left[\text{GP}(\tilde{\mathbf{x}}_i)\right].$$

To calculate the uncertainty estimate, we can simply use the GP standard deviation, i.e.,

$$\sigma_{\text{pred}}^{(i)} = \left(\text{Var}\left(\text{GP}(\tilde{\mathbf{x}}_i)\right)\right)^{1/2}.$$

We used the same software (a combination of the scikit-learn, GPyTorch, keras, and tensorflow Python packages) to implement the hybrid model.

Bayesian multilayer perceptron (BMLP)

A more involved, Bayesian approach to augmenting neural networks with uncertainty is to impose a Bayesian prior on the parameters of the neural network. We train an MLP regressor with the same architecture described above (two hidden layers with 200 neurons per layer and ReLU non linearities) but with a unit-variance Gaussian prior on each weight and bias entry [Nea12]. Within the respective biological task, the Gaussian prior mean for each entry corresponds to a Kd of 10,000 nM (i.e., no biochemical affinity) or a log-fluorescence of 3 (i.e., a dark protein). Optimization was performed under a mean-field independence assumption with gradient descent-based variational inference [TKD$^+$16]. When making predictions, we sample 100 neural networks and evaluate each neural network on each prediction example. We use the mean prediction across the 100 neural networks as the prediction value and the variance across the 100 neural networks as the uncertainty estimate. To implement the BMLP, we used the Edward Python package (version 1.3.5) for probabilistic programming [TKD$^+$16] with a tensorflow CPU (version 1.5.1) backend.

Gaussian negative log-likelihood-trained MLP ensemble (GMLPE)

Rather than a Bayesian approach to uncertainty, another group of uncertainty methods is based on model ensembles. Ensembling involves fitting multiple models to a training dataset; then, variation in the predictions of the models can be used to estimate uncertainty. For our ensemble method, we use the model described by Lakshminarayanan et al. [LPB17]. We train an MLP regressor with the same architecture described above (two hidden layers with 200 neurons per layer and ReLU non linearities) but, instead

of mean square error loss, with Gaussian negative log-likelihood loss

$$\mathcal{L}\left(y_{\text{pred}}^{(i)}, \sigma_{\text{pred}}^{(i)}; y_{\text{true}}^{(i)}\Big|_{i=1}^{N}\right) \triangleq \sum_{i=1}^{N}\left(\log\left(\left(\sigma_{\text{pred}}^{(i)}\right)^2\right) + \frac{\left(y_{\text{true}}^{(i)} - y_{\text{pred}}^{(i)}\right)^2}{\left(\sigma_{\text{pred}}^{(i)}\right)^2}\right)$$

where $y_{\text{pred}}^{(i)}$ is the predicted value and $\sigma_{\text{pred}}^{(i)}$ is the predicted uncertainty (both outputted by the neural network), and $y_{\text{true}}^{(i)}$ is the ground truth value for training example $i \in \{1, 2, \ldots, N\}$. We train five such models to create a neural network ensemble and we combine prediction distributions across the ensemble as with a Gaussian mixture. As an implementation detail, we trained the neural network to output the log variance to enforce positivity. We implemented the GMLPE with the keras Python package using a tensorflow backend with CUDA-based GPU acceleration.

Acquisition function

For models that output uncertainty scores, an acquisition function is used to rank compound-kinase pairs for acquisition, which in the biological setting often corresponds to further experimental validation, in a way that balances both the prediction value and the associated uncertainty.

A standard acquisition function is the upper confidence bound (UCB). When low prediction values are desirable, UCB acquisition takes the form

$$a_{\text{UCB}}(i) \triangleq y_{\text{pred}}^{(i)} + \beta\left(\sigma_{\text{pred}}^2\right)^{(i)},$$

where $y_{\text{pred}}^{(i)}$ and $\left(\sigma_{\text{pred}}^2\right)^{(i)}$ are the predicted Kd and the uncertainty score, respectively, for the ith training example and where β is a parameter controlling the weight assigned to the uncertainty score. An acquisition function with a high β prioritizes low uncertainty; in contrast, a low β deprioritizes uncertainty, and $\beta = 0$ ignores uncertainty.

In practice, we use a rank-based modification to the above UCB function, which

we call rank-UCB, with the form

$$a(i) \triangleq \text{rank}\left(y_{\text{pred}}^{(i)}\right) + \beta \text{rank}\left((\sigma_{\text{pred}}^2)^{(i)}\right)$$

where rank$(\cdot)$ denotes the low-to-high rank index of the respective score across all predictions. Rank transformation makes β easier to calibrate, especially across different uncertainty models. When high prediction values are desirable (for example, in our fluorescence prediction and gene imputation experiments), we can reverse the sign of $y_{\text{pred}}^{(i)}$ while keeping the rest of the function the same. When acquiring the top k examples for further experimentation, we simply take the examples with the k lowest values of the acquisition function, i.e., we acquire the set

$$\{\tilde{\mathbf{x}}_i : \text{rank}(a(i)) \leq k\}$$

which is a subset of the full unknown test set $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_N\}$.

7.3.3 Results

The results of our cross-validation experiment using standard, average-case performance metrics show that GP-based models are consistently competitive with, and often better than, other methods based on average-case performance metrics. The Pearson correlations between the predicted Kds and the ground truth Kds for our GP and MLP + GP models over all test data are 0.35 and 0.38, respectively ($n = 24,048$ compound-kinase pairs), in contrast with 0.26, 0.23, and 0.21 for the MLP, CMF, and DGraphDTA baselines, respectively (Figure A-19B). Good regression performance of GP-based methods is also consistent across all our metrics (Pearson correlation, Spearman correlation, and mean square error) when partitioning the test set based on exclusion of observed compounds, kinases, or both (Figure A-19B).

We also observed that, in this relatively data-limited training setting, rich pre-trained features combined with a relatively lightweight regressor (e.g., a GP or MLP) outperformed a more complex regressor architecture (i.e., DGraphDTA) trained end-to-

end on simpler features (Figure A-19B). This provides evidence that pretraining with state-of-the-art unsupervised models contributes valuable information in a data-limited setting. Where robust GP-based prediction has a substantially large advantage is in prioritizing compound-kinase pairs for further study.

In contrast to average-case metrics, focusing on top predictions directly mimics biological discovery, since researchers typically choose only a few lead predictions for further experimentation rather than testing the full, unexplored space. In GP-based models, we observed that predictions with lower uncertainty are more likely to be correct, whereas high-uncertainty predictions have worse quality (Figure A-19C), allowing us to prioritize compound-kinase pairs with high predicted affinity and low prediction uncertainty. In contrast, models without uncertainty like the MLP do not distinguish confident and uncertain predictions (Figure A-19C). The top compound-kinase pairs acquired by the GP-based models have strong, ground-truth affinities, while the other methods with poorly calibrated or nonexistent uncertainty quantification struggle to prioritize true interactions and acquire interactions with significantly higher Kds (Figures 7-2B and A-20A). Performance of the GP-based models decreases when ignoring uncertainty (Figure A-20B), suggesting that GP uncertainty helps reduce false-positives among top-acquired samples; however, other methods (BLMP and GMLPE) seem to have trouble learning meaningful uncertainty estimates (Figure A-20B).

7.4 Experimental validation with uncertainty: Breakthrough

7.4.1 Setup

We then sought to perform machine learning-guided biological discovery of previously unknown compound-kinase interactions. We use all information across the pairs of 72 compounds and 442 kinases [DHH+11] as the model training data. For the test set, we use a collection of 10,833 compounds from the ZINC database [IS05] that is

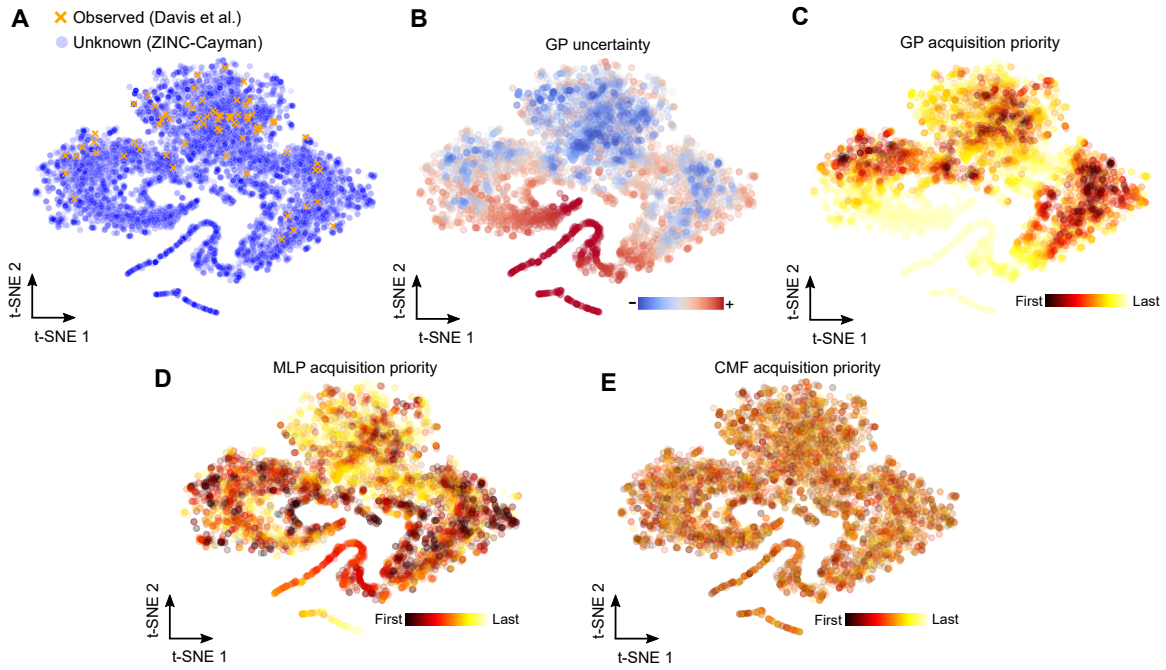

Figure 7-3: Compound feature space visualized.

(**A**) A t-SNE visualization of the compound feature space reveals regions of the compound landscape without any representative compounds with known PknB affinity measurement. (**B**) A GP assigns lower uncertainty to regions of the compound landscape close to the observed data. (**C**) A subset of the low uncertainty compounds is prioritized for experimental acquisition based on predicted binding affinity to PknB. (**D**) The MLP assigns high predicted PknB binding to a large number of out-of-distribution compounds. (**E**) CMF predictions for PknB appear to lack any meaningful structure with regards to the compound landscape. Example acquisition for other kinases is provided in Figure A-21.

commercially available through the Cayman Chemical Company. Chemicals were selected solely based on commercial availability, regardless of potential associations with kinases or any other biochemical property. The resulting "ZINC/Cayman library" consists of heterogeneous compounds (molecular weights range from 61 to 995 Da) with a median Morgan fingerprint Tanimoto similarity of 0.09; additional statistics for this library can be found in Table A.8.

7.4.2 Intuition check

We first wanted to test our intuition that test set compounds very different from any compound in the training set would also have high associated uncertainty. To do so, we visualized the 72 compounds from the training set [DHH+11] and the

10,833 unknown-affinity compounds using a two-dimensional t-SNE [vdMH08] of the structure-based compound feature space. The embedding shows large regions of the compound landscape that are far from any compounds with known affinities (Figure 7-3A).

Consistent with our intuition, a GP trained on just 72 compounds assigns uncertainty scores that are lower in regions near compounds with known affinities (Figure 7-3B), with high correlation between the uncertainty score and test compound distance to its Euclidean nearest neighbor in the training set (Spearman $r = 0.87$, $n = 10,833$ compounds). The GP prioritizes compounds within the low uncertainty regimes that also have high predicted binding affinity (Figures 7-3C and A-21). In contrast, the MLP assigns high priority to many compounds far from the known training examples (Figures 7-3D and A-21), which is most likely due to pathological behavior on out-of-distribution examples. For comparison, CMF seems unable to learn generalizable patterns from the small number of training compounds (Figures 7-3E and A-21).

7.4.3 Results: New nanomolar interactions

We then performed machine-guided discovery of compound-kinase interactions. Since our *in vitro* binding assays are optimized to screen many compounds for a given kinase, we focused our validation efforts on a set of four diverse kinases: human IRAK4, a serine/threonine kinase involved in Toll-like receptor signaling [WWS$^+$09]; human c-SRC, a tyrosine kinase and canonical proto-oncogene [WID09]; human p110δ, a lipid kinase and leukocytic immune regulator [ASP$^+$14]; and Mtb PknB, a serine/threonine kinase essential to mycobacterial viability [FSJB$^+$06]. These kinases have well-documented roles in cancer, immunological, or infectious disease [ASP$^+$14, LOS$^+$11, WWS$^+$09, WID09].

We used either our GP or MLP models to acquire compounds from the ZINC/Cayman library with high predicted affinity for each of the four kinases of interest. We validated the top five predictions returned by the GP or MLP for each kinase using an *in vitro* biochemical assay to determine the Kd. Training our models on information from 72 compounds to make predictions over a 10,833-compound library is a more

imbalanced train/test split than other reported drug-target interaction prediction settings [CLH+13, ZDMZ13, LZZ+17, HCB18, ÖÖO18, JLZ+20]. More details on how we did the biochemical validation of our prediction can be found in Appendix D.1.

We observed that none of the predictions acquired by the MLP had a Kd of less than the top tested concentration of 10 μM (Figure 7-4 and Table A.9), consistent with out-of-distribution prediction resulting in pathological model bias (Figure A-21). In contrast, the GP yielded 18 compound-kinase pairs with Kds less than 10 μM (out of 20 pairs tested, or a hit rate of 90%), 10 of which are lower than 100 nM (Figure 7-4 and Table A.9). Notably, GP acquisition yielded sub-nanomolar affinities between K252a and IRAK4 (Kd = 0.85 nM) and between PI-3065 and p110δ (Kd = 0.36 nM), automating discoveries that previously had been made with massive-scale screens or expert biochemical reasoning [ASP+14, OSM+09]. Some compounds had predicted and validated affinities for multiple kinases, such as K252a, a member of the indolocarbazole class of compounds, many of which have broad-spectrum kinase inhibition [DHH+11]. Other compounds were only acquired for one of the kinases, including PI-3065 for p110δ, WS3 for c-SRC (Kd = 4 nM), and SU11652 for PknB (Kd = 76 nM). Interestingly, the latter two of these interactions do not seem to have existing experimental support; WS3 was developed as an inducer of pancreatic beta cell proliferation [STD+13] and SU11652 was developed for human receptor tyrosine kinase inhibition [LCS+02].

To further assess the impact of uncertainty on prediction quality, we also performed PknB acquisition with another GP-based model (MLP + GP) and varied the weight β on the uncertainty. We validated the top five predictions from the GP and MLP + GP at $\beta = 1$ (tolerates some uncertainty) and $\beta = 20$ (prefers lowest Kds with a very low tolerance for uncertainty), as well as the top five predictions from the GP at $\beta = 0$ (i.e., ignoring uncertainty). At $\beta = 20$, the MLP + GP acquired a similarly potent set of compounds as the GP. Tolerating greater amounts of uncertainty, or ignoring it completely, led to more false-positive predictions (Figure 7-4B).

GP-based uncertainty quantification also enables an absolute assessment of predic-

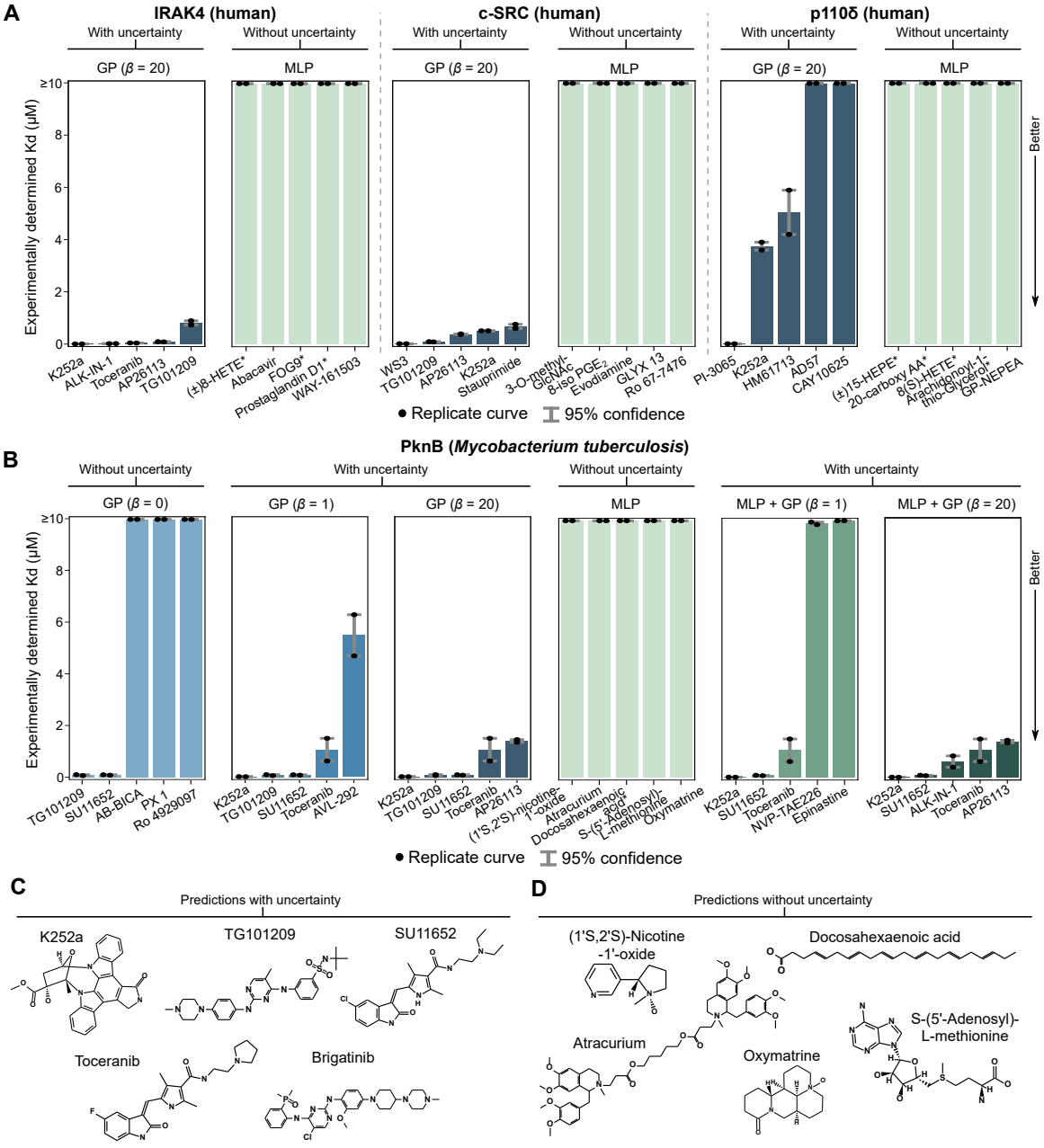

Figure 7-4: Acquisition of potent compound-kinase interactions.

(**A**) Binding affinity Kd for top five acquired compounds for three human kinases using a model with uncertainty (GP) (Figure A-22) and without (MLP). Asterisks after compound names indicate compounds incompatible with the validation assay. Mean Kd values are provided in Table A.9. (**B**) We validated the top five compound predictions at different acquisition β parameters for the models with uncertainty (GP and MLP + GP) and the top five compound predictions provided by the MLP. Incorporating uncertainty information (Figure A-22) reduces false-positive predictions. Asterisks after compound names indicate compounds incompatible with the validation assay. Mean Kd values are provided in Table A.9. (**C**) The structures of the compounds prioritized by the GP for PknB-binding affinity with acquisition $\beta = 20$. (**D**) The structures of the compounds prioritized by the MLP for PknB-binding affinity, none of which have a strong affinity (Kd $\geq$ 10,000 nM).

tion quality. For example, all predictions with a mean less than 10 μM (our top-tested concentration) and an interquartile range less than 2 μM resulted in true positive hits (Figure A-22). In contrast, more dispersed prediction distributions had higher variability in the potency of the true binding interaction including false positives (Figure A-22), suggesting that our GP-based models make better predictions when they are more confident. Uncertainty adds an interpretable dimension to machine-generated predictions, so a researcher with a low tolerance for false positives might ignore a generated hypothesis with a low predicted Kd but a high uncertainty.

The ability to discover new nanomolar-range interactions provided excellent validation for our uncertainty-based approach. Not only did a consideration of uncertainty help fill the gap between a ~50% hit rate in the previous chapter to a 90% hit rate in this chapter, but it also helps us reason about an exploration/exploitation trade-off in which a lower hit rate might be tolerated for greater biological novelty.

7.5 Application note: Discovering potential tuberculosis drugs

7.5.1 Novel anti-Mtb activity

Given the potent interactions discovered by our models, we wanted to further probe the implications of our findings in an infectious disease setting, especially since we discovered new nanomolar binders of PknB, a kinase that is essential to Mtb viability [FSJB+06]. Bacterial kinases are less well studied than human (or mammalian) kinases [JCP18] but are nonetheless important therapeutic targets [FSJB+06, LOS+11, OLA+14]. Tuberculosis remains the leading cause of infectious disease death globally [FCP19], underscoring the importance of further therapeutic development. Given the essentiality of PknB and our *in silico* identification of PknB-binding compounds, we sought to examine if the compounds with high binding affinity to PknB would have any impact on mycobacterial growth. This would not be guaranteed since factors like cell wall permeability or intracellular stability were not explicitly encoded in the

training data.

We focused on the compounds with a Kd less than 100 nM: K252a (Kd = 11 nM), TG101209 (Kd = 71 nM), and SU11652 (Kd = 76 nM). Using the colorimetric, resazurin microtiter assay (alamar blue) [LOS$^+$11, Ram12], we determined the minimum inhibitory concentration (MIC) of these compounds as well as rifampicin, a frontline antibiotic for tuberculosis [FCP19] (Appendix D.2); the MICs for these compounds with H37Rv are shown in Table A.10. We observed that K252a and SU11652 inhibited the growth of H37Rv compared to a dimethyl sulfoxide (DMSO) vehicle control (one-sided t-test P-value of 7.0×10^{-8} for K252a and 3.9×10^{-8} for SU11652, $n = 3$ replicate cultures per condition) (Figures 7-5A). SU11652 is a well-documented inhibitor of human receptor tyrosine kinases including PDGFR, VEGFR, and Kit [LCS$^+$02]. TG101209 did not inhibit growth of H37Rv (one-sided t-test P-value of 0.11, $n = 3$ replicate cultures per condition) (Figures 7-5A), perhaps due to low cell permeability [Bre03, HLN$^+$08]. These results were corroborated using additional validation where Mtb expressing the luxABCDE cassette (luxMtb) was incubated with increasing concentrations of K252a, SU11652, and TG101209 (Appendix D.2).

We further validated these results in a more complex, host-pathogen model. Macrophages were infected with luxMtb and luminescence is measured as a proxy of bacterial growth [AZF$^+$10, BTZ$^+$17] (see Appendix D.2). We infected macrophages with luxMtb for 4 hours prior to the addition of compounds dissolved in cell culture media. Consistent with our axenic culture experiments, treatment with K252a and SU11652 resulted in less luminescence as compared to DMSO (one-sided t-test P-value of 2.9×10^{-6} for K252a and 2.8×10^{-6} for SU11652; $n = 3$ replicate cultures per condition) (Figure 7-5B,C). In examining the literature for prior work on compounds targeting PknB, we identified support for K252a as an inhibitor of PknB kinase activity and Mtb growth [FSJB$^+$06, OLA$^+$14]. These previous studies and our results nominate future experiments to further investigate the biochemistry of PknB and the potential use of K252a and SU11652 as scaffolds for PknB- and Mtb-related drug development.

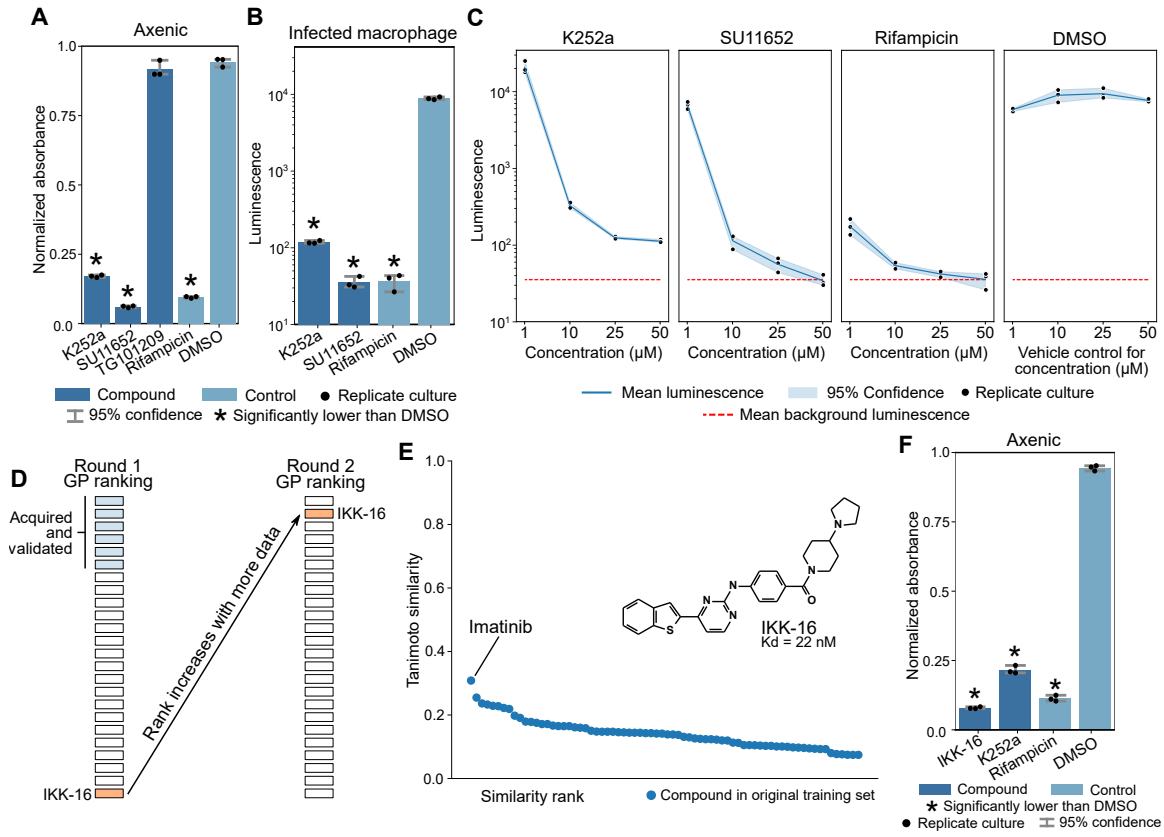

Figure 7-5: Anti-Mtb whole-cell activity and an out-of-distribution inhibitor.

(**A**) Growth of axenic Mtb measured via alamar blue absorbance after 5 days of axenic incubation in media treated with compounds, or a DMSO vehicle control, at 50 μM. Statistical significance was assessed with a one-sided t-test P-value at FDR < 0.05. (**B**) Luminescence of luciferase-expressing Mtb from within infected human macrophages cultured in media treated with compounds at 50 μM. Statistical significance was assessed with a one-sided t-test P-value at FDR < 0.05. (**C**) Dose-response of K252a, SU11652, rifampicin, or a DMSO vehicle control on the luminescence of luciferase-expressing Mtb from within infected human macrophages after 5 days of culture post-infection. (**D**) IKK-16 was ranked 24 by the GP during the first round of compound acquisition. Six of the compounds above IKK-16 in the first-round GP ranking were acquired for experimental validation (the sixth-ranked compound was in the top five for the MLP + GP). Following model retraining on first-round PknB-binding acquisitions across all models, IKK-16 was the second-ranked compound. (**E**) All 72 compounds in the original training set have a Morgan fingerprint (radius 2, 2,048 bits) Tanimoto similarity of 0.31 or less with IKK-16 (structure shown). See also Table A.11. (**F**) An additional follow-up assessment of Mtb growth via alamar blue absorbance after five days of axenic incubation in media treated with IKK-16, other compounds, or a DMSO vehicle control, at 50 μM. Statistical significance was assessed with a one-sided t-test P-value at FDR < 0.05.

7.5.2 Active learning

We then pushed this even further with a follow-up analysis in which we incorporated the results of the initial kinase inhibition experiments to make new predictions, a setting in which sample-efficiency is paramount. This iterative cycle involving prediction, acquisition, model retraining, and subsequent prediction and acquisition is referred to as "active learning" [Eis20, SC17]. We conducted a second round of PknB binding affinity predictions after training on both the original dataset and the results from our first round of *in vitro* affinity experiments (Figure 7-4B). We trained GP and MLP models on this data and again acquired the top five predictions made by each.

All MLP-acquired compounds again had a PknB Kd greater than 10 μM. Although the GP uncertainty scores increased by as much as a factor of 2 from the first round (Figure A-22), indicating hypotheses that explored riskier, more distal regions of the compound landscape, we still found that one of the GP-acquired compounds, IKK-16 [WBB$^+$06], binds PknB with a Kd of 22 nM, the second lowest PknB Kd over all our experiments (Table A.9). IKK-16 had an acquisition ranking of 24 during the first round but a ranking of 2 in the second round (Figure 7-5D), indicating that the GP efficiently adapted its beliefs based on a handful of new datapoints to make a successful second-round prediction. Notably, among all training compounds in both the first and second prediction rounds, the most similar structure to IKK-16 is imatinib with a Tanimoto similarity of 0.31 (Figure 7-5E and Table A.11), indicating that IKK-16 is structurally remote to any compound in the training data; for reference, a recently used threshold was a Tanimoto similarity of 0.40 [SYS$^+$20].

Follow-up experiments also revealed whole-cell activity of IKK-16 against H37Rv Mtb in axenic culture (Figure 7-5F and Table A.10), with significant growth inhibition compared to a DMSO vehicle control (one-sided t-test P-value of 6.9×10^{-9}, $n = 3$ replicate cultures per condition). We could not find existing literature linking IKK-16 to PknB or Mtb in general. These results also illustrate how uncertainty combined with an active learning strategy can explore regions of the compound space that are more distal to the original training set.

More broadly, these experiments provide an example of how machine learning can help accelerate the drug discovery process. Here, we identified drugs that could be repurposed for Mtb inhibition in a matter of a few weeks through the help of machine learning, in particular leveraging prediction uncertainty through GPs and modern neural pretraining. We hope that the work on DTIs and uncertainty in the last two chapters provides a roadmap for future algorithms hoping to translate good cross-validated performance into new biological discoveries.

Chapter 8

Fighting Disease III: Resistance

We will now discuss in a little more detail the Struggle for Existence.

—Charles Darwin, *On the Origin of Species* (1859)

Once a therapy or vaccine has been developed for a given pathogen, one may be tempted to assume that all is well. Unfortunately, over time, pathogens can acquire *resistance* to a drug or to immunity, reducing or eliminating efficacy. This happens due to simple evolution: pathogens randomly mutate their genomes across many generations and widespread drug use or vaccination in the host will select mutations that confer resistance to the drug or vaccine.

In this chapter, we focus on viruses that mutate their surface proteins to acquire resistance to a neutralizing antibody response, a problem known as *immune escape*. Escape is a tremendous problem when developing effective vaccines against some of the world's deadliest pathogens, including influenza and HIV. We therefore develop a cutting edge approach based on neural language models that learns patterns of viral sequence variation in order to predict escape mutations from sequence data alone [HZBB21][1]. The work described in this chapter is, to the best of our knowledge, the first computational approach directly designed to predict escape.

Predicting escape is useful because it could inform therapeutic design that antic-

[1]Software related to this chapter is available at `https://github.com/brianhie/viral-mutation`.

ipates escape before it occurs. For example, it may be possible to design a vaccine that elicits an immune response against regions of a viral protein that are less prone to escape, or to vaccinate against future forms of a virus. This chapter and its implications are exciting because they open up many new lines of research; for example, in theory, our approach can generalize to any form of selection pressure, including drug selection. Moreover, this work gives us hope that, while many pathogens have so far largely evaded human attempts to eradicate them, we can ultimately gain the upper hand.

8.1 Glossary

- *Resistance*: In infectious disease, pathogens can gain resistance to a therapy or to immunity by mutating their genome so that the therapy/immune response no longer works, but also so that the pathogen preserves viability and infectivity.

- *Immune escape*: A form of resistance where a pathogen mutates to evade an immune response, usually in the context of humoral immunity and antibody neutralization.

- *Language model*: A machine learning model that predicts the probability of a token given some sequence context, e.g., the probability of the next word given the preceding words in a sentence.

- *Constrained semantic change search (CSCS)*: An algorithm that searches for mutations to an entity that preserves a notion of "grammaticality" but also induces high "semantic change."

8.2 Preliminaries

(Background on the human immune system helpful for this chapter, including on humoral immunity and antibodies, is provided in Section 2.1.4.)

8.2.1 Distributional semantics

Key concepts in this chapter come from the field of linguistics. The first is the notion that information encoded about the meaning or the "semantics" of a word is encoded in how different words appear together or "co-occur." The idea now known as the "distributional hypothesis" was introduced by Zellig Harris in 1954 [Har54] and summarized well by J.R. Firth in 1957 [Fir57]:

You shall know a word by the company it keeps.

This observation—that word co-occurence patterns have insight into the meaning of words—has been tremendously productive in the field on natural language processing (NLP).

The idea behind distributional semantics also makes a lot of intuitive sense. Humans are quite good at inferring missing words based on sequence context. For example consider the sentences below with the same missing word:

- Let's keep the kitchen _____.
- The new design has _____ lines.
- I forgot to _____ out the cabinet.

Usually, native English speakers can come up with the correct missing word, "clean," fairly easily.

Distributional semantics has been particularly influential in machine learning because it is data friendly. Word co-occurences can be learned from a large sequence corpus and similarities in co-occurence patterns can define semantic similarity. Therefore, "semantics," an abstract concept can be expressed in the language of mathematics and probability.

8.2.2 Language models

An important machine learning algorithm built on the distributional hypothesis is the *language model* [MSC+13, DL15, PNI+18, DCLT18, RWC+19a]. A language model

learns a distribution of sequence continuations given an initial sequence prefix as input. For example, given the sequence "Let's keep the kitchen", a language model might assign high probability to the single-word continuation "clean" and a lower probability to the continuation "telescope." Neural language models use a neural network architecture, especially those designed for sequences like recurrent neural networks, to learn the function from sequence context to the distribution over tokens.

Language models can also learn a latent variable, from which it predicts the word continuations; these latent variables can be interpreted as an embedding of the input sequence. Because of the distributional hypothesis, distance in this embedding space can capture semantic similarity. For example, the sequences of words "the men advance," "the soldiers advance," and "the three advance" have a similar set of possible word continuations and would have similar embeddings, while "the cash advance" has a nearly disjoint set of continuations and thus a different embedding. Learning continuous semantic embeddings has been a large area of work in NLP [MSC+13, PNI+18].

8.3 Viral escape and mutational semantics

8.3.1 Motivation

In this chapter, we are motivated by the problem of viral escape. Viral mutation that escapes from recognition by neutralizing antibodies has prevented the development of a universal antibody-based vaccine for influenza [EK01, KWW18, Kra19, KLR+15] or human immunodeficiency virus (HIV) [EK01, AJS12, RWLP03, RKK01] and remains a concern in the development of therapies for COVID-19 [BFW+20], caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) infection [ARL+20, WPT+20]. Obtaining a better understanding of viral escape has motivated high-throughput experimental techniques, such as deep mutational scans (DMS), that perform causal escape profiling of all single-residue mutations to a viral protein [DAW+19, DLB18, LEZ+19, GSG+20]. Such techniques, however, require substantial

effort to profile even a single viral strain, so empirically testing the escape potential of all (combinatorial) mutations in all viral strains remains infeasible.

A more efficient model of viral escape could be achieved computationally. One of our key initial insights is that it may be possible to train an algorithm to learn to model escape from existing viral sequence data alone. Such an approach is not unlike recent algorithmic successes in learning properties of natural language from large text corpuses [DCLT18, PNI+18, RWC+19a]; like viral evolution, natural languages like English or Japanese use linear sequence to encode complex concepts (e.g., semantics) and are under complex constraints (e.g., grammar). We pursued the intuition that critical properties of a viral escape mutation have linguistic analogs: first, the mutation must preserve viability and infectivity, i.e., it must be grammatical; second, the mutation must be antigenically altered to evade immunity, i.e., it must have substantial semantic change.

Currently, computational models of viral protein evolution focus on viral fitness [LKB+18, HGS+19] or on functional/antigenic similarity [MYD+11, AKB+19, BB19, RBT+19] alone. The novel concept critical to our study is that computationally predicting viral escape requires modeling both fitness and antigenicity (Figure 8-1A). Moreover, rather than developing two separate models of fitness and function, we reasoned that we could develop a single model that simultaneously achieves both these tasks. To do so, we leverage state-of-the-art machine learning algorithms (originally developed for natural language understanding) called language models [MSC+13, DL15, PNI+18, DCLT18, RWC+19a], which learn the probability of a token (e.g., an English word) given its sequence context (e.g., a sentence) (Figure 8-1B). As done in natural language tasks, we can use a hidden layer output within a neural language model as a semantic embedding [PNI+18] and the language model output to quantify mutational grammaticality (Figure 8-1B); moreover, the same principles used to train a language model on a sequence of English words can be used to train a language model on a sequence of amino acids.

The main hypothesis underlying this whole chapter, therefore, is that

i. language model-encoded semantic change corresponds to antigenic change,

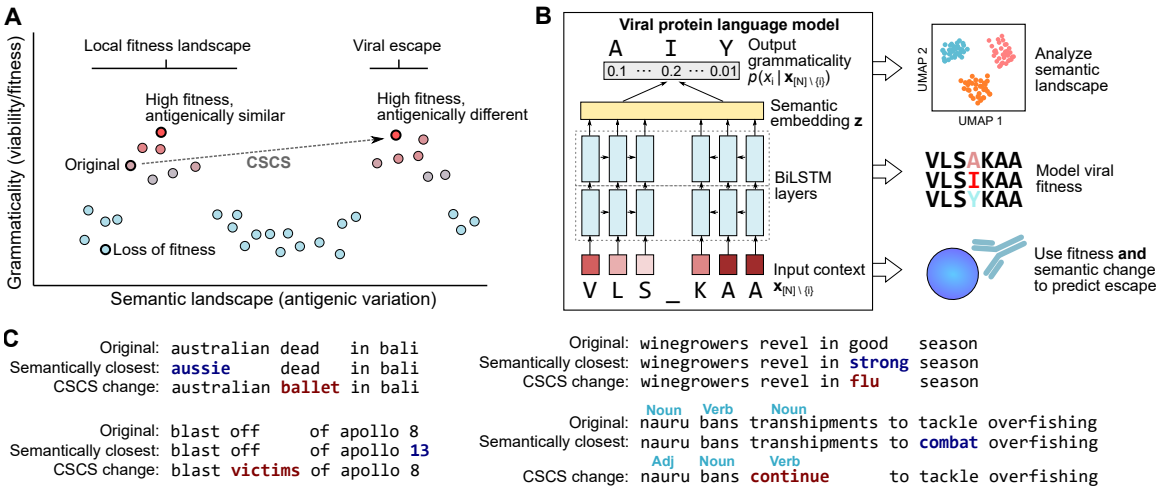

Figure 8-1: Semantic change and grammaticality for escape prediction.

(**A**) Constrained semantic change search (CSCS) for viral escape prediction is designed to search for mutations to a viral sequence that preserve fitness while being antigenically different. This corresponds to a mutant sequence that is grammatical (conforms to the structure and rules of a language) but has high semantic change with respect to the original (e.g., wildtype) sequence. (**B**) A neural language model with a bidirectional long short-term memory (BiLSTM) architecture is used to learn both semantics (as a hidden layer output) and grammaticality (as the language model output). CSCS combines semantic change and grammaticality to predict escape (12). (**C**) CSCS-proposed changes to a news headline (implemented using a neural language model trained on English news headlines) makes large changes to the overall semantic meaning of a sentence or to the part-of-speech structure. The semantically closest mutated sentence according to the same model is largely synonymous with the original headline.

ii. language model grammaticality captures viral fitness, and

iii. both high semantic change *and* grammaticality help predict viral escape.

Searching for mutations with both high grammaticality and high semantic change is a newly formulated task that we call constrained semantic change search (CSCS) and which we describe in detail in the next section.

8.3.2 Problem formulation

Intuitively, our goal is to identify mutations that induce high semantic change (e.g., a large impact on biological function) while being grammatically acceptable (e.g, biologically viable). More precisely, we are given a sequence of tokens defined as $\mathbf{x} \triangleq (x_1, ..., x_N)$ such that $x_i \in \mathcal{X}, i \in [N]$, where $\mathcal{X}$ is a finite alphabet (e.g., characters or words for natural language, or amino acids for protein sequence). Let $\tilde{x}_i$ denote a mutation at position i and the mutated sequence as $\mathbf{x}[\tilde{x}_i] \triangleq (..., x_{i-1}, \tilde{x}_i, x_{i+1}, ...)$.

We first require a semantic embedding $\mathbf{z} \triangleq f_s(\mathbf{x})$, where $f_s : \mathcal{X}^N \to \mathbb{R}^K$ embeds discrete-alphabet sequences into a continuous space, where, ideally, closeness in embedding space would correspond to semantic similarity. We denote semantic change as the distance in embedding space, i.e.,

$$\Delta\mathbf{z}[\tilde{x}_i] \triangleq \|\mathbf{z} - \mathbf{z}[\tilde{x}_i]\| = \|f_s(\mathbf{x}) - f_s(\mathbf{x}[\tilde{x}_i])\| \tag{8.1}$$

where $\|\cdot\|$ denotes a vector norm. The grammaticality of a mutation is described by

$$p(\tilde{x}_i|\mathbf{x}), \tag{8.2}$$

which takes values close to zero if $\mathbf{x}[\tilde{x}_i]$ is not grammatical and close to one if it is grammatical.

Our objective combines semantic change and grammaticality as a linear combination

$$a(\tilde{x}_i; \mathbf{x}) \triangleq \Delta\mathbf{z}[\tilde{x}_i] + \beta p(\tilde{x}_i|\mathbf{x})$$

for each possible mutation $\tilde{x}_i$ and a user-specified parameter $\beta \in [0, \infty)$. Mutations $\tilde{x}_i$ are prioritized based on $a(\tilde{x}_i; \mathbf{x})$. We refer to ranking mutations based on semantic change and grammaticality as CSCS.

8.4 Algorithms

8.4.1 Language modeling

Algorithms for CSCS could potentially take many forms; for example, separate algorithms could be used to compute $\Delta\mathbf{z}[\tilde{x}_i]$ and $p(\tilde{x}_i|\mathbf{x})$ independently, or a two-step approach might be possible that computes one of the terms based on the value of the other.

Instead, we reasoned that a single approach could compute both terms simultaneously, based on learned language models that learn the probability distribution of a word given its context [MSC+13, DL15, PNI+18, DCLT18, RWC+19a]. The language model we use throughout our experiments considers the full sequence context of a word and learns a latent variable probability distribution $\hat{p}$ and function $\hat{f}_s$, where, for all $i \in [N]$,

$$\hat{p}(x_i|\mathbf{x}_{[N]\setminus\{i\}}, \hat{\mathbf{z}}_i) = \hat{p}(x_i|\hat{\mathbf{z}}_i) \quad \text{and} \quad \hat{\mathbf{z}}_i = \hat{f}_s(\mathbf{x}_{[N]\setminus\{i\}}),$$

i.e., latent variable $\hat{\mathbf{z}}_i$ encodes the context $\mathbf{x}_{[N]\setminus\{i\}} \triangleq (..., x_{i-1}, x_{i+1}, ...)$ such that x_i is conditionally independent of its context given the value of $\hat{\mathbf{z}}_i$.

We use different aspects of the language model to describe semantic change and grammaticality by setting terms (8.1) and (8.2) as

$$\Delta\mathbf{z}[\tilde{x}_i] \triangleq \|\hat{\mathbf{z}} - \hat{\mathbf{z}}[\tilde{x}_i]\|_1 \quad \text{and} \quad p(\tilde{x}_i|\mathbf{x}) \triangleq \hat{p}(\tilde{x}_i|\hat{\mathbf{z}}_i),$$

where $\hat{\mathbf{z}} \triangleq \begin{bmatrix} \hat{\mathbf{z}}_1^{\mathrm{T}} & \dots & \hat{\mathbf{z}}_N^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ is the concatenation of embeddings for each token, $\hat{\mathbf{z}}[\tilde{x}_i]$ is defined similarly but for the mutated sequence, and $\|\cdot\|_1$ is the ℓ_1 norm, chosen because of more favorable properties compared to other standard distance metrics,

though other metrics could be empirically quantified in future work [AHK01].

Effectively, distances in embedding space are used to approximate semantic change and the emitted probability approximates grammaticality. We note that these modeling assumptions are not guaranteed to be perfectly specified, since, in the natural language setting for example, antonyms may also be close in embedding space and the language model output can also encode linguistic pragmatics in addition to grammaticality. However, we still find these modeling assumptions to have good empirical support.

Training or parameterizing the language model is separate from CSCS, and the novelty of CSCS is in leveraging these models in a new way. An advantage of this approach is that it does not require any bespoke modifications to the general language modeling framework, other than requiring a continuous latent variable. CSCS can therefore leverage the noted multitask generality of language models [RWC+19a].

Importantly, this approach to CSCS is completely unsupervised. Rather than assume access to labels explicitly encoding semantics or grammaticality, the model instead extracts this information from a large unlabeled corpus. This is critical in domains, like viral genomics, in which large sequence corpuses are available but functional profiling is limited. These corpuses implicitly contain information related to grammaticality or infectivity (e.g., all sequences are grammatically acceptable or come from infectious virus), but the algorithm must learn these rules from data.

8.4.2 Architecture

Based on the success of recurrent architectures for protein-sequence representation learning [BB19, RBT+19, AKB+19], we use similar encoder models for viral protein sequences (Figure 8-1). Our model passes the full context sequence into bidirectional long-short-term-memory (BiLSTM) hidden layers. We used the concatenated output of the final LSTM layers as the semantic embedding, i.e.,

$$\hat{\mathbf{z}}_i \triangleq \left[\text{LSTM}_f(g_f(x_1, ..., x_{i-1}))^{\text{T}} \quad \text{LSTM}_r(g_r(x_{i+1}, ..., x_N))^{\text{T}}\right]^{\text{T}}$$

where g_f is the output of the preceding forward-directed layer, LSTM_f is the final forward-directed LSTM layer, and g_r and LSTM_r are the corresponding reverse-directed components. The final output probability is a softmax-transformed linear transformation of $\hat{\mathbf{z}}_i$, i.e.,

$$\hat{p}(x_i|\mathbf{x}_{[N]\setminus\{i\}}) \triangleq \text{softmax}(\mathbf{W}\hat{\mathbf{z}}_i + \mathbf{b})$$

for some learned model parameters $\mathbf{W}$ and $\mathbf{b}$. In our experiments, we used a 20-dimensional dense embedding for each element in the alphabet $\mathcal{X}$, two BiLSTM layers with 512 units, and categorical cross entropy loss optimized by Adam with a learning rate of 0.001, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. Additional details on hyperparameter selection are given in Appendix E.2.1.

8.4.3 Rank-based acquisition

Rather than acquiring mutations based on raw semantic change and grammaticality values, which may be on very different scales, we find that selecting β is much easier in practice when first rank-transforming the semantic change and grammaticality terms, i.e., acquiring based on

$$a'(\tilde{x}_i; \mathbf{x}) \triangleq \text{rank}(\Delta\mathbf{z}[\tilde{x}_i]) + \beta\,\text{rank}(p(\tilde{x}_i|\mathbf{x})).$$

All possible mutations $\tilde{x}_i$ are then given priority based on the corresponding values of $a'(\tilde{x}_i; \mathbf{x})$, from highest to lowest. Our empirical results have consistently good performance by simply setting $\beta = 1$ (equally weighting both terms), which we used in all experiments below unless otherwise noted. In this study, we deal with the unsupervised setting where β is a parameter but note that adding some supervision could learn β (or other, non-rank, transformations) from data.

8.4.4 Connection to viral escape

A language model is a probability distribution over sequences learned from a corpus of data. For any sequence $\mathbf{x}$, the model will output a predicted probability $p(\mathbf{x})$ of observing that sequence in the training data distribution. We call $p(\mathbf{x})$ "grammaticality" because in natural language tasks, $p(\mathbf{x})$ tends to be high for grammatically correct sentences. In the case of viral sequences, the training distribution consists of viral proteins that have evolved for high fitness/virality, so we hypothesize that high grammaticality corresponds to high viral fitness.

However, high fitness alone does not indicate an escape mutation. For example, a viral protein with a neutral mutation will have equally high fitness but may not look different enough to escape detection by the immune system, i.e., it will have no "antigenic" change. To identify mutations that do lead to large antigenic changes, we exploit the internal sequence embeddings learned by the language model. If two sequences have similar embeddings, then they have similar distributions over sequence continuations given the input tokens. We hypothesize that neutral mutations should not affect the distribution over amino acids at other positions, while mutations that affect antigenicity do affect the distribution over other positions. Thus, the combination of high sequence probability (high fitness) and a large change in embedding (antigenic change) indicates an escape mutation. The natural language analogy is to find the single-token change that induces the highest semantic change while preserving grammaticality (Figure 8-1C).

We propose, to our knowledge, the first general computational model of viral escape. Notably, our neural language model implementation of CSCS is based on sequence data alone (beneficial since sequence is easier to obtain than structure) and requires no explicit escape information (i.e., it is completely unsupervised), does not rely on multiple sequence alignment (MSA) preprocessing (i.e., it is alignment-free), and captures global relationships across an entire sequence (e.g., since word choice at the beginning of a sentence can influence word choice at the end).

8.4.5 Extension to combinatorial mutations

The above exposition is limited to the setting in which mutations are assumed to be single-token. We perform a simple extension to handle combinatorial mutations. We denote such a mutant sequence as $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_N)$, which has the same length as $\mathbf{x}$, where the set of mutations that consists of the tokens in $\tilde{\mathbf{x}}$ that disagree with those at the same position in $\mathbf{x}$ is

$$\mathcal{M}(\mathbf{x}, \tilde{\mathbf{x}}) \triangleq \{\tilde{x}_i | \tilde{x}_i \neq x_i\}.$$

The semantic embedding can simply be computed as $f_s(\tilde{\mathbf{x}})$ from which semantic change can be computed as above. For the grammaticality score, we make a simple modeling assumption and compute grammaticality as

$$\prod_{\tilde{x} \in \mathcal{M}(\mathbf{x}, \tilde{\mathbf{x}})} p(\tilde{x}_i | \mathbf{x}),$$

i.e., the product of the probabilities of the individual point-mutations (implemented in the log domain for better numerical precision). Other ways of estimating joint, combinatorial grammaticality terms while preserving efficient inference are also worth considering in future work.

While we do not consider insertions or deletions in this study, we do note that, in viral sequences, insertions and deletions are rarer than substitutions by a factor of four or more (49) and the viral mutation datasets that we considered exclusively profiled substitution mutations alone. Extending our algorithms to compute semantic change of sequences with insertions or deletions would be essentially unchanged from above. The more difficult task is in reasoning about and modeling the grammaticality of an insertion or a deletion. While various grammaticality heuristics based on the language model output may be possible, this is also an interesting area for further methodological development.

8.4.6 Related work

The CSCS problem is related to work focused on identifying the best interventions to structured data to produce a desired outcome [MRDJ17, PBM16]. Such work often assumes a dataset that includes both the observed features and corresponding outcomes, which allows for supervised learning. In contrast, we assume no explicit labels of semantic change and must resort to unsupervised learning to extract this information. This is because in domains like viral mutation, data that directly measures viral fitness is very limited, while unlabeled sequence data is abundant.

Importantly, our CSCS task is distinct from representation learning tasks that construct semantically meaningful embeddings, but CSCS does stand to benefit from innovation in representation learning. Using hidden states in a language model to represent natural language semantics has been an influential and productive idea [PNI+18]. Rather than acquiring mutations based on greatest semantic change as in CSCS, acquisition based instead only on lowest $\Delta\mathbf{z}[\tilde{x}_i]$ essentially performs semantic similarity search among all sequences that differ by a single token.

In biological applications, neural language models have been developed to learn unsupervised or weakly supervised protein sequence embeddings that encode generic protein similarity [BB19, RBT+19, AKB+19]. To our knowledge, however, no previous work has considered how mutations affect these embeddings, nor have such methods been applied to evolutionary change. Furthermore, while many variants of recurrent or transformer-based architectures have been proposed for protein sequence modelling tasks, we note any such current or future language model architecture could be used in CSCS.

Some work in computational biology has focused on identifying deleterious mutations in human or mammalian genomes with clinical relevance [SGP+18, RWC+19b]. However, these approaches are based on direct supervision under the assumption that rare or poorly conserved mutations are deleterious. Such an assumption, however, does not apply to escape mutations, which could be both frequent or infrequent in a population. Viral genomes are also more highly variable than mammalian genomes

(e.g., "Drake's rule"), so aligning mutations across viral strains is more difficult [Dra91, CDCPGS09, SNC+10].

Most computational analyses specific to viral mutation require rich metadata beyond raw sequence or make virus-specific assumptions [BRB+15, YLD+20] (for example, vaccine-related temporal patterns in influenza, which are absent for HIV). Most similar to our approach, models exist for learning viral fitness from a large sequence corpus [HIP+17, HGS+19]. These approaches, however, requires time-consuming and error-prone multiple sequence alignment (MSA) preprocessing [KS13] and only consider pairwise information couplings among residues, which, as demonstrated below, limit performance when predicting escape. To our knowledge, our work is the first to effectively model viral escape that generalizes to any relevant genomic sequence from diverse viruses, without the need for sequence alignment, complex metadata, or special assumptions on mutational processes.

8.5 Learning the language of viral escape

8.5.1 Experimental setup

We wanted to assess the empirical performance of CSCS and assess generality across viruses by analyzing three important proteins: influenza A hemagglutinin (HA), HIV-1 envelope glycoprotein (Env), and SARS-CoV-2 spike glycoprotein (Spike). All three are found on the viral surface, are responsible for binding host cells, are targeted by antibodies, and are important drug targets given their role in pandemic disease events and widespread human mortality [EK01, KWW18, Kra19, KLR+15, AJS12, RWLP03, RKK01, BFW+20, ARL+20, WPT+20]. We trained a separate language model for each protein using a large corpus of virus-specific amino acid sequences.

Influenza HA amino acid sequences were downloaded from the "Protein Sequence Search" section of `https://www.fludb.org`. We only considered complete hemagglutinin sequences from virus type A. We trained an amino acid residue-level language model on a total of 44,851 unique influenza A HA amino acid sequences observed in

animal hosts from 1908 through 2019.

HIV Env protein sequences were downloaded from the "Sequence Search Interface" at the Los Alamos National Laboratory (LANL) HIV database (`https://www.hiv.lanl.gov`). All complete HIV-1 Env sequences were downloaded from the database, excluding sequences that the database had labeled as "problematic." We additionally only considered sequences that had length between 800 and 900 amino acid residues, inclusive. We trained an amino acid residue-level language model on a total of 57,730 unique Env sequences.

Coronavidae spike glycoprotein sequences were obtained from the Gene/Protein Search portal of the ViPR database (`https://www.viprbrc.org/brc/home.spg?decorator=corona`) across the entire *Coronavidae* family. We only included amino acid sequences with "spike" gene products. SARS-CoV-2 Spike sequences were obtained from the Severe acute respiratory syndrome coronavirus 2 datahub at NCBI Virus (`https://www.ncbi.nlm.nih.gov/labs/virus/vssi/`). Betacoronavirus spike sequences from GISAID also used in Starr et al.'s analysis [SGH+20] were obtained from the accompanying GitHub repository. Across all coronavirus datasets, we furthermore excluded sequences with a protein sequence length of less than 1,000 amino acid residues. We trained an amino acid residue-level language model on a total of 4,172 unique Spike (and homologous protein) sequences.

8.5.2 Semantics predicts antigenicity

We initially sought to investigate the first part of our hypothesis, namely that the semantic embeddings produced by a viral language model would be antigenically meaningful. We computed the semantic embedding for each sequence in the influenza, HIV, and coronavirus corpuses; we then visualized the semantic landscape by learning a two-dimensional approximation of the high-dimensional semantic embedding space using Uniform Manifold Approximation and Projection (UMAP) [MH18]. The semantic landscape of each protein shows clear clustering patterns corresponding to subtype, host species, or both (Figure 8-2), suggesting that the model was able to learn functionally meaningful patterns from raw sequence alone.

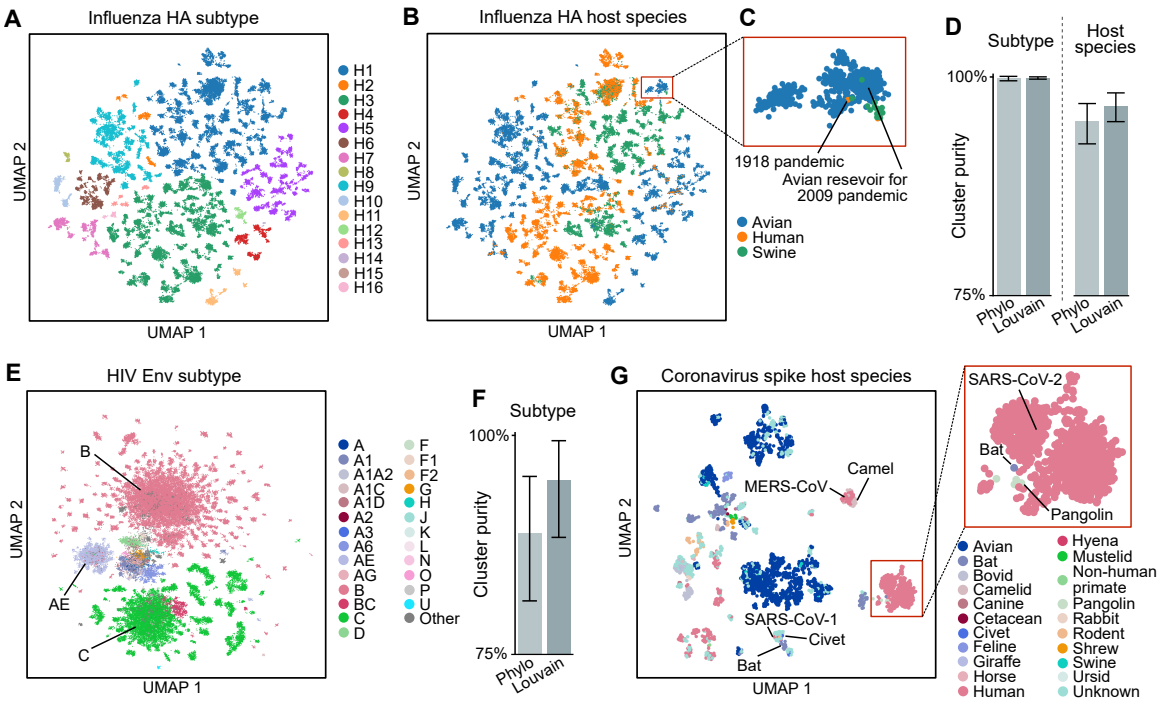

Figure 8-2: Semantic embedding landscape is antigenically meaningful.

(**A**, **B**) UMAP visualization of the high-dimensional semantic embedding landscape of influenza HA. (**C**) A cluster consisting of avian sequences from the 2009 flu season onwards also contains the 1918 pandemic flu sequence, consistent with their antigenic similarity (15). (**D**) Louvain clusters of the HA semantic embeddings have similar purity with respect to subtype or host species compared to phylogenetic sequence clustering (Phylo). Bar height: mean; error bars: 95% confidence. (**E**, **F**) The HIV Env semantic landscape shows subtype-related distributional structure and high Louvain clustering purity. Bar height: mean; error bars: 95% confidence. (**G**) Sequence proximity in the semantic landscape of coronavirus spike proteins is consistent with the possible zoonotic origin of SARS-CoV-1, MERS-CoV, and SARS-CoV-2.

We can quantify these clear clustering patterns, which are visually enriched for particular subtypes or hosts, by using Louvain clustering [BGLL08] to group sequences based on their semantic embeddings (Figure A-23), followed by measuring the clustering purity based on the percent composition of the most represented metadata category (sequence subtype or host species) within each cluster. Average cluster purities for HA subtype, HA host species, and Env subtype are 99%, 96%, and 95%, respectively, which are comparable to or higher than the clustering purities obtained by more traditional MSA-based phylogenetic reconstruction [BMM+19, KS13] (Figure 8-2D,F).

Within the HA landscape, clustering patterns suggest interspecies transmissibility. Interestingly, the sequence for 1918 H1N1 pandemic influenza belongs to the main avian H1 cluster, containing sequences from the avian reservoir for 2009 H1N1 pandemic influenza (Figures 8-2C and A-23). Our model's suggested antigenic similarity between H1 HA from 1918 and 2009, though nearly a century apart, has well-established structural and functional support [WBD+10, XEK+10]. Within the HIV Env landscape, unlike in HA, clusters corresponding to a few subtypes dominate the landscape (Figure 8-2E), perhaps due to the absence of vaccine pressure leading to abundant representation of similar viral strains.

Within the landscape of SARS-CoV-2 Spike and homologous proteins, clustering proximity is consistent with the suggested zoonotic origin of several human coronaviruses (Figure 8-2G), including bat and civet for SARS-CoV-1 [WE07]; camel for Middle East respiratory syndrome-related coronavirus (MERS-CoV) [CPG+14]; and bat and pangolin for SARS-CoV-2 [ARL+20]. Analysis of these semantic landscapes strengthens our hypothesis that our viral sequence embeddings encode functional and antigenic variation.

8.5.3 Grammaticality predicts fitness

Not only does escape prediction stand to benefit from modeling antigenic change, but from modeling viral fitness as well. Therefore, in line with the second part of our hypothesis, we assessed the relationship between viral fitness and language model grammaticality using high-throughput DMS characterization of hundreds or thousands

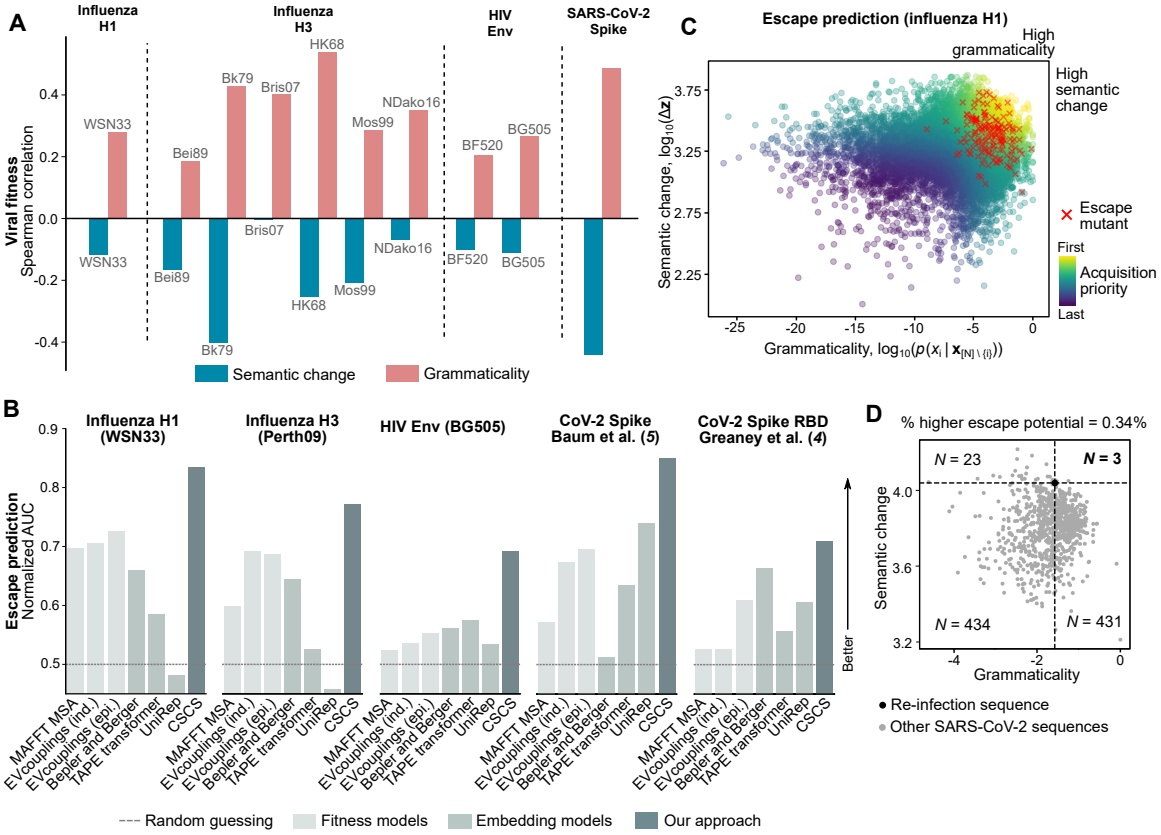

Figure 8-3: Biological interpretation of language models predicts escape.

(**A**) While grammaticality is positively correlated with fitness, semantic change has negative correlation, suggesting that most semantically altered proteins lose fitness. (**B**, **C**) However, a mutation with both high semantic change and high grammaticality is more likely to induce escape. Considering both semantic change and grammaticality enables identification of escape mutants that is consistently higher than that of previous fitness models or generic functional embedding models. (**D**) Across 891 surveilled SARS-CoV-2 Spike sequences, only three have both higher semantic change and grammaticality than a Spike sequence with four mutations that is associated with a potential re-infection case.

of mutants to a given viral protein. We obtained datasets measuring replication fitness of all single-residue mutations to `A/WSN/1933` (`WSN33`) HA H1 [DB16], combinatorial mutations to antigenic site B in six HA H3 strains [WOT+20], or all single-residue mutations to `BG505` and `BF520` HIV Env [HDH+18], as well as a dataset measuring the dissociation constant (Kd) between combinatorial mutations to SARS-CoV-2 Spike receptor binding domain (RBD) and human ACE2 [SGH+20], which we use to approximate the fitness of Spike.

We found that language model grammaticality was significantly correlated with

viral fitness consistently across all viral strains and across studies that performed single or combinatorial mutations (Figure 8-3A and Table A.12), even though our language models were not given any explicit fitness-related information and were not trained on the DMS mutants. Strikingly, when we instead compared viral fitness to the magnitude of mutant semantic change (rather than grammaticality), we observed significant negative correlation in nine out of ten strains tested (Figure 8-3A and Table A.12). This makes sense biologically, since a mutation with a large effect on function is on average more likely to be deleterious and result in a loss of fitness. These results suggest that, as hypothesized, grammatical "validity" of a given mutation captures fitness information, and adds an additional dimension to our understanding of how semantic change encodes perturbed protein function.

8.5.4 CSCS predicts escape

Based on these promising analyses of viral semantics and grammaticality, we therefore sought to test the third part of our hypothesis, namely that combining semantic change and grammaticality enables escape mutation prediction. Our experimental setup initially involves making, *in silico*, all possible single-residue mutations to a given viral protein sequence; then, each mutant is ranked according to the CSCS objective that combines semantic change and grammaticality. We validate this ranking based on enriched CSCS acquisition of experimentally verified mutants that causally induce escape from neutralizing antibodies. Three of these causal escape datasets used DMS followed by antibody selection to identify escape mutants to `WSN33` HA H1 [DLB18], `A/Perth/16/2009` (`Perth09`) HA H3 [LEZ+19], and `BG505` Env [DAW+19]. The fourth identified escape mutations to SARS-CoV-2 Spike using natural replication error after in vitro passages under antibody selection (5), while the fifth performed a DMS to identify mutants that affect antibody binding to yeast-displayed Spike RBD [BFW+20].

We computed the area under the curve (AUC) of acquired escape mutations versus the total acquired mutations. In all five cases, escape prediction with CSCS resulted in both statistically significant and strong AUCs of 0.83, 0.77, 0.69, 0.85, and 0.71 for

H1 `WSN33`, H3 `Perth09`, Env `BG505`, Spike, and Spike RBD, respectively (one-sided permutation-based $P < 1 \times 10^{-5}$ in all cases) (Figure 8-3B and Table A.13). We emphasize that none of the escape mutants are present in the training data, and we did not provide the model with any explicit information on escape, a challenging problem setup in machine learning referred to as "zero-shot prediction" [RWC+19a].

Crucially, in support of our hypothesis, the escape AUC strictly decreases when ignoring either grammaticality or semantic change, evidence that both are useful in predicting escape (Figures 8-3C, A-24, and Table A.13). Note that while semantic change is negatively correlated with fitness, it is positively predictive (along with grammaticality) of escape (Table A.13); the analogous biological interpretation is that functional mutations are often deleterious but, when fitness is preserved, they are associated with antigenic change and subsequent escape from immunity.

For a benchmark comparison, we also tested how well alternative models of fitness (each requiring MSA preprocessing) or of semantic change (pretrained on generic protein sequence) predict escape, noting that these models were not explicitly designed for escape prediction. Additional details regarding these benchmark methods are provided in Appendix E.1. We found that CSCS with our viral language models was substantially more predictive of causal escape mutants in all four viral proteins (Figure 8-3B). Moreover, the individual grammaticality or semantic change components of our language models often outperformed the corresponding benchmark models (Table A.13), demonstrating the value of nonlinear, high-capacity fitness models or of virus-specific, finetuned semantic embedding models, respectively. In total, our results provide strong empirical support for the hypothesis that both semantic change and grammaticality are useful for escape prediction.

8.5.5 Structural patterns from sequence alone

A notable aspect of our results is that, though viral escape is mechanistically linked to a viral protein's structure, our models are trained entirely from sequence and bypass explicit structural information altogether (which is often difficult to obtain). Given our validated escape prediction capabilities, we wanted to look at our model's escape

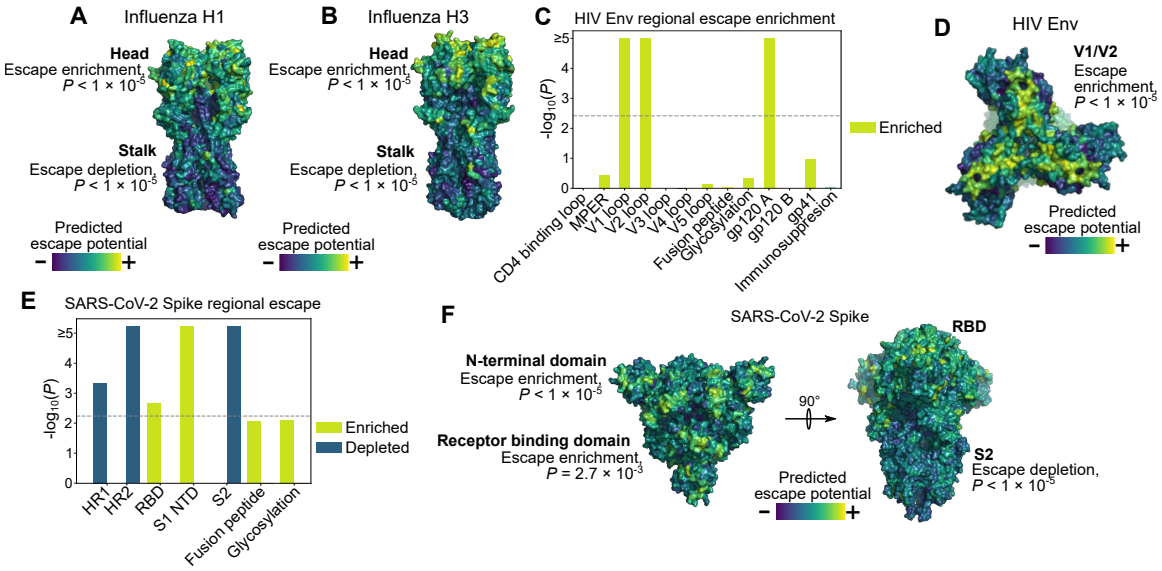

Figure 8-4: Structural localization of predicted escape potential.

(**A**, **B**) HA trimer colored by escape potential. (**C**) Escape potential P-values for HIV Env; gray dashed line indicates statistical significance threshold. (**D**) The Env trimer colored by escape potential, oriented to show the V1/V2 regions. (**E**, **F**) Potential for escape in SARS-CoV-2 Spike is significantly enriched at the N-terminal domain and receptor binding domain (RBD) and significantly depleted at multiple regions in the S2 subunit; gray dashed line indicates statistical significance threshold.

predictions in the context of three-dimensional protein structure to see if our model was able to learn structurally relevant patterns from sequence alone. We used CSCS to score each residue based on predicted escape potential, from which we could visualize escape potential across the protein structure and quantify significant enrichment or depletion of escape potential based on a null distribution constructed by permuting CSCS acquisition scores across positions.

For both HA H1 and H3, we found that escape potential is significantly enriched in the HA head and significantly depleted in the HA stalk (Figures 8-4A,B and A-25), consistent with existing literature on HA mutation rates and supported by the successful development of anti-stalk broadly neutralizing antibodies [EBE+09, KCC+16]. Also consistent with existing knowledge is the significant enrichment of escape mutations in the V1/V2 hypervariable regions of Env (Figure 8-4C,D and A-25) [SWLO06]. An important point is that our model only learns escape patterns that can be linked to mutations, rather than post-translational changes like glycosylation

that contribute to HIV escape [WDW+03], which may explain the lack of statistically significant escape potential assigned to Env glycosylation sites (Figure 8-4C).

8.6 Application note: Escape potential of SARS-CoV-2 re-infection

One highly practical and very important question about controlling COVID-19 is how often re-infection occurs, since a high rate of re-infection would raise questions about vaccine efficacy. We therefore estimated the antigenic change and fitness of a set of four mutations to the SARS-CoV-2 Spike associated with a reported re-infection event [THI+20]. In doing so, we show how language modeling can characterize sequence changes beyond single-residue mutations, e.g., from accumulated replication error or recombination [SC20], though our approach is agnostic to how a sequence acquires its mutations.

To do so, we obtained the Spike sequences from the reported first and second rounds of SARS-CoV-2 infection of a single patient from To et al. [THI+20]. We computed the re-infection sequence's grammaticality as the average log language model probability across the individual mutant positions and the semantic change (relative to the first infection sequence) as the ℓ_1 distance between the original and mutant language model embeddings. The re-infection Spike sequence has four mutated positions relative to the first infection sequence. We note that the re-infection sequence was not present in the training corpus. We compared the predicted semantic change and grammaticality of the re-infection sequence to those of 891 unique SARS-CoV-2 Spike sequences from our training corpus, where semantic change was similarly defined with respect to the first infection sequence from To et al.

Additionally, we compared the re-infection sequence to a null distribution of 100 million sequences with four mutations compared to the first infection sequence. The mutations were chosen uniformly at random across each position and across the amino acid alphabet. A sequence from the null distribution was considered to have higher

escape potential than the re-infection sequence if it had both higher fitness and higher semantic change.

As positive controls, we performed the same analysis on sequences in which SARS-CoV-2 Spike RBD was artificially replaced *in silico* with the RBD-ACE2 contacts of bat coronavirus RaTG13 (eight mutated positions relative to wildtype) or of SARS-CoV-1 (twelve mutated positions relative to wildtype), creating antigenically dissimilar sequences while preserving ACE2 binding, albeit with lower affinity [SGH$^+$20]. We note these *in silico* "recombinant" sequences are also not present in the training corpus. We again compared the semantic change and grammaticality of these recombinant sequences to the 891 surveilled Spike sequences in our training corpus (Figure A-24B), as described in the previous paragraph; here, semantic change was defined relative to the wildtype Spike sequence.

Among 891 other unique, surveilled Spike sequences, we found that only three (0.34%) represent both higher semantic change and grammaticality (Figure 8-3D). We estimate significant escape potential of these four mutations (random mutant null distribution $P < 1 \times 10^{-8}$) and we observed similar patterns for known antigenically dissimilar sequences (Figure A-24B). Our analysis suggests a way to quantify the escape potential of interesting combinatorial sequence changes, like those from possible re-infection [THI$^+$20], and calls for more information relating combinatorial mutations to re-infection and escape.

More broadly, we anticipate that more complex models of the *distribution* of viral sequence evolution (like language models) will play an important role in understanding and controlling viral pandemics. A distributional approach to viral evolution may yield additional insight over current approaches based on phylogenetic reconstruction, which assumes sequence divergence. Phylogenetic assumptions fall short when viruses have more complex evolutionary strategies, like influenza and HIV. More complex models, therefore, give hope that they can lead to better vaccine design that can ultimately eradicate pandemic viruses.

Chapter 9

Perspectives

Yes, she thought, laying down her brush in extreme fatigue,
I have had my vision.

—Virginia Woolf, *To the Lighthouse* (1927)

In this thesis, we have seen that intelligent algorithms, in concert with old and new biological techniques, can make a nontrivial contribution to understanding and fighting infectious disease. We have also, hopefully, conveyed some sense of the excitement that permeates the field of computational biology as we enter the third decade of the twenty-first century. The possibilities for creative new approaches are endless; the problems are complex but many solutions seem within reach; and the opportunity to benefit the global human population offers a strong guiding force.

In this final chapter, we hope to outline ways to move the field forward using the ideas outlined in this thesis as a launching pad. We organize these future directions from the most immediate to the long-term. In particular, we start with ways to remix themes in this thesis into new projects like influenza forecasting, antibody cocktail selection, and mosaic vaccine design. We then discuss higher-level research topics in computational biology, with implications not only for infectious disease but also many other areas as well, like evolution and systems biology.

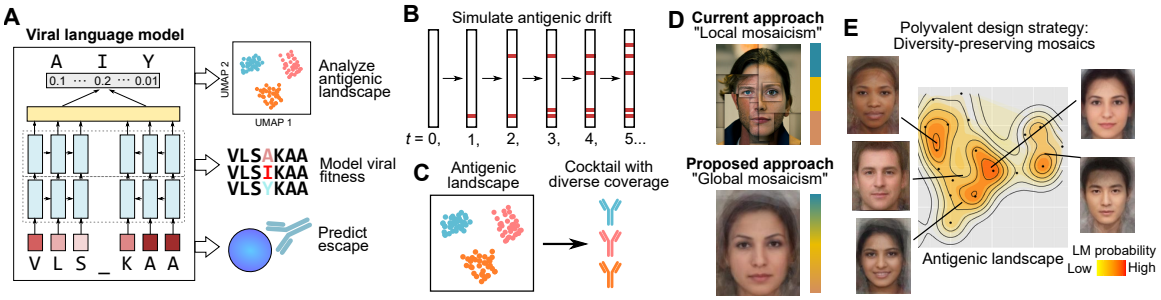

Figure 9-1: Examples of near-term directions.

(**A**) Language models (LMs) learn functional properties from sequence variation. (**B**) LMs can iteratively propose point mutations to simulate drift. (**C**) Antigenic embeddings can guide therapies that maximize diversity. (**D**) LMs enable a global notion of the "maximum likelihood" sequence that avoids the unnatural discontinuities of current design strategies. (**E**) Quantifying antigenic diversity can also inform design of polyvalent vaccines.

9.1 The near term

Many of the ideas described in this thesis can be combined and remixed into new projects with high impact. In the near term, there are many exciting applications for understanding evolutionary diversity of pathogens.

In Chapter 8, we saw that a machine learning algorithm called a language model (LM) can learn functional patterns from viral sequence variation alone. For a sequence S, an LM learns a probability $p(S)$ where a high $p(S)$ means that S has sequence patterns that are commonly observed in the training data (e.g., among surveilled sequences), whereas a low $p(S)$ indicates a less likely sequence. We also saw how $p(S)$ significantly correlates with viral fitness (12), since fitter sequences are more commonly observed. A second important capability provided by a neural LM is that it learns a sequence "embedding," which is a vector of numbers where more similar embeddings encode more similar proteins, providing a way to quantify antigenic diversity (12). We used both the LM probabilistic output and the LM semantic embeddings to predict viral escape mutations with high accuracy and without any prior information about protein structure or function [HZBB21] (Figure 9-1A). Language modeling of viral sequences is a powerful new technique and has the potential to inform antiviral and vaccine design.

In Chapter 4, we also saw an algorithm that can select a combinatorial subset of a massive dataset in a way that maximizes the diversity of the subset [HCD+19]. While this was originally applied to sampling diverse transcriptomes, the same principles can also sample diverse viral states. And in Chapter 7, we saw algorithms that enable closer human-algorithm experimental cooperation [HBB20] so that even if a given machine learning prediction fails, it can be reincorporated into the algorithm to improve future predictions. This feedback loop facilitates principled exploration of combinatorially large biological spaces.

Listed below are some practical examples of projects that make use of some or all of these principles. This by no means an exhaustive list, as the ideas are general, can be applied beyond the realm of infectious disease, and the number of possible extensions is only bounded by creativity.

9.1.1 Forecasting antigenic drift

One extension of the viral LM work is to simulate antigenic drift. In each *in silico* "generation," the LM proposes a point mutation that preserves fitness but induces high antigenic change. These point mutations will accumulate over multiple generations (Figure 9-1B). Adding randomness into each generation enables many simulated mutational paths that all start at the same sequence. To validate, we will start the simulation at a historical influenza HA sequence and run many (~10^6) simulations for many (~10^2 to 10^3) generations.

In silico cross validation would train on sequences from a given time period and test on a separate period, with success determined as high sequence/antigenic similarity between simulated drift and actual drift. Such an approach is not unlike weather forecasting, which also runs many simulations to determine the probability of an outcome (e.g., chance of rain). Accurately forecasting mutational dynamics could then inform vaccine development, e.g., vaccinating against a likely future mutant of a current viral strain.

9.1.2 Designing antiviral cocktails

Quantifying antigenic diversity with viral LMs also enables therapeutic design that maximizes coverage of this diversity. In LM embedding space, more similar antigens cluster closer together, and preliminary follow-up data suggests that such clusters have significant differential binding to various monoclonal antibodies. These patterns can therefore help design antibody cocktails that are robust to escape [BFW+20, GSG+20]. First, different antigenic variants will be linked to different antibodies, e.g., by identifying mutants that affect binding [GSG+20]. These variants can be embedded into the LM's antigenic landscape in which, building off of previous work on diversity-preserving subsampling [HCD+19], we can select a subset of antibodies that maximizes diversity of neutralized antigens.

Metrics from computational geometry [Hau37] can quantify diversity in antigenic space (Figure 9-1C). The antibody subset should ideally be small to enable easier clinical translation, leading to a constrained optimization problem: maximize diversity with a minimal set of antibodies. Validation would passage a virus *in vitro* with the cocktail [BFW+20, GSG+20] followed by sequencing to identify escape mutations. Ideally, no escape should occur even after many passages.

9.1.3 Engineering polyvalent mosaic vaccines

Finding the sequence S^* that maximizes the LM probability $p(S)$, i.e., the "maximum likelihood" sequence, could be used as a better way to engineer mosaic vaccines [FPT+07, SLZ+10, BOS+10, BSB+13, BTW+18] than current heuristics. These current strategies are based on local sequence variation, e.g., a mosaic based on 9-mer or 15-mer sequence fragments, whereas neural network LMs can capture global patterns across an entire sequence, leading to more natural sequence designs (Figure 9-1D). The maximum likelihood sequence reflects the most frequently observed viral sequence patterns and might therefore elicit a broader immune response. Finding S^* would naively require brute-force enumeration of all possible sequences, but combinatorial search algorithms like a genetic algorithm [FPT+07] or Monte Carlo tree search,

famously used to search for game playing strategies in chess and Go [SHM+16], will eliminate large spaces of improbable sequences. Importantly, LMs generalize to any viral sequence, enabling the same strategy to be used for different antigens (e.g., HIV Gag, Env, or Pol). As a sanity check, *in silico* models that flag biochemically invalid protein sequences will also be applied to the maximum likelihood sequence designs. We also expect that mosaic vaccine candidates that have been confirmed to elicit broad immune responses in primates would have a significantly high $p(S)$, though perhaps not as high as $p(S^*)$.

Most mosaic vaccines are polyvalent [FPT+07], i.e., composed of multiple unique antigens. Because LMs give us a continuous, multidimensional representation of antigenic diversity, we can use a geometric approach to select diverse mosaic antigens. Rather than returning a single S^*, a search algorithm can also return multiple sequences with high $p(S)$, e.g., the top three sequences based on $p(S)$ are used as a trivalent mosaic vaccine. We would also want to select antigens such that no naturally observed antigen is too far away from any vaccine antigen in the LM antigenic embedding space; in computational geometry, this is formalized by minimizing the Hausdorff distance [Hau37] from a set of surveilled antigens to the set of vaccine antigens. Algorithms can efficiently select a diversity-preserving subset of a larger dataset and can be combined with a maximum likelihood antigen design strategy to select a polyvalent vaccine composed of diverse mosaic antigens (Figure 9-1E).

A maximum likelihood polyvalent mosaic vaccine will need to be evaluated *in vivo*, e.g., with rhesus macaque models for an HIV vaccine candidate. The evaluation pipeline will leverage previously reported methods for determining vaccine immunogenicity and protection [FPT+07, SLZ+10, BOS+10, BSB+13, BTW+18]. Should the initial mosaic show no improvement or worse performance, failure can be incorporated as prior information in the model [HBB20], which will then propose a sufficiently different new mosaic.

9.2 The long term

The work in this thesis is also part of a longer term vision for computational biology as a driving force for biological discovery. There are a number of areas in which algorithms can make a particularly effective contribution, which are (briefly) discussed here.

9.2.1 Making sense of combinations

Biology achieves tremendous complexity through combinations; for example, just four nucleotides are sequentially combined to encode all of nature's biodiversity. Solving biological problems requires understanding combinatorial complexity, but most systematic/high-throughput strategies for biological exploration do not scale beyond singleton perturbations (e.g., point mutations, single-gene knockouts, monoclonal antibodies). Exploring combinatorially-large biological search spaces will require intelligent search algorithms, particularly those that can learn from data.

One particular application is in combinatorial drug design with the goal of preventing resistance mutations. A particularly notable success of HIV research since the beginning of the AIDS epidemic in the 1980s is the development of drug cocktails that can push a patient's viral load below the limit-of-detection of antigen or even RNA tests. Similar strategies based on combinations of drugs are used to treat bacterial infections like tuberculosis or to treat cancer via chemotherapy. Selecting the components of a drug cocktail is time-consuming; often, the efficacy of each component must be established individually, followed by additional studies on the drugs in combination.

Rapid generation of combination therapy is an area in which algorithms can make a real contribution. For example, combinatorial diversity of a disease (e.g., different viral strains or tumor types) can be distilled into discrete modes by an unsupervised learning algorithm (e.g., via clustering of a continuous sequence embedding). Each mode of a diverse disease might then have a corresponding drug, e.g., each disease "cluster" is defined by a similar pattern of mutations that is targetable by the same

small molecule. These drugs might also be recommended by data-driven algorithms. Finally, the drug combination is thus selected according to the *distribution* of disease diversity.

Active learning algorithms can also guide rounds of biological experimentation that explore a large, combinatorially complex search space in a principled way. These algorithms would make use of the property that desirable biological phenotypes often have low-dimensional inherent structure and can therefore benefit from an exploration-followed-by-exploitation strategy. As an analogy, gold mining often requires large amounts of exploration to find an area with a small amount of gold, but this often leads to a large gold vein nearby; similarly, existing combinatorial drug treatments might also be refined by an active learning algorithm.

9.2.2 Gamification

Many advances in artificial intelligence research have been driven by the goal of meeting or surpassing human-level performance at games like chess, Go, or StarCraft [SHM+16, VBC+19]. Evolution in a pathogenic context can be thought of as a game, but with much more dire consequences. Viral escape and antibiotic resistance are persistent challenges and demand better ways to respond to adversarial evolution.

Thinking of pathogen-drug co-evolution as a game can lead to new biotechnologies and algorithms. In particular, there is increasing interest in using machine learning to help guide directed evolution. It may be possible to simulate co-evolution in the laboratory such that an algorithm is pitted against an *in vitro* (or even an *in vivo*) model as an adversary. For example, algorithmic generation of viral antigens can be pitted against *in vitro* technologies that simulate antibody generation via recombination and somatic hypermutation. Over multiple experimental rounds, *in vitro* evolution can therefore guide *in silico* exploration of an evolutionary landscape that could in turn provide insight into how escape occurs *in vivo*.

9.2.3 Greater human-algorithm cooperation

Routine cooperation between algorithmic predictions and biological data generation is the ultimate goal of research in computational biology. Importantly, algorithm-guided prediction provides an efficient alternative, in both time and resources, to large-scale screening or manual trial and error. Focused experimental decision-making is especially important in settings where high-throughput screens are not easy or even tractable. For example, a researcher might first obtain a training dataset with a tractable experiment (for example, a biochemical assay, or a single-gene reporter readout) and follow up a few machine-guided predictions with more complex experiments (for example, involving pathogenic models like Mtb-infected macrophages, or more complex designs like a high-throughput single-cell profiling experiment).

The work in this thesis dealing with algorithmic biological discovery (Chapter 7) has mostly focus on "exploitation," i.e., prioritizing more confident examples that are likelier to yield positive results. However, researchers can also use algorithms to guide "exploration" of highly novel biological regimes. Techniques like uncertainty will help researchers control the "exploration/exploitation" tradeoff to choose experiments that tolerate a higher risk of failure in order to probe novel regimes. For example, in the drug discovery setting, novelty is important since human-designed drugs are often appraised based on their creativity (in addition to effectiveness). Novelty is also important across biological domains, such as designing artificial proteins not found in nature or discovering new transcriptional circuits.

Although initializing a model with some training data is helpful, it is also possible to begin with zero training data (all predictions might therefore begin as equally uncertain). As more data is collected, a sample-efficient model with uncertainty can progressively yield better and more confident predictions. This is the iterative cycle of computation and experimentation at the heart of active learning, for which we provide a proof-of-concept in this thesis.

More generally, we anticipate that iterative experimentation and computation will have a transformative effect on the experimental process. In addition to learning from

high-throughput datasets, we also envision learning algorithms working intimately alongside bench scientists as they acquire new data, even on the scale of tens of new datapoints per experimental batch. This line of work may ultimately lead to algorithms that propose biological hypotheses that are competitive with (or even exceed) those generated by human creativity.

Ephesians 3:20–21

Appendix A

Supplementary figures and tables

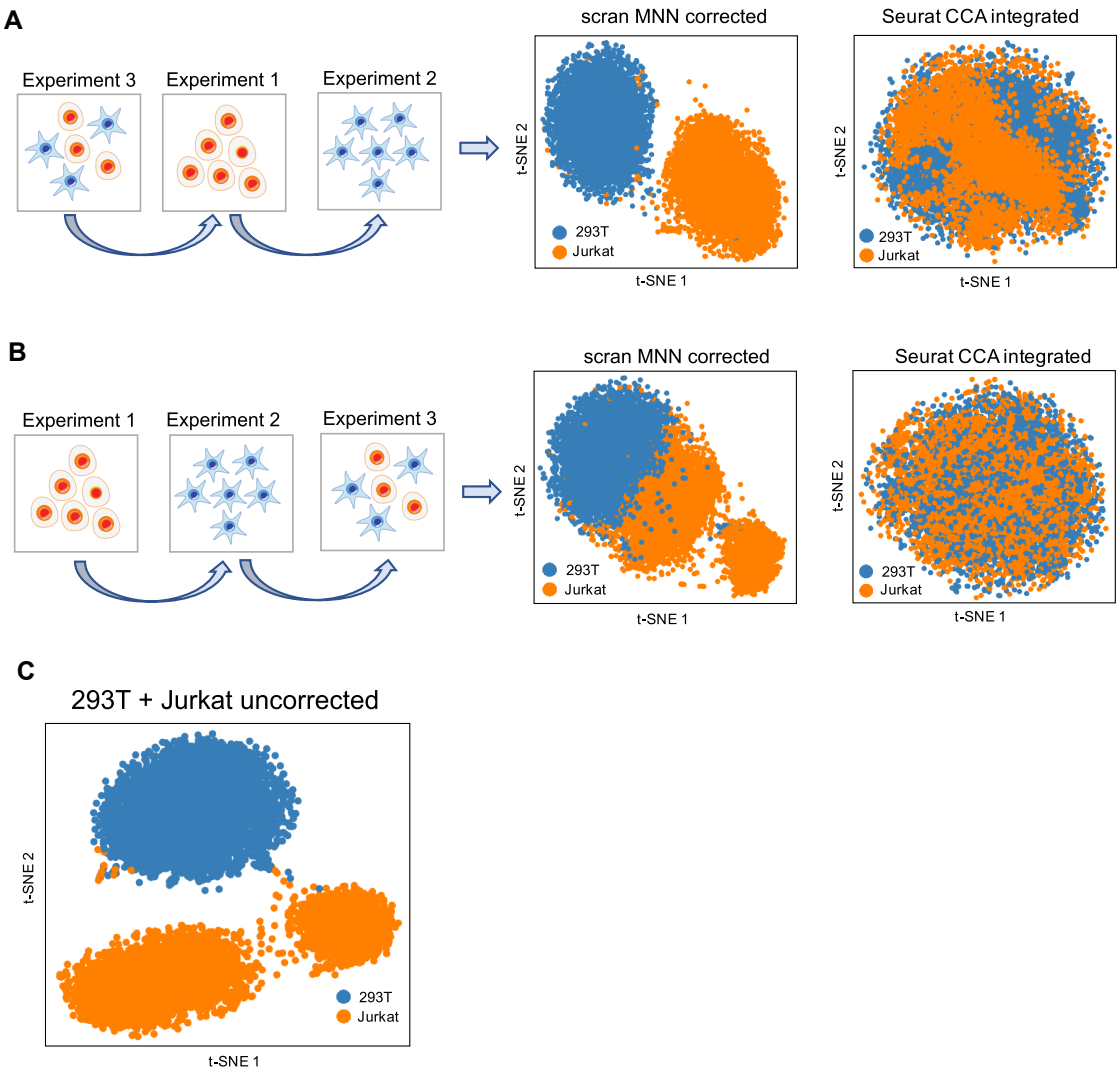

Figure A-1: Previous methods are sensitive to integration order.

(**A**) When a mixture dataset of 293T cells and Jurkat cells is chosen as the first reference dataset (n = 3,388 cells), scran MNN correctly integrates a second dataset of Jurkat cells (n = 3,257) and a third dataset of 293T cells (n = 2,885 cells). (**B**) When given the two datasets of 293T cells and Jurkat cells first, scran MNN incorrectly merges the two cell types together into a single cluster. Integration by scran MNN requires its first dataset to share at least one cell type with all other datasets that are successively integrated, which may not be a reasonable assumption. Seurat CCA was unsuccessful at integrating these three datasets in both cases. (**C**) Without correction, Jurkat cells cluster by batch instead of by cell type.

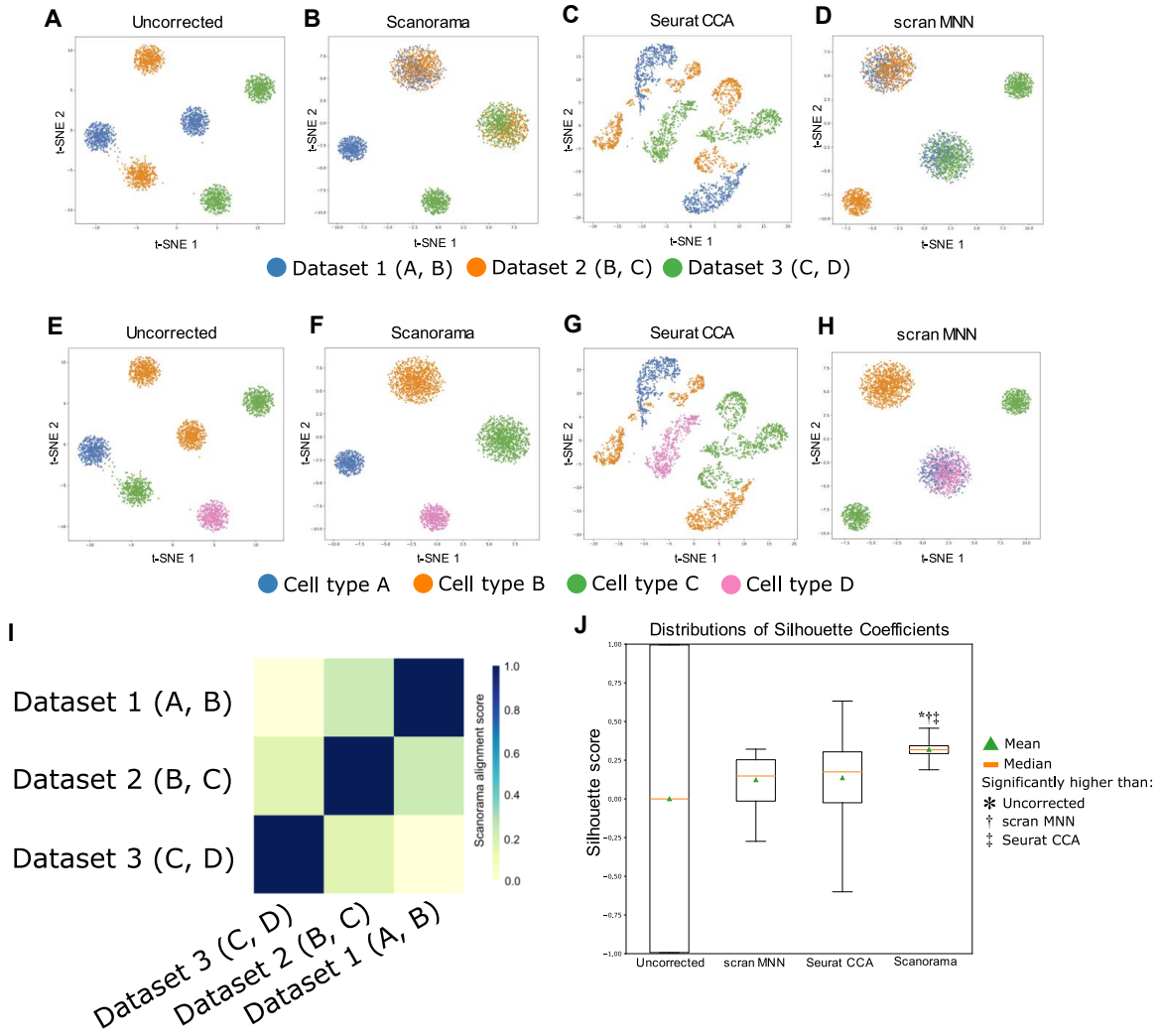

Figure A-2: Comparison of scRNA-seq integration methods on simulations.

(**A**-**H**) We use the Splatter package to simulate three datasets with four cell types in total, where dataset 1 has cell types A and B, dataset 2 has cell types B and C, and dataset 3 has cell types C and D. In each dataset, we assign cells to a cell type with a 50/50 probability. Each dataset contains 1,000 cells. The Splatter simulation also generates batch effects between datasets such that without batch correction cells cluster by both dataset and batch (**A**, **E**). For Seurat CCA and scran MNN, datasets are aligned in numerical order. Scanorama correctly aligns the same cell types together (**B**, **F**), whereas scran MNN incorrectly merges cell types A and D and does not merge cell type C across batches (**D**, **H**). Seurat CCA is unableto merge the datasets together (**C**, **G**). (**I**) Scanorama alignment scores find the correct pairwise matches between the simulated cell types. (**J**) Scanorama has significantly improved Silhouette scores than the uncorrected data (independent, two-sided t-test $P < 5 \times 10^{-324}$; n = 3,000 cells), scran MNN ($P = 1 \times 10^{-40}$), and Seurat CCA ($P = 3 \times 10^{-37}$). An asterisk (*) indicates a significantly higher Silhouette Coefficient distribution (Bonferroni corrected $P < 0.05$) between Scanorama and no correction, a dagger (†) indicates significance over scran MNN, and a double dagger (‡) indicates significance over Seurat CCA. Boxplot boxes extend from lower to upper quartiles with an orange line at the median and green triangle at the mean; whiskers show the range.

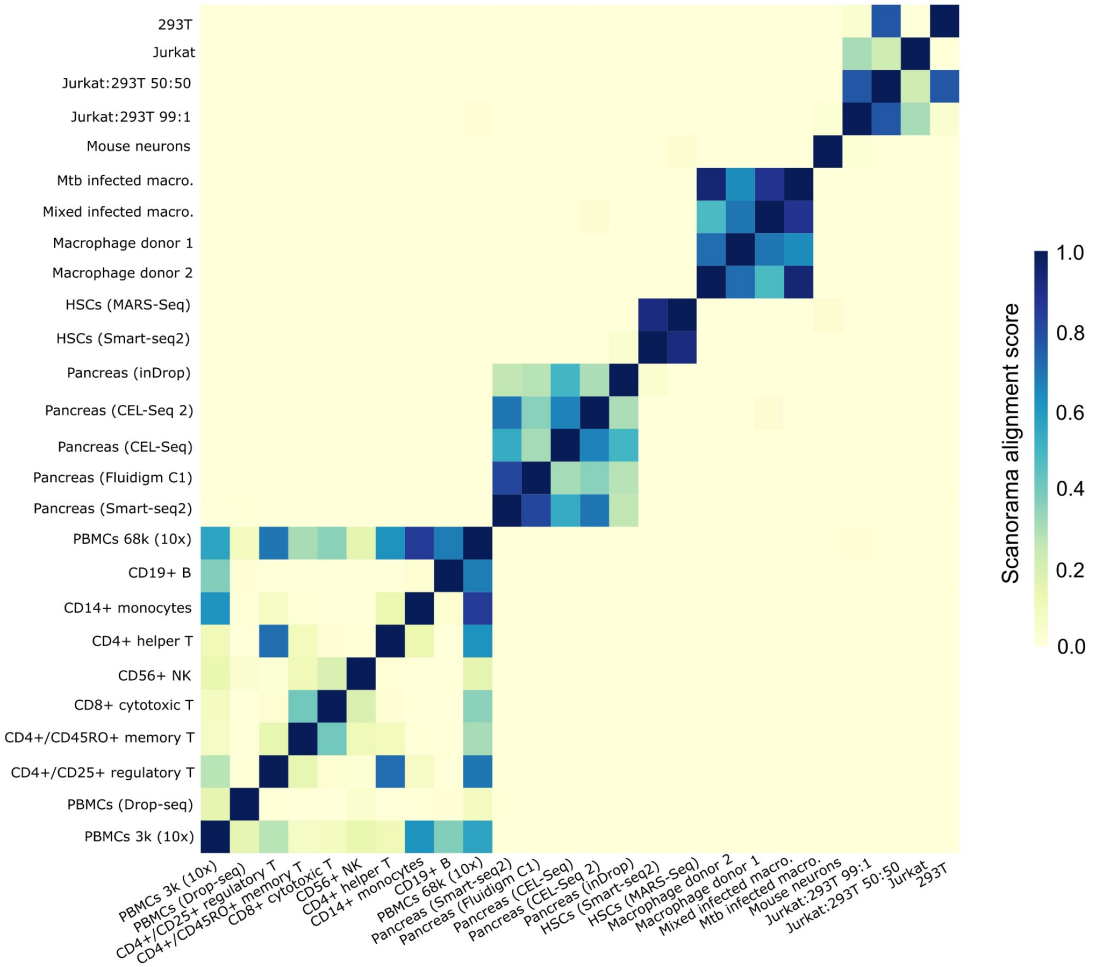

Figure A-3: Scanorama alignment scores across 26 datasets.

Scanorama alignment scores from aligning 26 heterogeneous scRNA-seq datasets reveal high amounts of alignment among biologically similar datasets and alignments scores close to or at zero for datasets that are not biologically similar. Heatmap rows and columns correspond to different datasets and diagonal entries are set to 1.

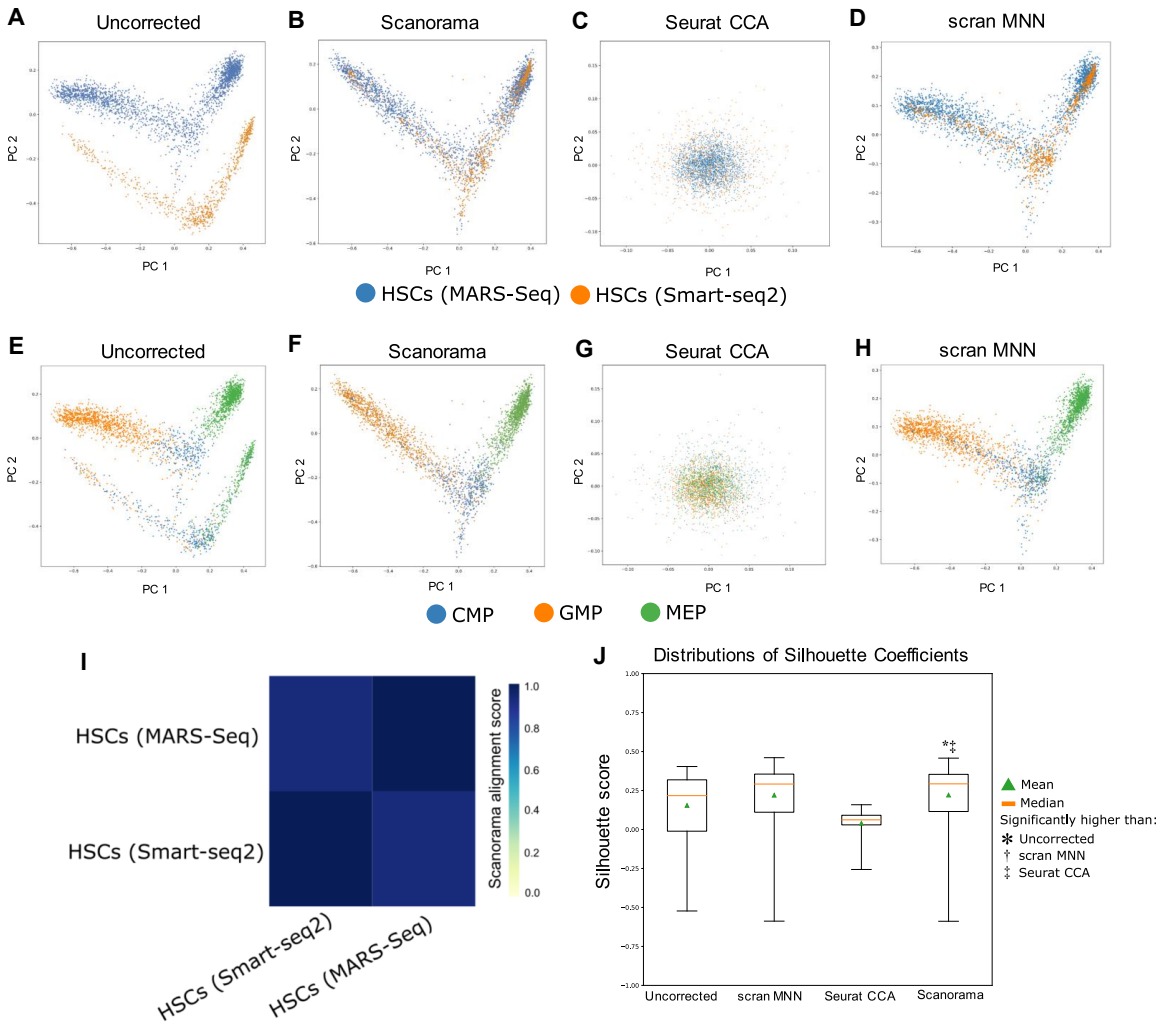

Figure A-4: Comparison of scRNA-seq integration methods on HSCs.

(**A**, **E**) Two datasets of HSCs plotted on the first two principal components (PCs) shows cell separated by batch effects along the second PC; granulocyte-macrophage progenitors (GMP) and megakaryocyte-erythrocytes (MEP) are derived from common myeloid progenitors (CMP). (**B**, **F**) Scanorama removes any significant difference due to experimental batch (natural log likelihood-ratio $= -902$; $n = 3,175$ cells). (**C**, **G**) Seurat CCA overcorrects and places all cell types into a single cluster. (**D**, **H**) scran MNN obtains a similar result to that of Scanorama. (**I**) Scanorama alignments consists of a substantial percentage of the cells in both datasets, as expected. (**J**) Scanorama and scran MNN have similar performance and the same median Silhouette Coefficient (median of 0.28; independent, two-sided t-test $P = 0.14$; $n = 3,175$ cells), but Scanorama has significantly better performance than no correction (median of 0.22; $P = 8 \times 10^{-10}$) and Seurat CCA (median of 0.07; $P = 2 \times 10^{-132}$). An asterisk (*) indicates a significantly higher Silhouette Coefficient distribution (Bonferroni corrected $P < 0.05$) between Scanorama and no correction and a double dagger (‡) indicates significance over Seurat CCA. Boxplot boxes extend from lower to upper quartiles, whiskers indicate range, an orange line indicates the median, and a green triangle indicates the mean(n = 3,175 cells).

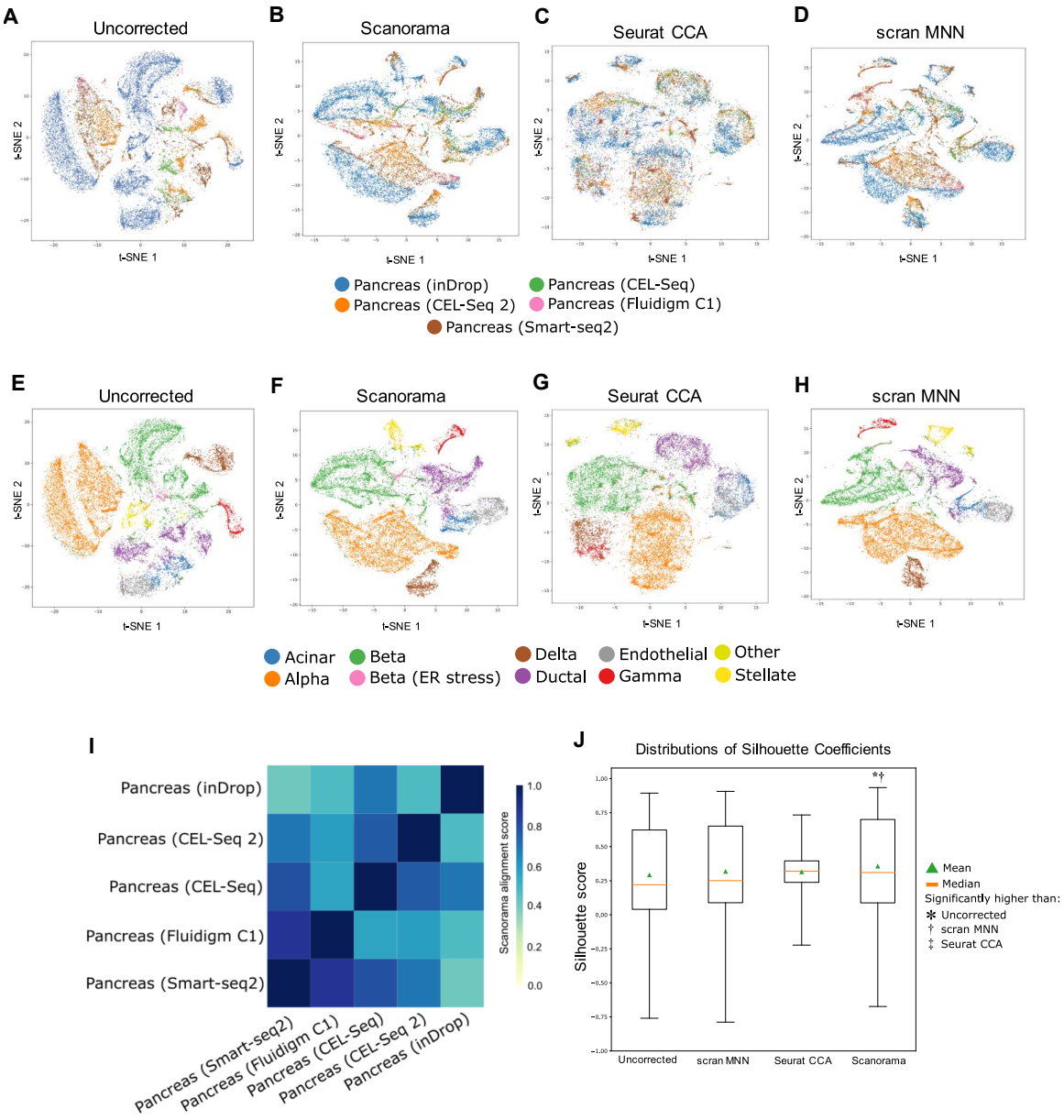

Figure A-5: Comparison of integration methods on pancreas cells.

(**A**, **E**) Pancreatic islets cluster by cell type and batch in the uncorrected setting. (**B-D**, **F-H**) Visually, Scanorama, Seurat CCA, and scran MNN have similar performance in merging cell-type specific clusters together across datasets. (**I**) Scanorama finds substantial overlap among all five pancreatic islet datasets. (**J**) All methods have relatively similar performance, but Seurat CCA has a higher Silhouette Coefficient distribution (compared to Scanorama, independent, two-sided t-test $P = 5 \times 10^{-3}$; n = 15,921 cells) followed by Scanorama, scran MNN ($P = 5 \times 10^{-4}$), and the uncorrected data ($P = 1 \times 10^{-4}$). An asterisk (*) indicates a significantly higher Silhouette Coefficient distribution (Bonferroni corrected $P < 0.05$) between Scanorama and no correction and a dagger (†) indicates significance over scran MNN. Boxplot boxes extend from lower to upper quartiles, whiskers indicate range, an orange line indicates the median, and a green triangle indicates the mean (n = 15,921 cells).

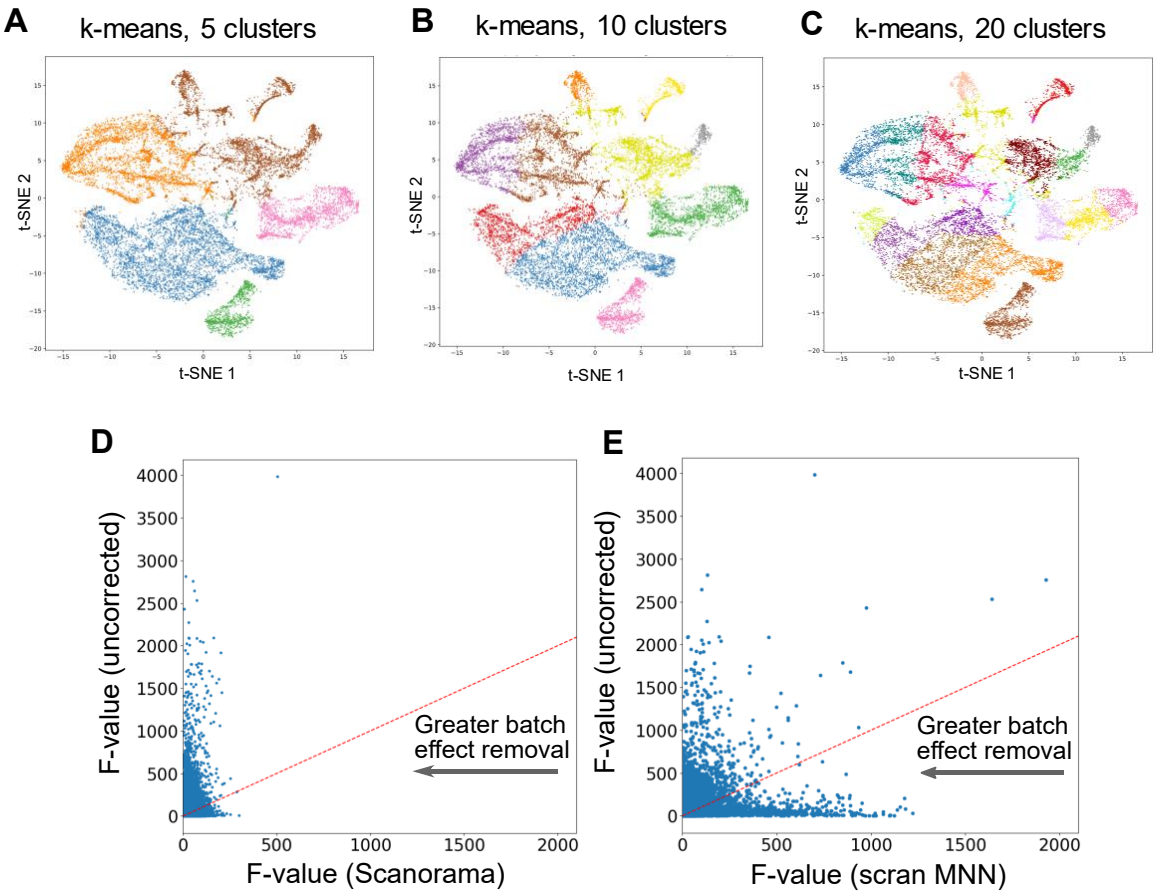

Figure A-6: Batch correction quality on pancreatic islets.

(**A**-**C**) k-means clustering of datasets integrated with Scanorama result in clusters that are orthogonal to differences due to batch, noting that even smaller sub-clusters do not find dataset-specific structure. (**D**, **E**) Scanorama batch correction of five pancreas datasets results in lower one-way ANOVA F-values compared to scran MNN (we note that this analysis is not applicable to Seurat CCA, which finds integrated embeddings and does not modify gene expression values). Each point represents a gene; results are for 15,369 genes. Closer to the left is better, indicating more similar gene expression distributions after batch correction. The red dashed line indicates equal F-values between uncorrected and corrected datasets.

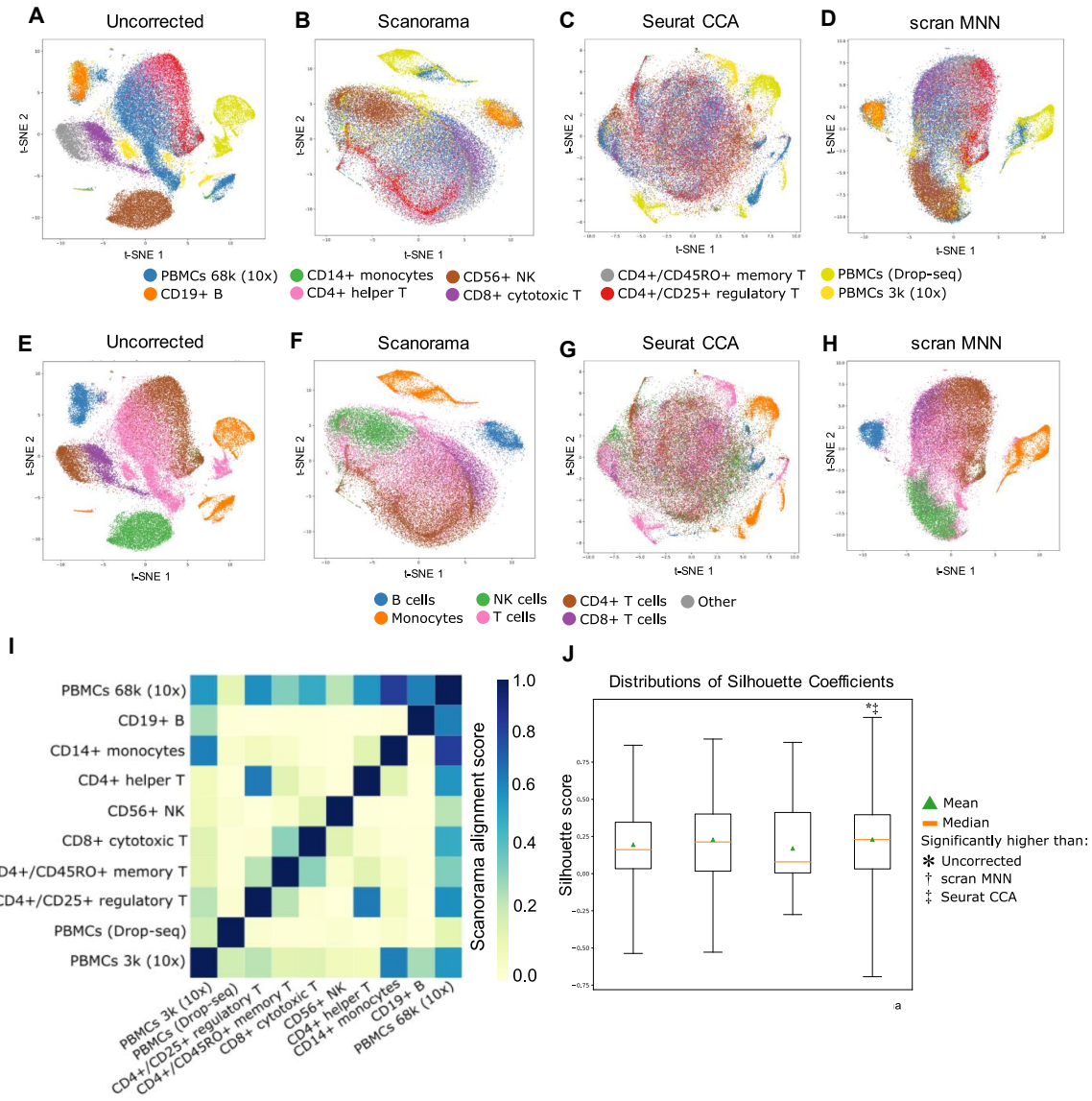

Figure A-7: Comparison of scRNA-seq integration methods on PBMCs.

(**A**,**E**) Without batch correction, PBMC datasets cluster by both cell type and dataset. (**B**, **F**) Scanorama integration results in cells clustering by cell type. (**C**, **G**) Seurat CCA integration results in overcorrection. (**D**, **H**) scran MNN obtains a similar result as that of Scanorama because a large dataset of PBMCs was chosen as the first dataset. We expect performance to degrade if the large dataset were not chosen first. (**I**) Scanorama alignment scores capture relationships between the datasets. (**J**) Scanorama has the highest distribution of Silhouette Coefficients compared to scran MNN (independent, two-sided t-test $P = 0.0011$; $n = 47{,}994$ cells), the uncorrected data ($P = 1 \times 10^{-51}$), and Seurat CCA ($P = 9 \times 10^{-194}$). An asterisk (*) indicates a significantly higher Silhouette Coefficient distribution (Bonferroni corrected $P < 0.05$) between Scanorama and no correction and a double dagger (‡) indicates significance over Seurat CCA. Boxplot boxes extend from lower to upper quartiles, whiskers indicate range, an orange line indicates the median, and a green triangle indicates the mean.

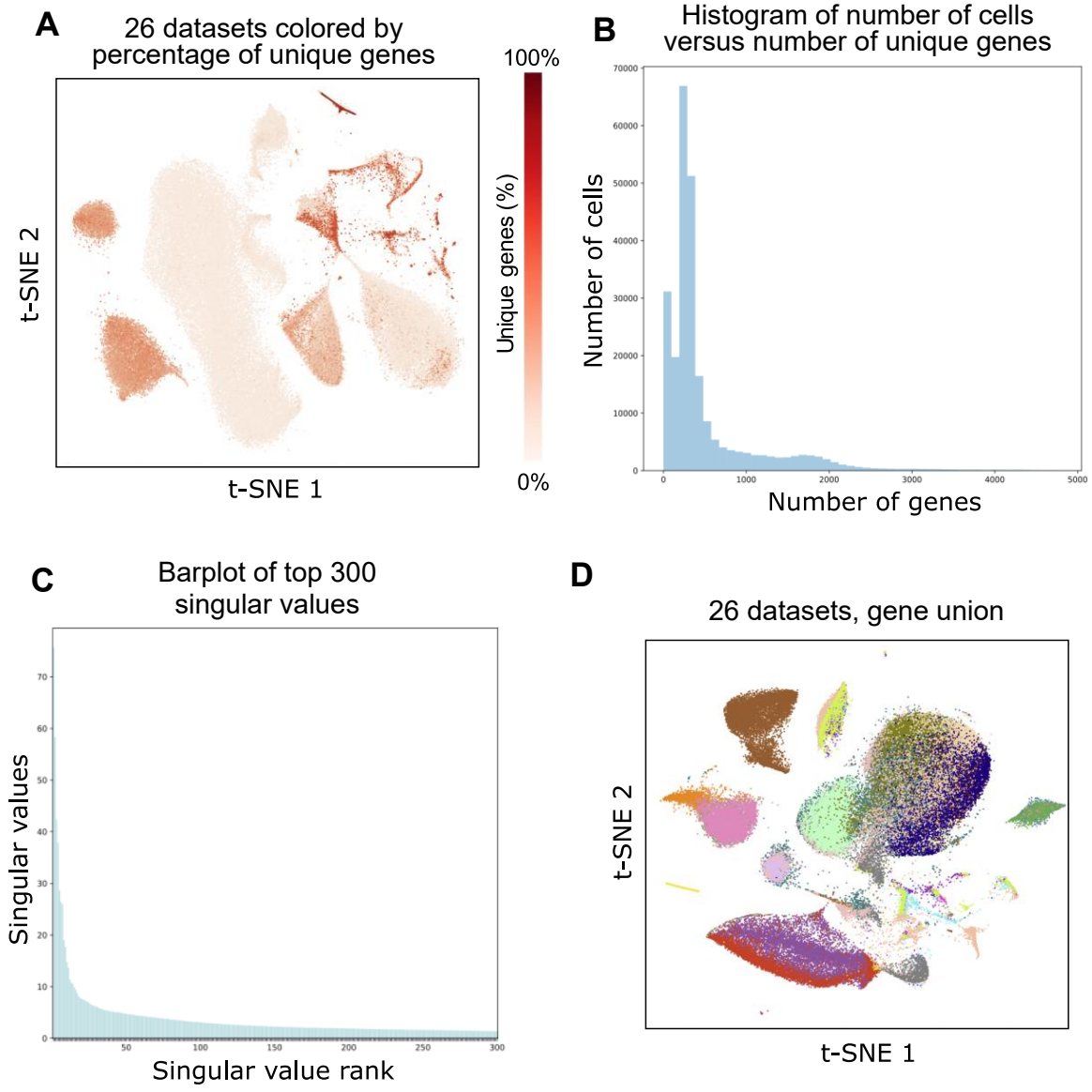

Figure A-8: Twenty-six-dataset quality control.

(**A**) Cells in our experiment integrating 26 diverse datasets (n = 105,476 cells) cluster according to cell type instead of by relative differences in the number of unique genes. E.g., the two HSC datasets are aligned despite different dataset-specific gene percentages (the MARS-Seq dataset has a relatively low average percentage of nonzero genes at 30% versus the Smart-seq2 dataset with an average of 79% nonzero genes), as are the pancreas datasets. (**B**) We observe a bimodal distribution of cells according to their number of unique genes. (**C**) We compute the SVD of the concatenation of the 26 datasets and visualize the top 300 singular values in a bar plot. (**D**) Integrating datasets (n = 105,476 cells) based on the union of all genes (setting unobserved gene expression values to zero) results in similar results as with taking the intersection.

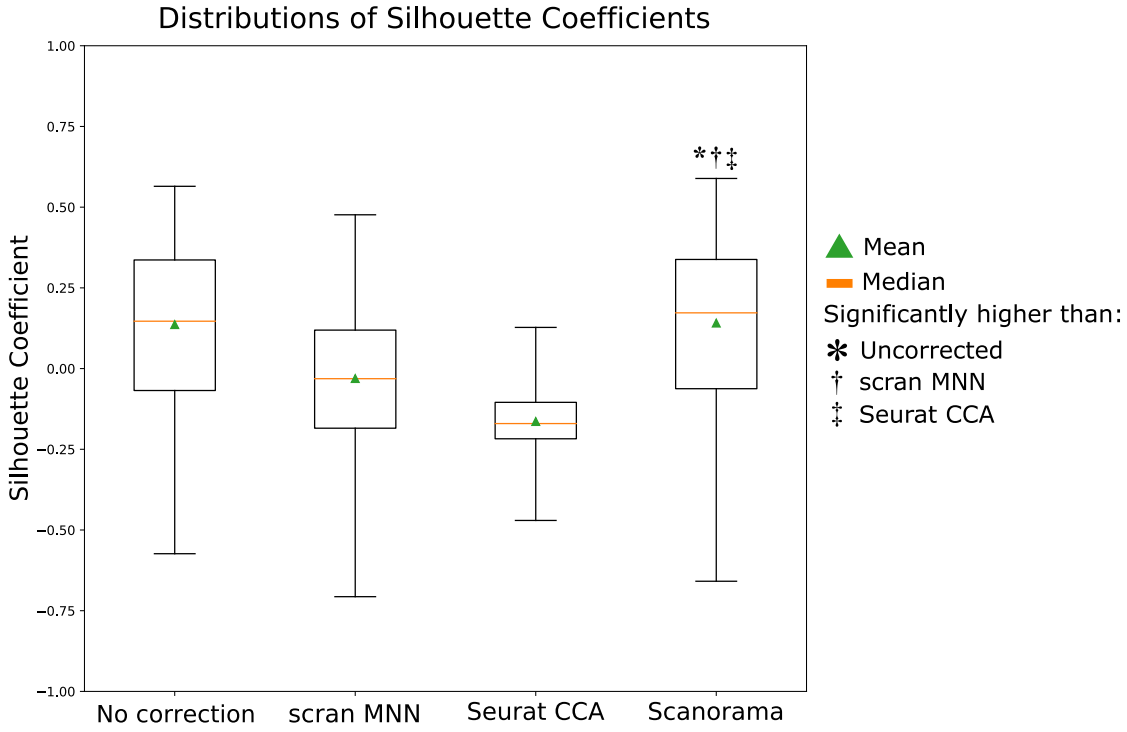

Figure A-9: Silhouette coefficient distributions across 26 datasets.

For our experiment in which we integrate 26 diverse scRNA-seq datasets, we compute Silhouette Coefficients based on Scanorama's integrated embeddings. Scanorama has a significantly higher Silhouette Coefficient distribution compared to scran MNN ($P < 5 \times 10^{-324}$), Seurat CCA ($P < 5 \times 10^{-324}$), and no correction ($P = 4 \times 10^{-6}$) when integrating our collection of 26 datasets containing 105,476 cells. Notably, scran MNN and Seurat CCA have lower median Silhouette Coefficients than if no correction had been applied, indicating large amounts of overcorrection. Boxplot boxes extend from lower to upper quartiles with an orange line at the median and green triangle at the mean; whiskers show the range. P-values are determined using an independent, two-sided t-test (n = 105,476 cells). An asterisk (*) indicates a significantly higher Silhouette Coefficient distribution (Bonferroni corrected $P < 0.05$) between Scanorama and no correction, a dagger (†) indicates significance over scran MNN, and a double dagger (‡) indicates significance over Seurat CCA.

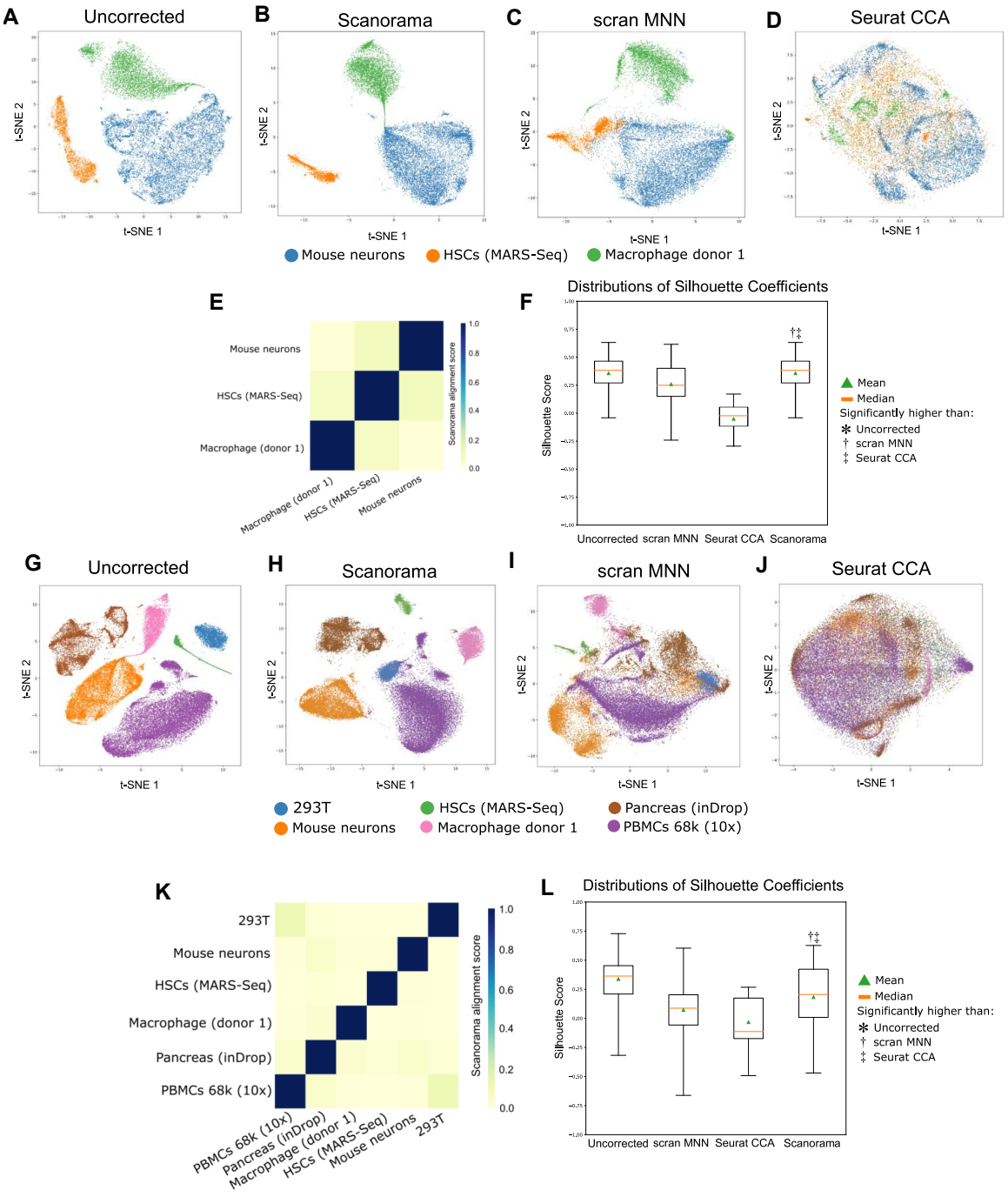

Figure A-10: Integration of datasets with no overlapping cell types.
Boxplot box: lower to upper quartiles; orange line: median; green triangle: mean; whiskers: range; dagger (†): significantly higher Silhouette Coefficient distribution (Bonferroni $P < 0.05$) between Scanorama and scran MNN; double dagger (‡): significance over Seurat CCA. (**A**, **B**, **G**, **H**) No spurious Scanorama alignments between three disparate datasets. (**C**, **D**, **I**, **J**) Other methods prone to overcorrection. (**E**, **K**) Scanorama alignment scores. (**F**) Uncorrected and Scanorama corrected data have the highest Silhouette Coefficients compared to scran MNN (independent, two-sided t-test $P = 7 \times 10^{-252}$) and Seurat CCA ($P < 5 \times 10^{-324}$). (**L**) Scanorama has the least overcorrection compared to scran MNN ($P = 5 \times 10^{-98}$) and Seurat CCA ($P < 5 \times 10^{-324}$).

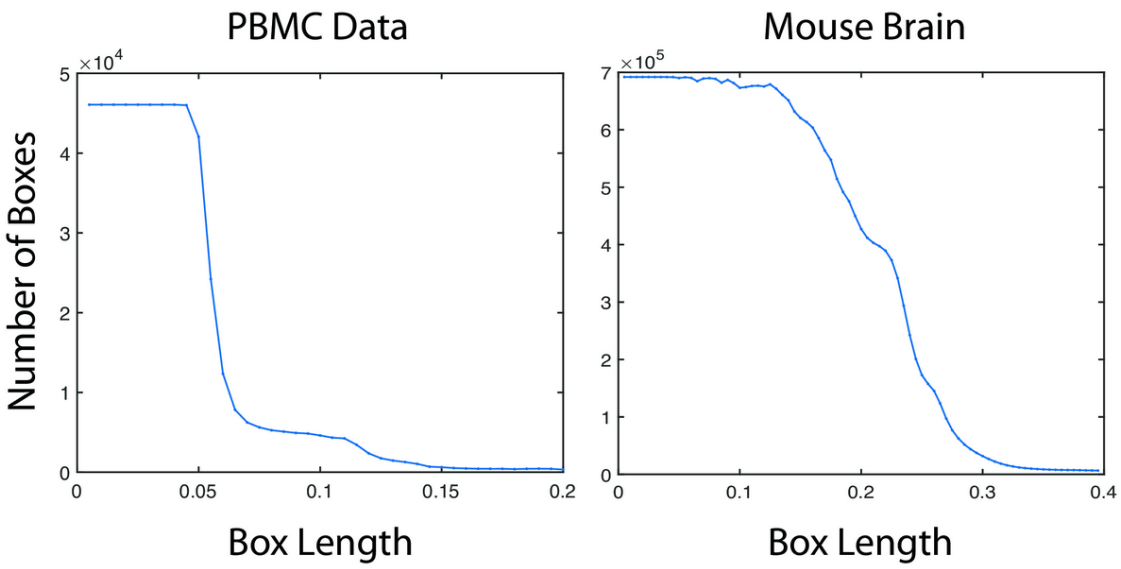

Figure A-11: Near monotonicity of covering boxes with box length.

Cardinality of plaid covering near-monotonically decreases with respect to the length parameter. For PBMC and adult mouse brain datasets, we plotted the number of boxes returned by our plaid covering algorithm as a function of box length provided as input. The overall monotonic relationship allows us to use binary search to find the length at which the plaid cover contains roughly the desired number of boxes.

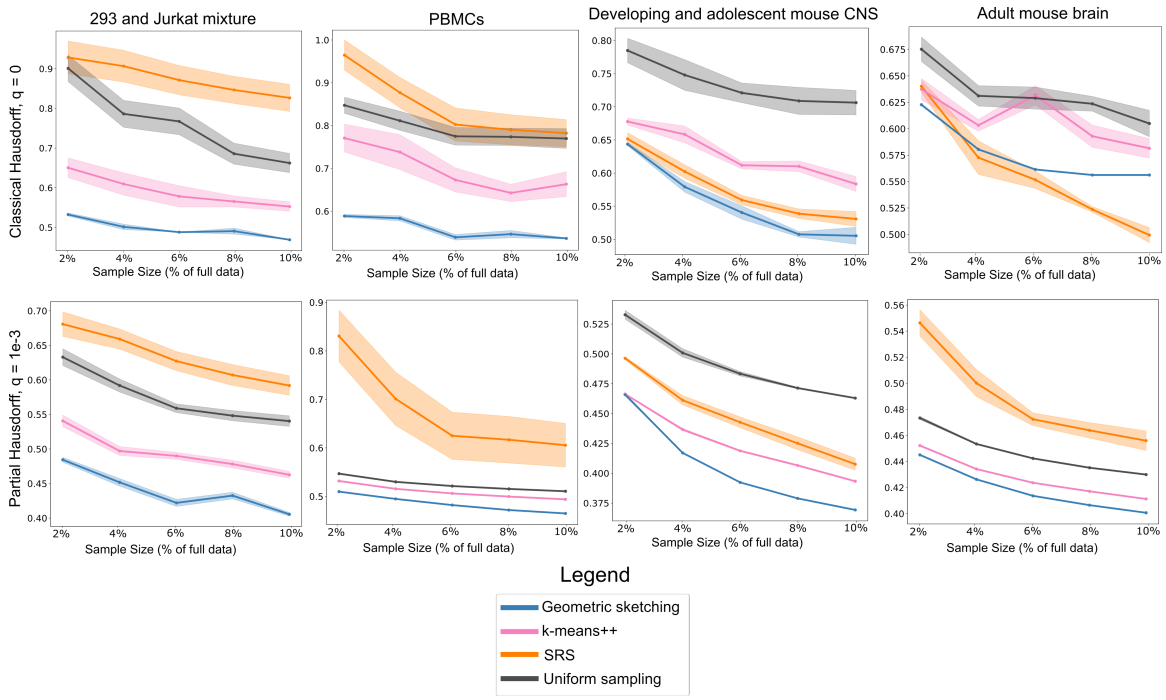

Figure A-12: Partial Hausdorff distance at different parameter cutoffs.

We measured the partial Hausdorff distance at different values of the parameter q (Section 4.5.1), including $q = 1 \times 10^{-4}$ (Figure 4-2), $q = 1 \times 10^{-3}$ and $q = 0$ (the last corresponding to the classical Hausdorff distance). Geometric sketching outperforms all other sampling methods when measured with a robust, partial Hausdorff distance with positive q. Under the classical Hausdorff distance, geometric sketching also outperforms all other sampling methods in almost all cases except for larger sketches in the adult mouse brain dataset due to a single outlier cell, but the anomalous cell was removed when computing more robust Hausdorff distance measures.

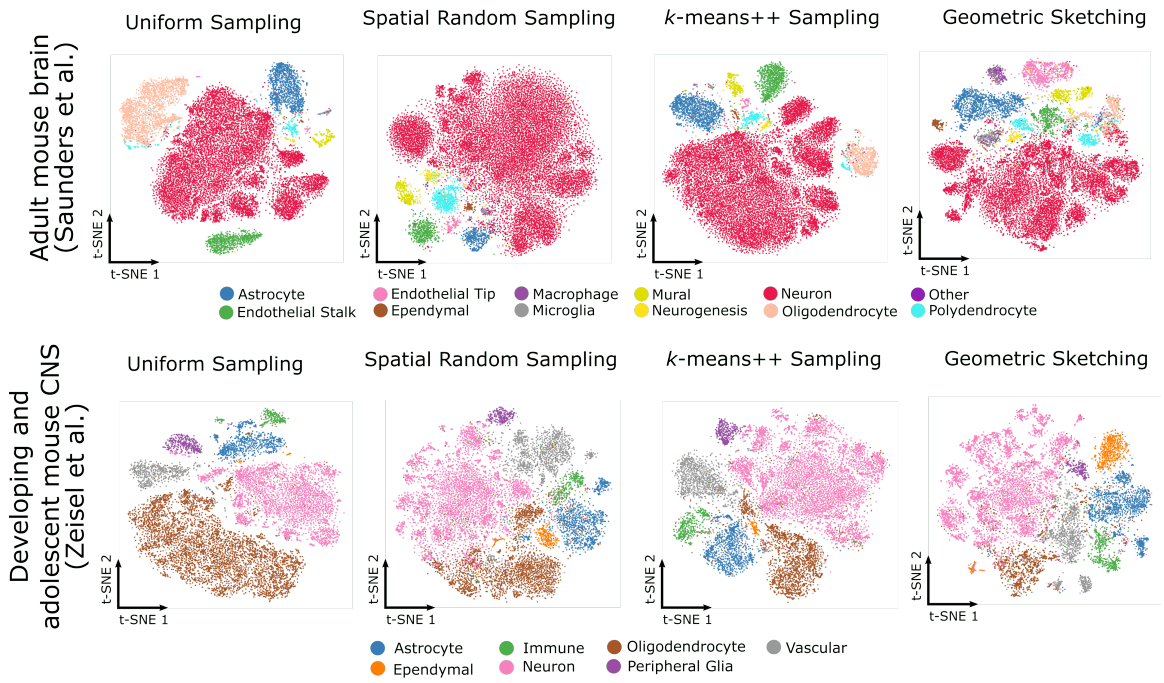

Figure A-13: t-SNE visualizations of large datasets.

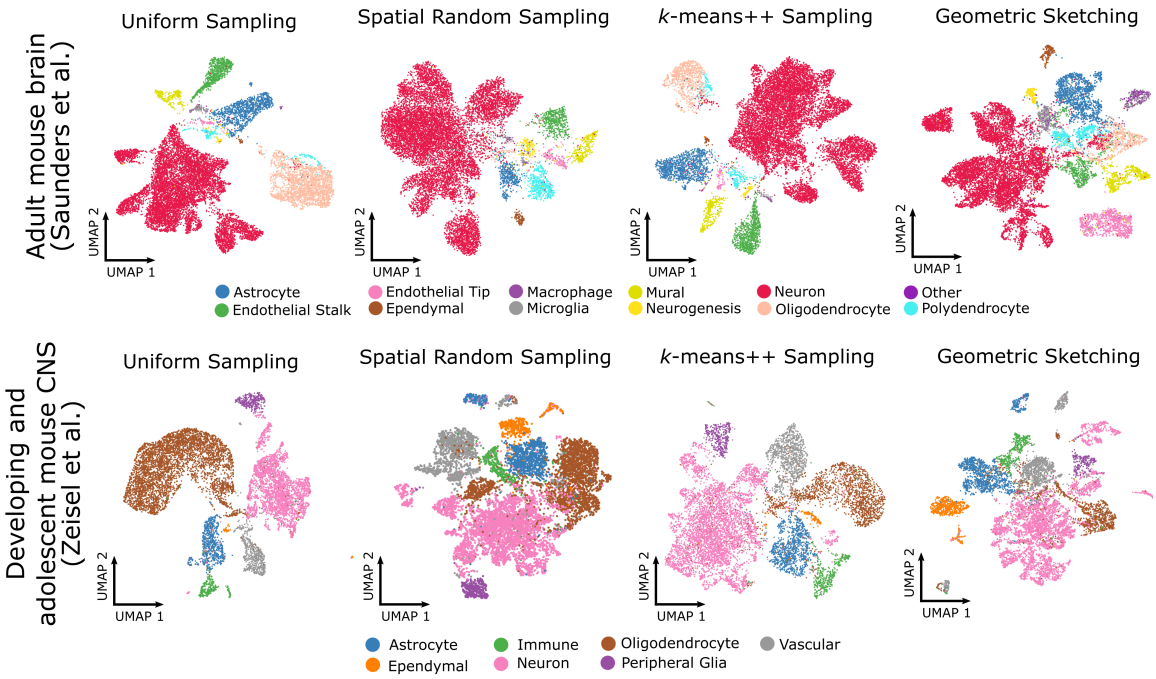

Figure A-14: UMAP visualizations of large datasets.

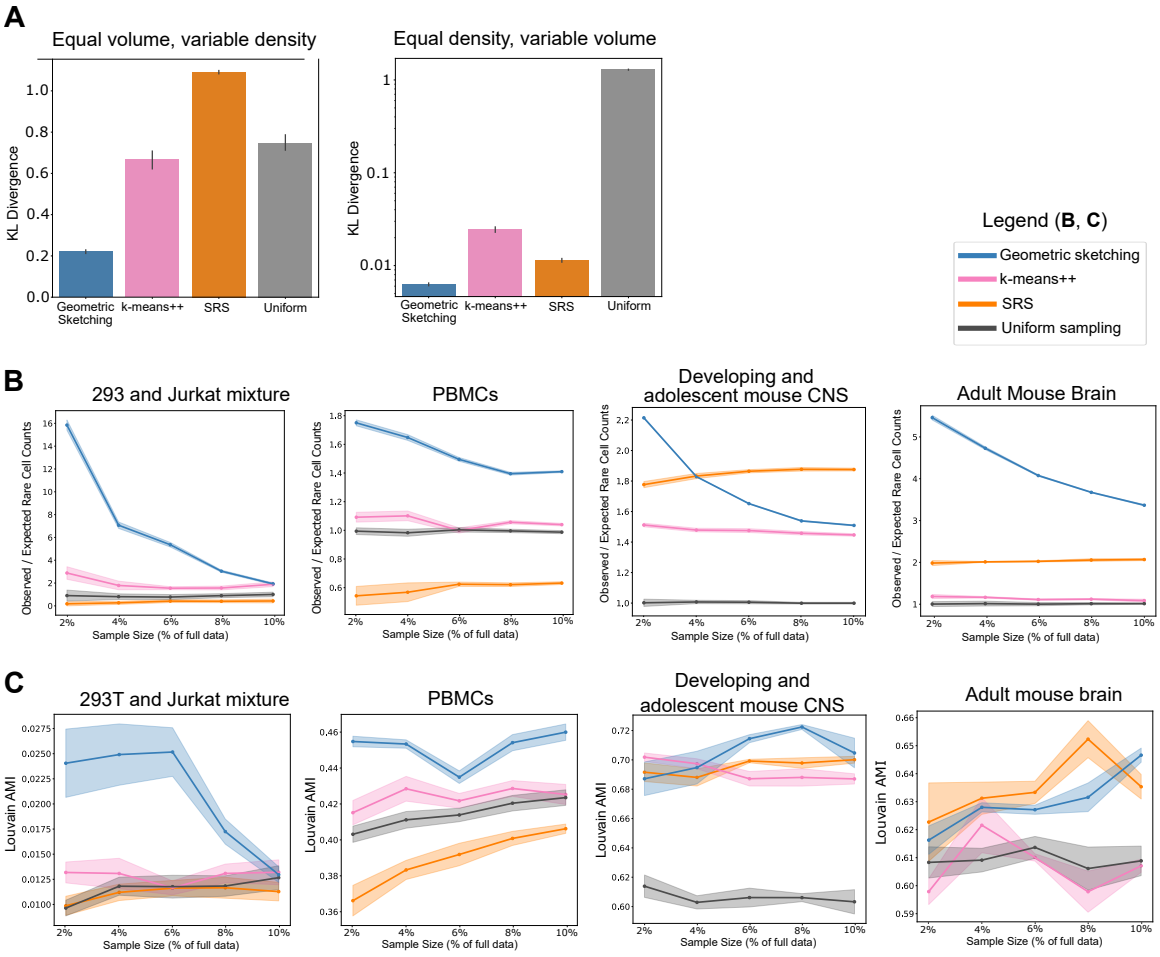

Figure A-15: Additional benchmark comparisons.

(**A**) Sampling with geometric sketching better reflects differences in cluster volume instead of density. Geometric sketching samples from clusters according to the volume of space occupied by each cluster. Bar height indicates means and error bars indicate standard error across 10 random seeds. The y-axis indicates the KL divergence of expected cluster representation based on known cluster volumes compared to observed cluster representation in the subsampled data. Closer to 0 is better (indicates less bias introduced by density). (**B**) Rarest cell types are more represented within a geometric sketch. We assessed overrepresentation of cell types within a sketch by computing the ratio of the observed number of cells over the expected number of cells (assuming uniform sampling probability) for each cell type; we then took the geometric mean of the ratios for the rarest half of all cell types within each dataset. Geometric sketching consistently overrepresents rare cell types and does so more than other sampling strategies in almost all cases. Because we set the number of covering boxes equal to the desired sketch size, as the sketch size increases, the overrepresentation ratio with respect to uniform sampling will converge to unity. (**C**) Unbalanced measurement of clustering recapitulation of biological cell types. Louvain clustering was applied to a sketch, transferred to the full dataset, and then measured for agreement with biological cluster labels using adjusted mutual information.

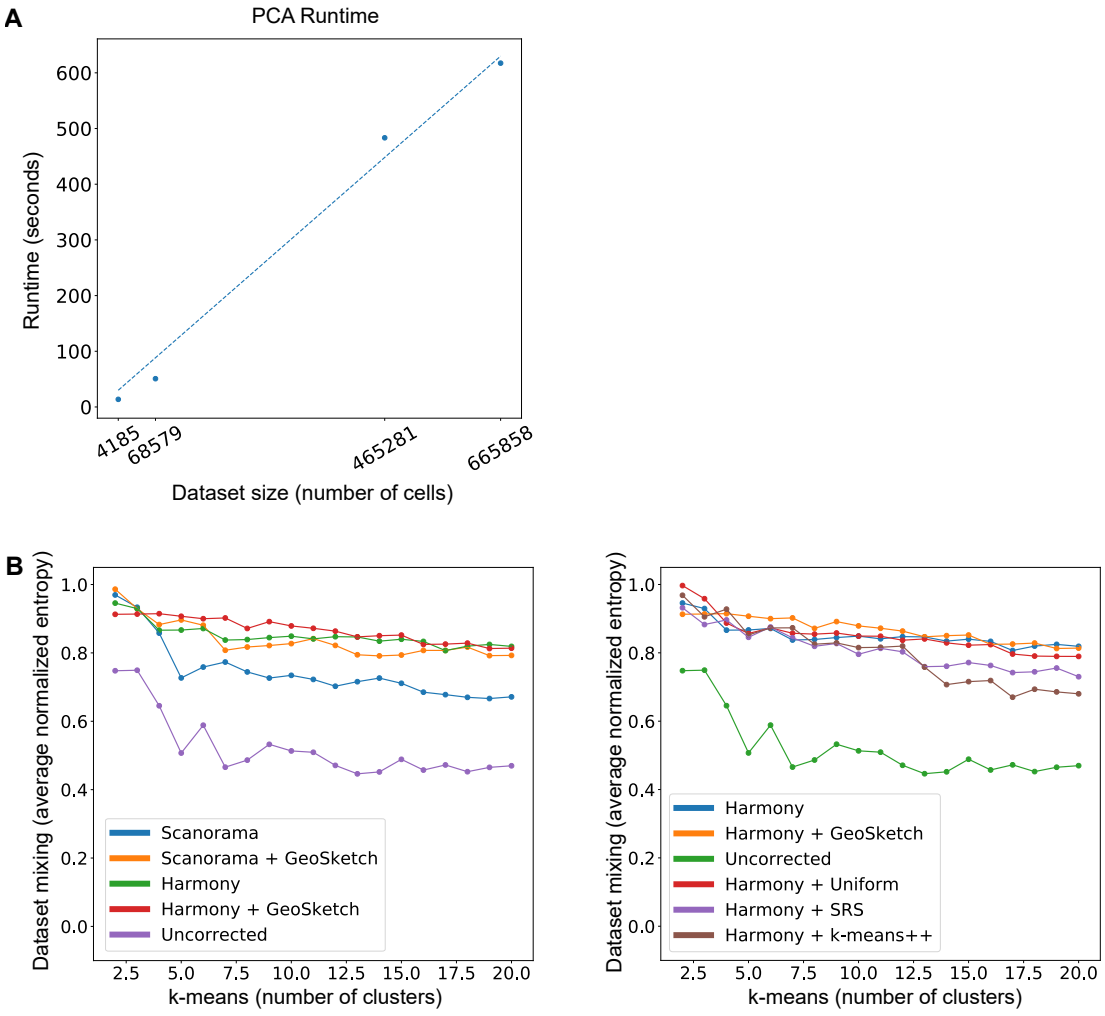

Figure A-16: Scalability and downstream acceleration.

(**A**) PCA runtime versus dataset size. The time required to learn a 100-dimensional representation of a scRNA-seq dataset using a randomized PCA scales linearly with the size of the dataset and has reasonable scalability to large-scale scRNA-seq experiments in the future. Each point given in the above plot corresponds to the time taken to compute a 100-dimensional embedding on each of the four main benchmark datasets used in the study. (**B**) Integration quality of methods with and without geometric sketching-based acceleration. Closer to 1 indicates more dataset mixing within clusters. Geometric sketching-based acceleration of integration methods yields integrations with comparable or better quality than applying the integration methods to the full dataset. Both geometric sketching and uniform sampling have comparable integration quality, but based on our other results, it is likely that geometric sketching would better align rare cell types in addition to common cell types. Using SRS and k-means++ sampling produces worse integration quality.

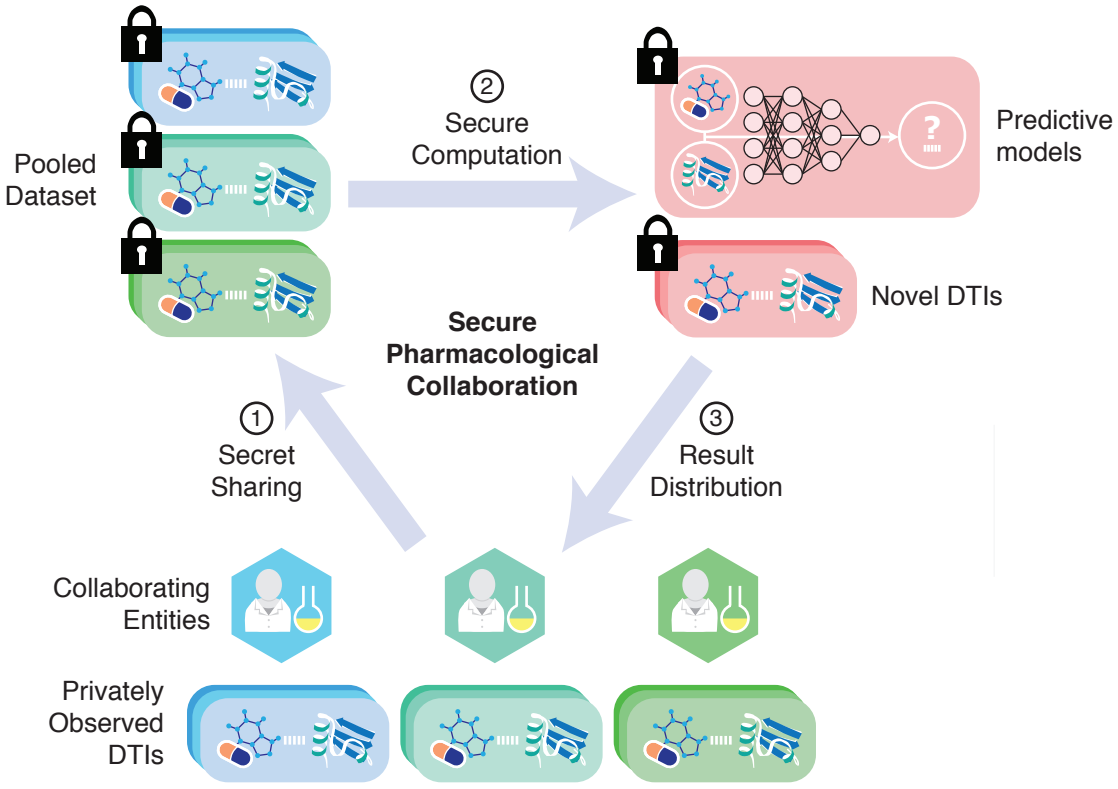

Figure A-17: Secure pipeline for pharmacological collaboration.

Collaborating entities (e.g., pharmaceutical companies or research laboratories) have large private datasets of DTIs, as well as corresponding chemical structures and protein sequences. In our protocol, the entities first use secret sharing to pool their data in a way that reveals no information about the underlying drugs, targets, or interactions (step 1). The collaborating entities then jointly execute a cryptographic protocol that trains a predictive model (e.g., a neural network) on the pooled dataset (step 2). The final model can be made available to participating entities or may be used to distribute DTI predictions to participants in a way that encourages greater data sharing (step 3).

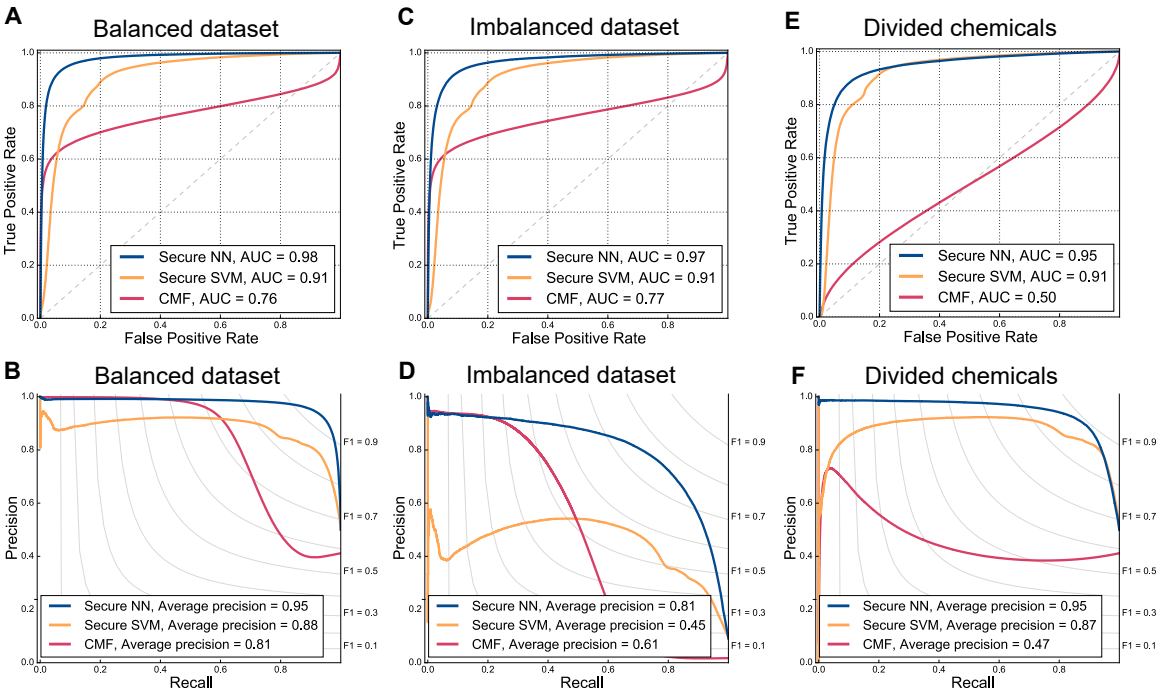

Figure A-18: ROC and precision-recall performance on STITCH.

Our secure neural network (Secure DTI) outperforms the secure SVM and plaintext CMF using the receiver-operator characteristic and precision-recall on a held-out test set consisting of (**A**, **B**) an equal number of positive interactions and negative pairs, (**C**, **D**) a 1:10 ratio of positive interactions to negative pairs, and (**E**, **F**) interactions involving chemicals not in the training or tuning datasets.

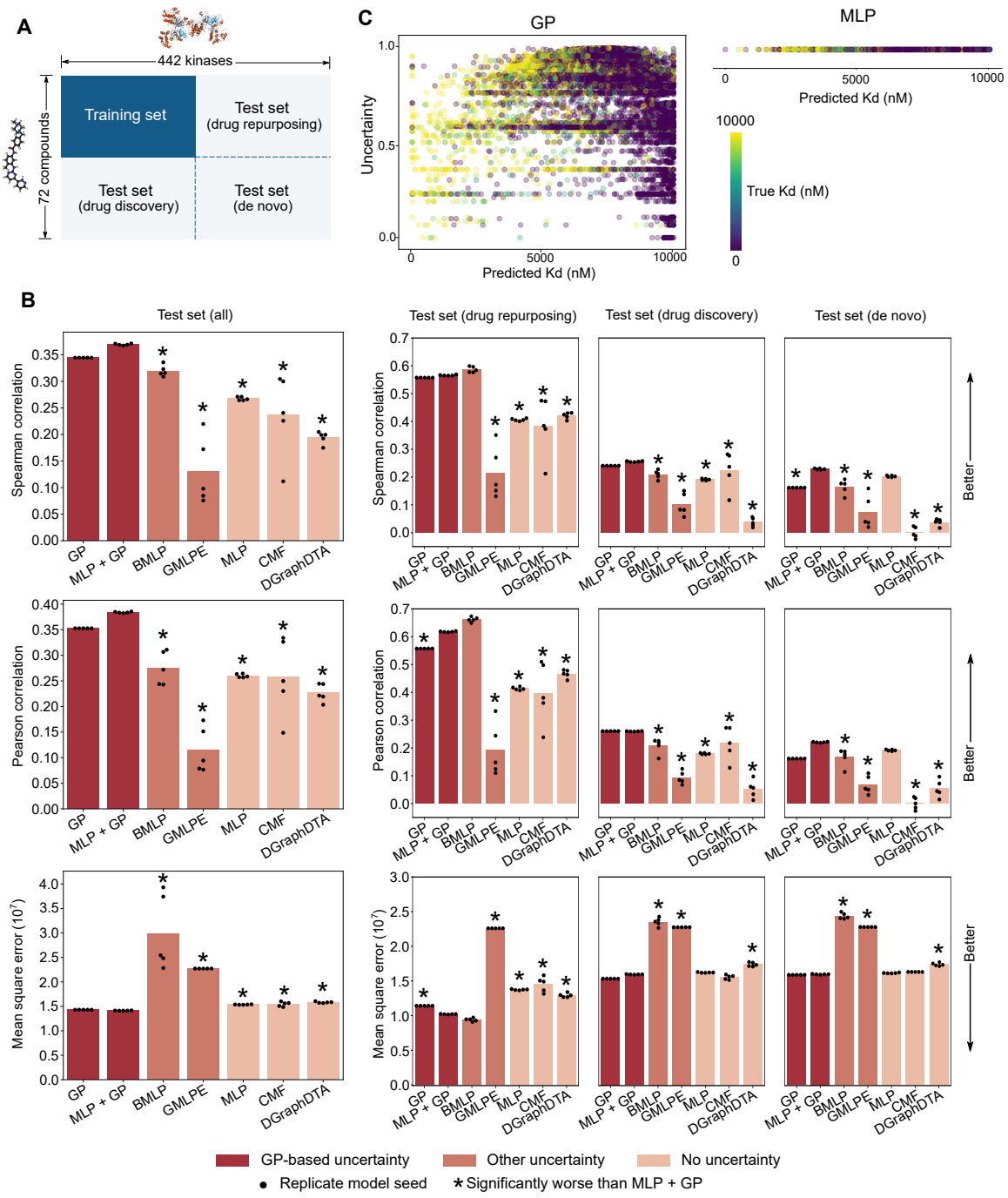

Figure A-19: Out-of-distribution cross validation experiments.

(**A**) Compound-kinase Kds among all pairs of 72 compounds and 442 kinases from Davis et al. was partitioned such that portions of the test set would have compounds not seen in the training data ("drug discovery"), kinases not seen in the training data ("drug repurposing"), and entire compound-kinase pairs not seen in the training data ("de novo"). (**B**) Performance within each out-of-distribution test set quadrant was measured with average-case metrics. Bar height indicates mean. (**C**) Scatter plots show the relationship between predicted Kd and model uncertainty, colored by the ground truth Kd, for all items in the test set. A normal MLP does not output uncertainty estimates.

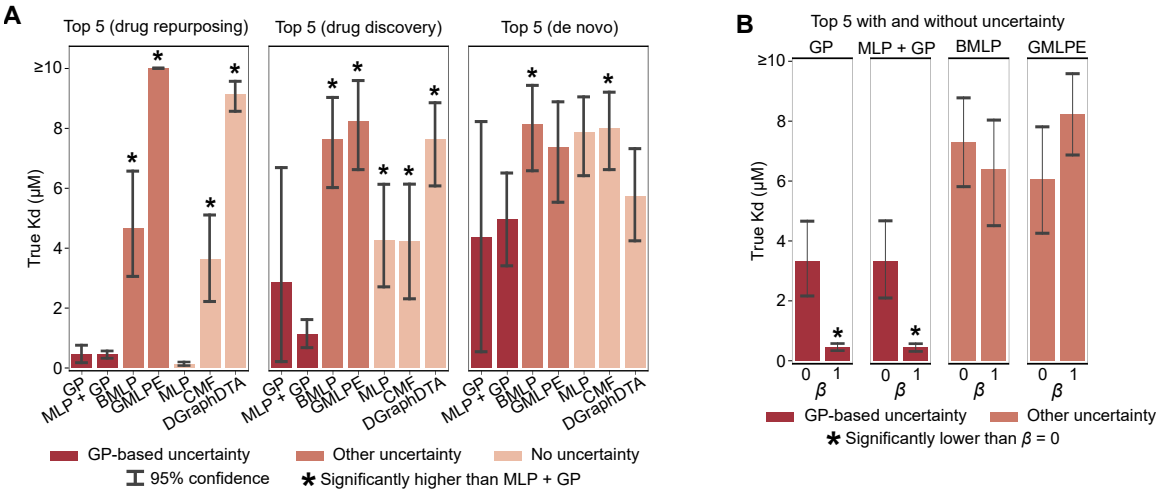

Figure A-20: Lead prioritization from cross validation experiments.

(**A**) Performance within each out-of-distribution test set quadrant was measured based on lead-prioritization (the true Kd of the top 5 acquired compounds in each random seed). Bar height indicates mean; statistical significance was assessed with a one-sided Welch's t-test P-value (FDR < 0.05). (**B**) The true Kd of the top 5 acquired compounds for each uncertainty model with ($\beta = 1$) and without uncertainty ($\beta = 0$). Bar height indicates mean; statistical significance was assessed with a one-sided Welch's t-test P-value (FDR < 0.05).

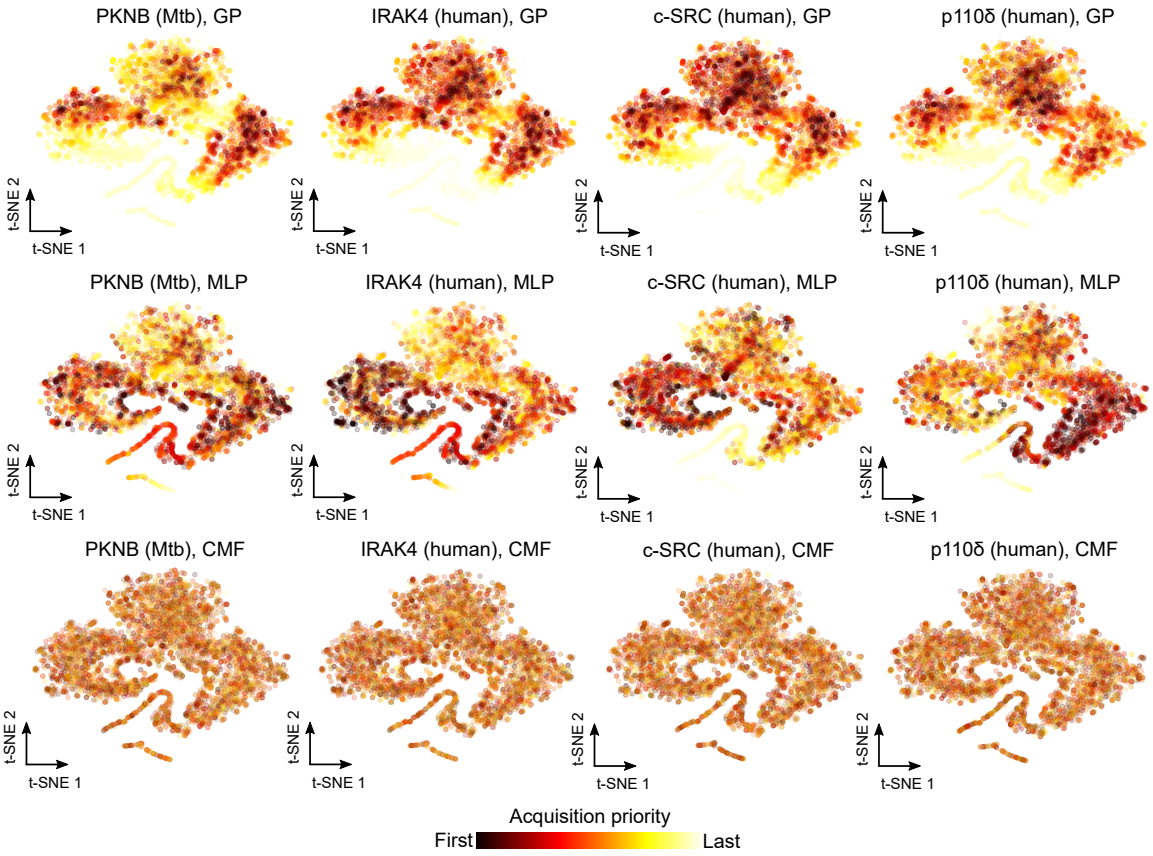

Figure A-21: Visualization of ZINC-Cayman acquisition priority.

Each compound in the ZINC-Cayman library is visualized as a two-dimensional t-SNE of the chemical embedding space, colored according to acquisition priority for high predicted binding affinity (and, if available, low uncertainty) to four kinases. GP-based acquisition prioritizes regions of the compound space close to available training data (Figure 7-3A,B). In contrast, MLP-based acquisition consistently prioritizes compounds that are out-of-distribution, indicating potentially pathological predictions. CMF predictions appear to lack any meaningful structure with regards to the compound landscape. PknB visualizations are the same as in Figure 7-3 and reproduced here for comparison.

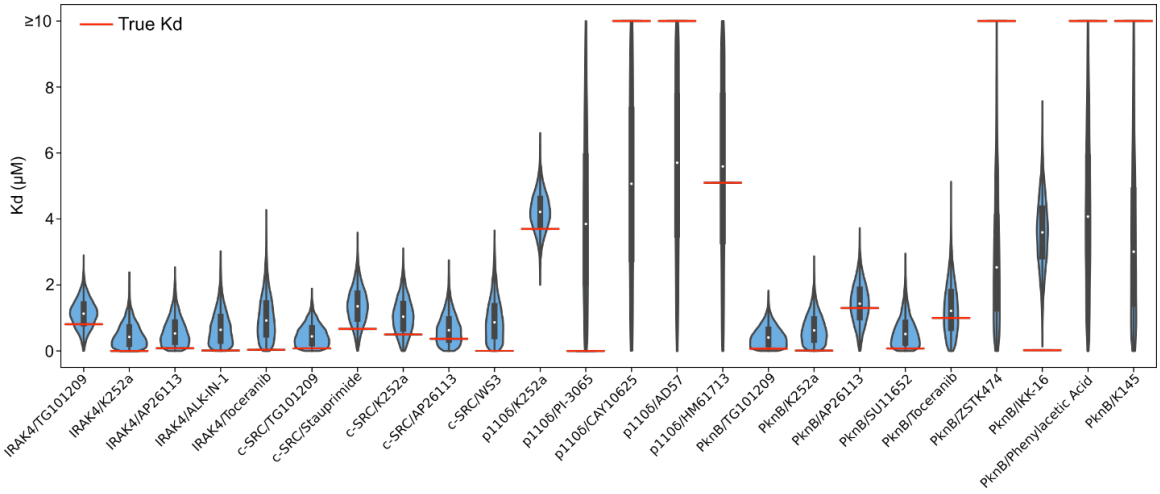

Figure A-22: Prediction uncertainty distributions and true values.

Violin plots and box plots correspond to the GP output for a given compound/kinase pair; the box extends from the first to third quartile, the whiskers extend from the min to max, and the white dot indicates the median. Horizontal red lines correspond to the true experimentally determined Kd. Note that uncertainty in addition to the prediction value adds interpretability; for example, the GP-outputted distributions corresponding to p110δ/K252a and PknB/phenylacetic acid have similar means but different variances, with greater tolerance for a false positive prediction in the latter.

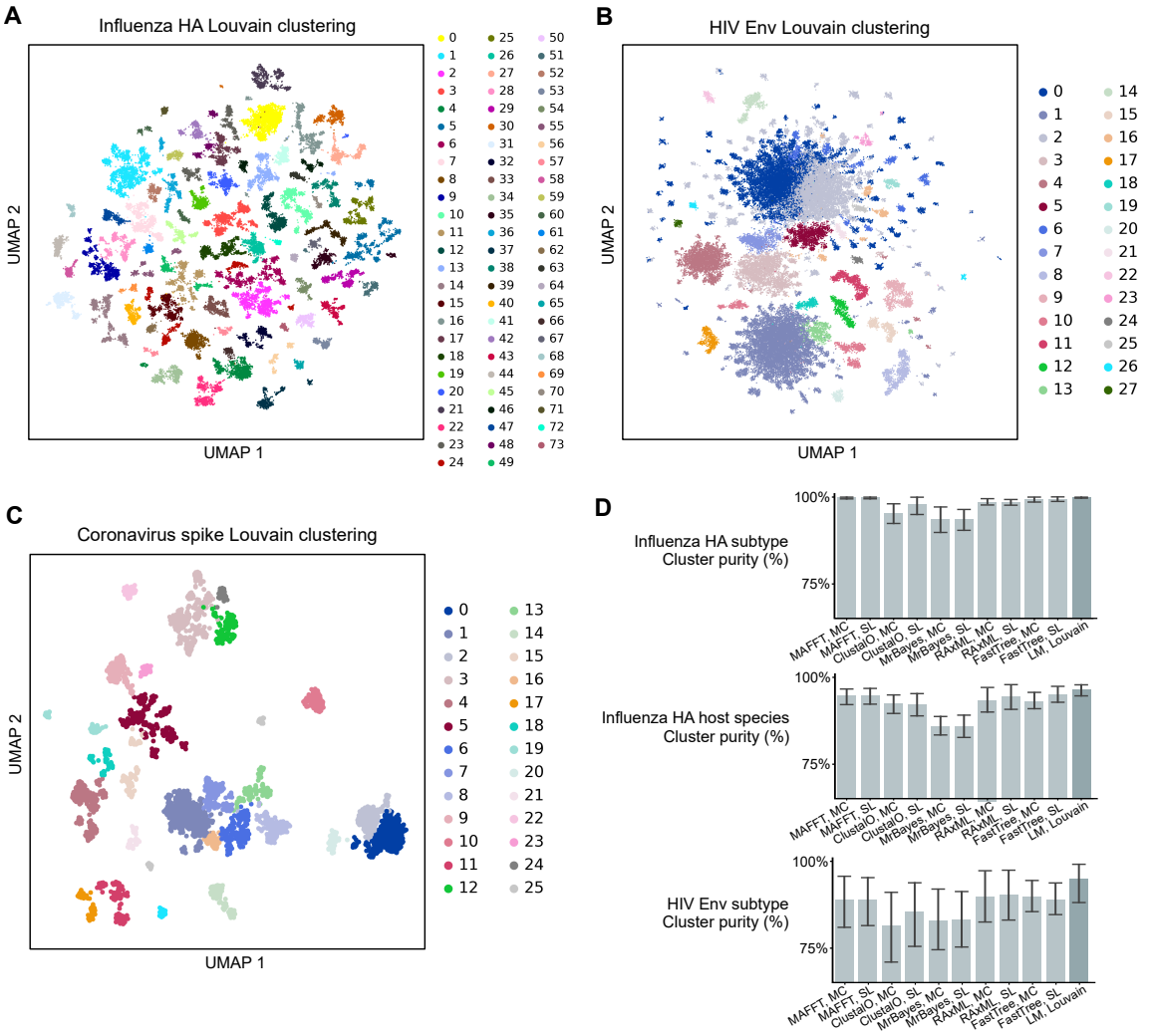

Figure A-23: Visualization of semantic landscape Louvain clustering.

(**A**-**C**) Louvain cluster labels, used to evaluate cluster purity of HA subtype, HA host species, and HIV subtype, are visualized with the same UMAP coordinates as in Figure 8-2. Part of HA cluster 30 was highlighted in Figure 8-2C. Coronavirus Louvain clusters 0 and 2 were highlighted in Figure 8-2G. (**D**) Cluster purities of Louvain clustering on language model (LM) semantic embeddings were compared to those of clustering with either the max clade (MC) or single linkage (SL) algorithms applied to phylogenetic trees constructed by MAFFT, Clustal Omega (ClustalO), MrBayes, RAxML, or FastTree. Results for Louvain compared to MAFFT with MC clustering are also shown in Figure 8-2. Bar height: mean; error bars: 95% confidence over $N = 74$ clusters for HA and $N = 28$ clusters for Env.

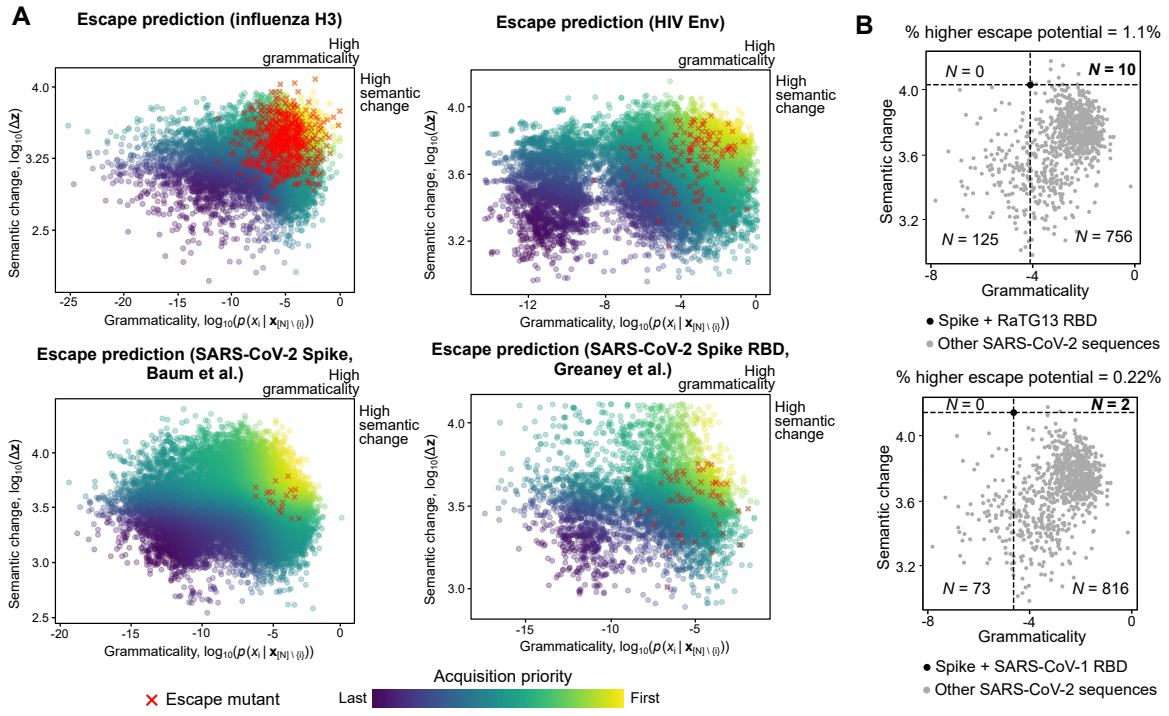

Figure A-24: Mutant semantic change and grammaticality.

(**A**) Each point in the scatter plot corresponds to a single-residue mutation of the indicated viral protein or protein domain. Points are colored by CSCS acquisition priority and a red X is additionally drawn over the points that correspond to escape mutations. (**B**) Across 891 unique, surveilled SARS-CoV-2 Spike sequences, ten (1.1%) have higher semantic change and grammaticality compared to a Spike sequence modified to have RaTG13 RBD-ACE2 contact residues and two (0.22%) have higher semantic change and grammaticality compared to a Spike sequence with a SARS-CoV-1 RBD-ACE2 contact residues.

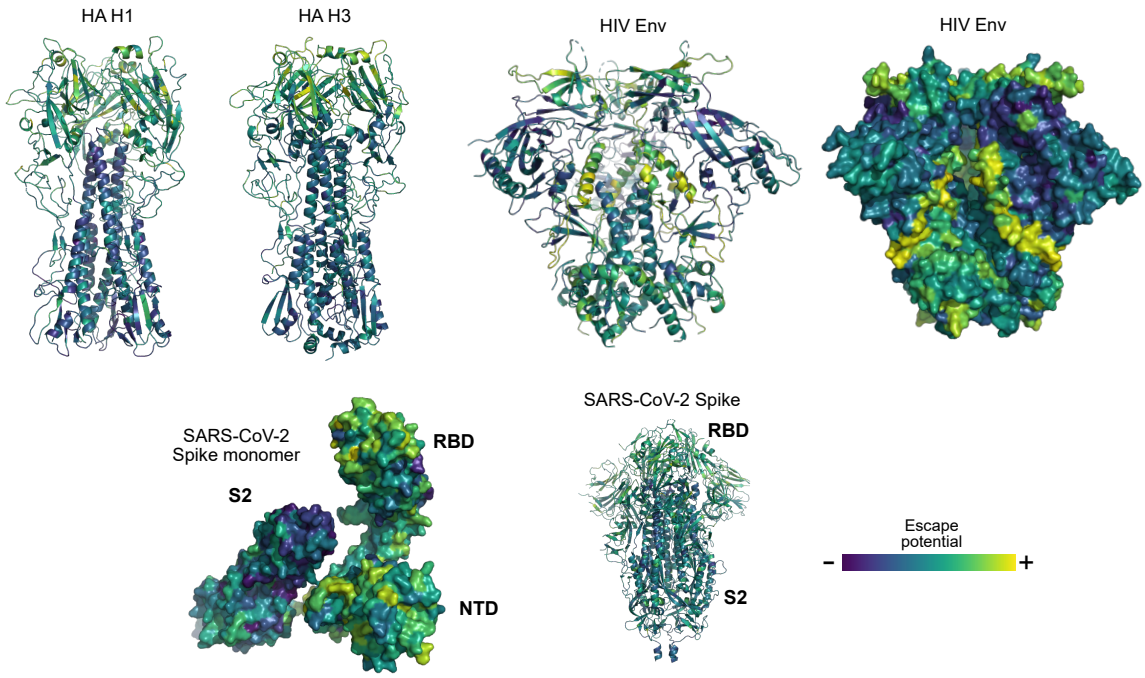

Figure A-25: Additional protein structure visualizations.

Cartoon illustration of HA H1 and HA H3; view of HIV Env as cartoon and surface oriented to illustrate the semantically important inner domain; and views of SARS-CoV-2 Spike in monomeric (surface) and trimeric form (cartoon) illustrating S2 escape depletion.

Dataset	# high-quality cells	Technology	Panorama
293T cells	2885	10X	1
Jurkat cells	3257	10X	1
Jurkat:293T 50:50 mixture	3388	10X	1
Jurkat:293T 99:1 mixture	4185	10X	1
Mouse neurons	9032	10X	2
Mtb infected macrophages	10827	SeqWell	3
Partially infected macrophages	212	SeqWell	3
Macrophages (donor 1)	4510	SeqWell	3
Macrophages (donor 2)	90	SeqWell	3
Mouse HSCs	2401	MARS-Seq	4
Mouse HSCs	774	Smart-seq2	4
Pancreatic islet cells	8569	inDrop	5
Pancreatic islet cells	2449	CEL-Seq 2	5
Pancreatic islet cells	1276	CEL-Seq	5
Pancreatic islet cells	638	Fluidigm C1	5
Pancreatic islet cells	2989	Smart-seq2	5
PBMCs	18018	10X	6
CD19+ B cells	2261	10X	6
CD14+ monocytes	295	10X	6
CD4+ helper T cells	3713	10X	6
CD56+ NK cells	6657	10X	6
CD8+ cytotoxic T cells	3990	10X	6
CD4+/CD45RO+ memory T cells	3628	10X	6
CD4+/CD25+ regulatory T cells	3365	10X	6
PBMCs	3774	Drop-seq	6
PBMCs	2293	10X	6

Table A.1: 26 datasets used in the panoramic integration experiments.

Cell Type	Number of cells	% of total	Differential Entropy
293T	28	0.669056	−461.66
Jurkat	4157	99.33094	−270.88

Table A.2: Statistics for 293/Jurkat mixture data.

Cell Type	Number of cells	% of total	Differential Entropy
$CD14^+$ Monocyte	3817	5.565844	−228.419
$CD19^+$ B	3306	4.820718	−213.47
$CD4^+/CD25^+$ T	2812	4.100381	−238.942
$CD4^+/CD45RA^+/CD25^-$ Naive T	3126	4.558247	−230.899
$CD4^+/CD45RO^+$ Memory T	5859	8.543432	−223.313
$CD4^+$ Helper T	11445	16.68878	−222.592
$CD56^+$ NK	14112	20.57773	−232.116
$CD8^+/CD45RA^+$ Naive Cytotoxic	21975	32.04334	−232.351
$CD8^+$ Cytotoxic T	1865	2.719491	−219.693
Dendritic	262	0.382041	−281.506

Table A.3: Statistics for PBMC data.

Cell Type	Number of cells	% of total	Differential Entropy
Astrocyte	54444	8.176518	−285.773
Endothelial Stalk	39298	5.901859	−271.857
Endothelial Tip	3818	0.573396	−277.978
Ependymal	2157	0.323943	−282.046
Macrophage	1695	0.254559	−290.916
Microglia	4614	0.692941	−275.472
Mural	12083	1.814651	−270.937
Neurogenesis	2372	0.356232	−257.468
Neuron	428051	64.28563	−232.534
Oligodendrocyte	104773	15.73504	−342.73
Other (unlabeled)	379	0.056919	−281.542
Polydendrocyte	12174	1.828318	−260.35

Table A.4: Statistics for adult mouse brain data.

Cell Type	Number of cells	% of total	Differential Entropy
Astrocyte	34915	7.504067	−293.16
Ependymal	2777	0.596844	−274.99
Immune/Blood	14081	3.026343	−289.20
Neuron	147059	31.60649	−243.09
Oligodendrocyte	219220	47.11561	−338.52
Peripheral Glia	16066	3.452967	−328.23
Vascular	31163	6.697673	−265.75

Table A.5: Statistics for developing mouse brain data.

Cell Type	Uniform	k-means++	SRS	Geometric sketching
Astrocyte	1088	1090	389	1277
Endothelial Stalk	761	782	533	556
Endothelial Tip	84	107	175	815
Ependymal	33	68	102	165
Macrophage	43	31	47	262
Microglia	86	75	99	397
Mural	247	297	346	519
Neurogenesis	47	89	171	151
Neuron	8655	9746	10821	7975
Oligodendrocyte	2031	747	53	627
Other (unlabeled)	5	10	24	49
Polydendrocyte	237	275	557	524

Table A.6: Statistics for adult mouse brain sketch.

Cell Type	Uniform	k-means++	SRS	Geometric sketching
Astrocyte	697	949	905	1194
Ependymal	49	115	339	614
Immune/Blood	265	418	350	466
Neuron	2982	4823	3911	4533
Oligodendrocyte	4371	1649	2273	1098
Peripheral Glia	332	322	259	230
Vascular	609	1029	1268	1170

Table A.7: Statistics for developing mouse brain sketch.

Property	# Samples	Min	Median	Max	Mean	S. Dev.
Exact molecular weight (Da)	10,833	61.0	352.2	994.5	367.9	140.3
SSSR	10,833	0	2	12	2.4	1.7
Balaban J	10,833	0.7	2.0	6.3	2.2	0.8
Bertz CT	10,833	17.2	661.3	2850.1	734.0	399.1
Tanimoto similarity (RDK Fingerprint, 2048 bits)	58,671,528	0.00	0.18	1.00	0.20	0.11
Tanimoto similarity (Morgan Fingerprint, 2048 bits, radius = 2)	58,671,528	0.00	0.09	1.00	0.11	0.07

Table A.8: Statistics for ZINC-Cayman dataset.

Compound	PknB	IRAK4	c-SRC	p110δ
(1′S,2′S)-Nicotine-1′-oxide	>10000			
3-O-methyl-N-acetyl-D-Glucosamine			>10000	
8-iso Prostaglandin E2			>10000	
AB-BICA	>10000			
Abacavir		>10000		
AD57				>10000
ALK-IN-1	620	13		
AP26113	1300	83	370	
Atracurium	>10000			
AVL-292	5500			
CAY10625				>10000
Epinastine	>10000			
Evodiamine			>10000	
GLYX 13			>10000	
GP-NEPEA				>10000
HM61713				5100
IKK-16	22			
K145	>10000			
K252a	11	0.85	500	3700
Lovastatin	>10000			
LY2886721	>10000			
Mevastatin	>10000			
NVP-TAE226	9900			
Oxymatrine	>10000			
Phenylacetic Acid	>10000			
PI-3065				0.36
PX 1	>10000			
Ro 4929097	>10000			
Ro 67-7476			>10000	
S-(5′-Adenosyl)-L-methionine chloride	>10000			
Stauprimide			670	
SU11652	76			
TG101209	71	810	79	
Toceranib	1000	37		
WAY-161503		>10000		
WS3			4	
ZSTK474	>10000			

Table A.9: Summary of tested interactions
Kd values are in nM. Blank cells indicate an interaction that was not tested.

Compound	MIC (μM)
IKK-16	25
K252a	25
Rifampicin	1.25
SU11652	25
TG101209	>50

Table A.10: MIC values for axenic Mtb.

Compound	Target (Kd < 100 nM)	Closest compound	Tanimoto sim.
IKK-16	PknB	Imatinib	0.31
PI-3065	p110δ	GDC-0941	0.46
ALK-IN-1	IRAK4	TAE-684	0.55
TG101209	PknB, c-SRC	TG-101348	0.68
Toceranib	PknB, c-SRC	Sunitinib	0.72
AP26113	IRAK4	TAE-684	0.73
WS3	c-SRC	AST-487	0.73
K252a	PknB, IRAK4	CEP-701	0.77
SU11652	PknB	Sunitinib	0.81

Table A.11: Closest training set compounds.

Model	WSN33	Bei89	Bk79	Bris07L194	HK68	Mos99	NDako16	BF520	BG505	Spike
Semantic change (Spearman r)	-0.1175	-0.1653	-0.4040	-0.0051	-0.2549	-0.2097	-0.0711	-0.1024	-0.1101	-0.4421
Grammaticality (Spearman r)	0.2789	0.1876	0.4274	0.4021	0.5396	0.2854	0.3508	0.2063	0.2684	0.4852
Semantic change (P-value)	2.94E-34	6.69E-05	5.02E-24	0.9034	5.38E-10	3.80E-07	0.08842	1.20E-30	1.16E-35	<1E-308
Grammaticality (P-value)	1.08E-190	5.81E-06	5.55E-27	8.47E-24	7.97E-45	2.96E-12	4.02E-18	6.54E-121	4.85E-209	<1E-308

Table A.12: Fitness correlation and P-values.

Model	HA H1	HA H3	Env BG505	Spike	Spike RBD
MAFFT	0.697*	0.598*	0.523	0.618	0.526
EVcouplings (ind.)	0.706*	0.691*	0.536	0.689	0.527
EVcouplings (epi.)	0.726*	0.687*	0.552	0.713	0.610*
Grammaticality (our model)	0.820*	0.684*	0.667*	0.820*	0.704*
Bepler	0.660*	0.644*	0.561	0.534	0.664*
TAPE transformer	0.584*	0.526	0.574*	0.667	0.556
UniRep	0.482	0.452	0.534	0.745*	0.606*
Semantic change (our model)	0.664*	0.709*	0.622*	0.660	0.584*
CSCS (our model)	**0.834***	**0.771***	**0.692***	**0.854***	**0.709***

Table A.13: Escape prediction normalized AUC values.

An asterisk (*) indicates a significant AUC based on a Bonferroni-corrected one-sided permutation-based P-value of less than 0.05.

Appendix B

Deferred details from Chapter 5

B.1 Details of the quadratic program

We introduce some notation to condense the expressions. Define $\mathbf{w} \in \mathbb{R}^k$ where $w_i \triangleq \omega_i^2$, $\boldsymbol{\delta}_{ij} \in \mathbb{R}^k$ with $[\boldsymbol{\delta}_{ij}]_s \triangleq ([\mathbf{x}_i]_s - [\mathbf{x}_j]_s)^2$ (i.e., squared elements of $\mathbf{x}_i - \mathbf{x}_j$) and, for convenience, let $\mathcal{P}$ be the set of pairs of observations $\mathcal{P} = \{\{i, j\} : 1 \leq i \leq j \leq N\}$. Using the fact that the covariance between random variables x and y is given by $\mathrm{Cov}(x, y) \triangleq \mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y]$ and the variance as $\mathrm{Var}(x) \triangleq \mathbb{E}[x^2] - \mathbb{E}^2[x]$, we have that

$$\begin{aligned}
\mathrm{Cov}(\mathbf{w}, \rho_\ell) &= \frac{1}{|\mathcal{P}|} \sum_{\{i,j\}\in\mathcal{P}} \rho_\ell(\mathbf{x}_i^{(\ell)}, \mathbf{x}_j^{(\ell)}) \boldsymbol{\delta}_{ij}^{\mathrm{T}} \mathbf{w} - \frac{1}{|\mathcal{P}|^2} \left(\sum_{\{i,j\}\in\mathcal{P}} \boldsymbol{\delta}_{ij}^{\mathrm{T}} \mathbf{w} \right) \left(\sum_{\{i,j\}\in\mathcal{P}} \rho_\ell(\mathbf{x}_i^{(\ell)}, \mathbf{x}_j^{(\ell)}) \right) \\
&= \left(\frac{1}{|\mathcal{P}|} \mathbf{a}_\ell - \frac{1}{|\mathcal{P}|^2} \mathbf{b}_\ell \right)^{\mathrm{T}} \mathbf{w} \qquad \text{and} \\
\mathrm{Var}(\mathbf{w}) &= \frac{1}{|\mathcal{P}|} \sum_{\{i,j\}\in\mathcal{P}} \mathbf{w}^{\mathrm{T}} \boldsymbol{\delta}_{ij} \boldsymbol{\delta}_{ij}^{\mathrm{T}} \mathbf{w} - \frac{1}{|\mathcal{P}|^2} \left(\sum_{\{i,j\}\in\mathcal{P}} \boldsymbol{\delta}_{ij}^{\mathrm{T}} \mathbf{w} \right)^2 \\
&= \mathbf{w}^{\mathrm{T}} \left(\frac{1}{|\mathcal{P}|} \mathbf{S} - \frac{1}{|\mathcal{P}|^2} \mathbf{T} \right) \mathbf{w}
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{a}_\ell &\triangleq \sum_{\{i,j\}\in\mathcal{P}} \rho_\ell(\mathbf{x}_i^{(\ell)}, \mathbf{x}_j^{(\ell)})\boldsymbol{\delta}_{ij}, \\
\mathbf{b}_\ell &\triangleq \left(\sum_{\{i,j\}\in\mathcal{P}} \rho_\ell(\mathbf{x}_i^{(\ell)}, \mathbf{x}_j^{(\ell)})\right) \sum_{\{i,j\}\in\mathcal{P}} \boldsymbol{\delta}_{ij}, \\
\mathbf{S} &\triangleq \sum_{\{i,j\}\in\mathcal{P}} \boldsymbol{\delta}_{ij}\boldsymbol{\delta}_{ij}^{\mathrm{T}}, \qquad \text{and} \\
\mathbf{T} &\triangleq \left(\sum_{\{i,j\}\in\mathcal{P}} \boldsymbol{\delta}_{ij}\right)\left(\sum_{\{i,j\}\in\mathcal{P}} \boldsymbol{\delta}_{ij}^{\mathrm{T}}\right).
\end{aligned}$$

Note that $\mathbf{a}_\ell$ and $\mathbf{b}_\ell$ depend only on dataset $\mathbf{X}_\ell$ while $\mathbf{S}$ and $\mathbf{T}$ depend only on the primary dataset $\mathbf{X}_1$.

Recall the general optimization problem in Equation (5.5) and the framework for quadratic programming in Equation (5.3). Using the notation defined there, the mapping is now straightforward, i.e.,

$$\begin{aligned}
\mathbf{v} &\triangleq \mathbf{w}, \\
\mathbf{Q} &\triangleq \frac{1}{|\mathcal{P}|}\mathbf{S} - \frac{1}{|\mathcal{P}|^2}\mathbf{T} + \lambda\mathbf{I}_k, \qquad \text{and} \\
\mathbf{q} &\triangleq -2\lambda\mathbf{1} - \sum_{j=1}^{r} \gamma_\ell \left(\frac{1}{|\mathcal{P}|}\mathbf{a}_\ell + \frac{1}{|\mathcal{P}|^2}\mathbf{b}_\ell\right).
\end{aligned}$$

To see that $\mathbf{Q}$ is psd, note that

$$\mathbf{Q} = \lambda\mathbf{I}_k + \frac{1}{|\mathcal{P}|}\sum_{\{i,j\}\in\mathcal{P}} (\boldsymbol{\delta}_{ij} - \boldsymbol{\mu})(\boldsymbol{\delta}_{ij} - \boldsymbol{\mu})^{\mathrm{T}}$$

where $\boldsymbol{\mu} = \frac{1}{|\mathcal{P}|}\sum_{\{i,j\}\in\mathcal{P}} \boldsymbol{\delta}_{ij}$, so $\mathbf{Q}$ is a sum of psd matrices.

For the linear constraint, we express $\mathbf{G}$ in block form

$$\mathbf{G} = \begin{bmatrix} \mathbf{H} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_k \end{bmatrix}$$

where each row in $\mathbf{H}$ is given by

$$\mathbf{H}_j \triangleq -\frac{1}{|\mathcal{P}|}\mathbf{a}_j - \frac{1}{|\mathcal{P}|^2}\mathbf{b}_j \text{ for } 1 \leq j \leq r.$$

For the right side of the inequality constraint, we have a $(r+k)$-dimensional vector $\mathbf{h}$ where each element is

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

where if $\beta_j = 0$ for some j (i.e., no correlation constraint) the corresponding rows can be deleted from $\mathbf{H}$ and $\mathbf{h}$. We have no equality constraints in our optimization, so $\mathbf{A}$ and $\mathbf{b}$ from Equation (5.3) are not needed.

B.2 Correlation and neighborhood structure

We first introduce some notation. $\mathbf{X}_1$ and $\mathbf{X}_2$ are the two datasets under consideration, each covering the same set of points $\mathcal{S}$, with $|\mathcal{S}| = N$. Each dataset has associated metrics ρ_1 and ρ_2, respectively. We denote the Spearman rank correlation between the $\binom{N}{2}$ pairwise distances within each of $\mathbf{X}_1$ and $\mathbf{X}_2$ as $\text{pwcs}(\rho_1, \rho_2)$. Also, we will often refer to point-triplets $(A, B, C) \in \mathcal{S}$. We recapitulate that the Spearman correlation is analogous to the more popular Pearson correlation with the distinction that the former works with ranks while the latter works with actual values.

Below, we assume WLOG that the $\binom{N}{2}$ pairwise distances in a dataset are distinct. If they are not, it is easy to break the ties in a way that preserves the ordering of all non-distinct distances. For example, let δ be the smallest *non-zero* difference between two pairwise distances; for each point in $\mathcal{S}$, we pick a random direction and move it $\frac{\delta}{100}$ units along that direction.

Lemma 3. $\text{pwcs}(\rho_1, \rho_2) = 1 \iff$ *the neighborhood structures of* $\mathbf{X}_1$ *and* $\mathbf{X}_2$ *are identical.*

Proof. If $\text{pwcs}(\rho_1, \rho_2) = 1$, then the pairwise distances in $\mathbf{X}_1$ and $\mathbf{X}_2$ are ranked identically; this follows from the definition of Spearman correlation. In other words, $\rho_1(A, B) < \rho_1(A, C) \iff \rho_2(A, B) < \rho_2(A, C)$. This, in turn, implies that, for each A, its distance to the remaining $N - 1$ points is in the same order in both datasets. Thus, the neighborhood-structures are identical.

The other direction also follows directly from our definition of identical neighborhood structures. If $\mathbf{X}_1$ and $\mathbf{X}_2$ have identical neighborhood structures, then for any point $A \in \mathcal{S}$ and any k, the set of k-nearest neighbors of A are the same in $\mathbf{X}_1$ and $\mathbf{X}_2$. This is equivalent to stating that the j-th nearest point to A is the same in $\mathbf{X}_1$ and $\mathbf{X}_2$, for all j. Repeating this for all points establishes that the pairwise distances are ranked identically in $\mathbf{X}_1$ and $\mathbf{X}_2$. □

We now describe a more general connection between $\text{pwcs}(\rho_1, \rho_2)$ and neighborhood similarity in the case when there *are* some mis-alignments between $\mathbf{X}_1$ and $\mathbf{X}_2$.

Lemma 4. $\text{pwcs}(\rho_1, \rho_2) > 1-\epsilon \implies \text{mismatch}(\mathbf{X}_1, \mathbf{X}_2) < \sqrt{2\epsilon}$, *where* $\text{mismatch}(\mathbf{X}_1, \mathbf{X}_2)$ *is the fraction of point-triplets that are misaligned, i.e.,* $\rho_1(A, B) < \rho_1(A, C)$ *but* $\rho_2(A, B) > \rho_2(A, C)$.

Proof. We first convert pairwise distances in $\mathbf{X}_1$ and $\mathbf{X}_2$ to ranks, i.e., $1, \ldots, \frac{N(N-1)}{2}$. The rank-ordering $\mathcal{R}_2$ for $\mathbf{X}_2$ is a permutation of the rank-ordering $\mathcal{R}_1$ for $\mathbf{X}_1$. The lemma above is essentially relating two measures of distance between these permutations: the Spearman distance and the Kendall Tau distance. The latter counts the number of inversions, i.e., the number of pairs $p_{AB}, p_{AC} \in \mathcal{R}_1$ whose order is inverted in $\mathcal{R}_2$. We appeal to a well-known result from Durbin and Watson [DW51] that states

$$r_S \leq 1 - \frac{1 - r_K}{2(M+1)}[(M-1)(1-r_K) + 4]$$

where r_K is the Kendall Tau correlation and the r_S is the Spearman correlation between the permutations, and M is their length (here, $M \triangleq \frac{N(N-1)}{2}$). For large N, this simplifies to

$$\frac{(1 - r_K)^2}{2} \geq 1 - r_S$$

The inequality stated in the lemma directly follows from this. □

The results above provide support to our intuition that Spearman rank correlation of pairwise distances is a direct way of measuring distortions of neighborhoods. However, while rank correlation is hard to compute especially in a QP framework and the Pearson correlation can be quite different than the rank correlation in general settings, on low-metric-entropy biological datasets we find that empirically it captures the right intuition.

Appendix C

Deferred details from Chapter 6

C.1 Secure-DTI

C.1.1 Secure computation preliminaries

We consider the simplest setting with two collaborating entities (academic labs or pharmaceutical companies), denoted CP1 and CP2 ("computing parties"). Our protocol also involves a third auxiliary entity CP0 that is involved only during an offline pre-computation phase. As we require CP0 to not collude with other entities in the protocol for our security guarantee to hold, we expect a trusted party (e.g., NIH) to play the role of CP0. Using a cryptographic technique called secret sharing, each of CP1 and CP2 shares its DTI data (i.e., drug- and target-specific input features and observed interaction scores) with the other participant in such a way that enables privacy-preserving computation over the pooled data. During this computation, CP1 and CP2 leverage pre-computed data from CP0 (which is input-agnostic) to greatly speed up the computation using a technique called generalized Beaver partitioning (12). Finally, CP1 and CP2 combine their outputs to reconstruct the final results (e.g., neural network weights or predicted DTIs).

We adopt the "honest-but-curious" security model in which the protocol participants are assumed to follow the protocol exactly as specified, but at the end of

the protocol execution, a party may try to infer additional information about other parties' private inputs based on their view of the protocol. Under this setting, our protocol is secure as long as CP0 and at least one of the other CPs remains honest. We also assume all communication occurs over a secure and authenticated channel (e.g., over the TLS protocol). This work generally follows the paradigm of computing on secret-shared data first formalized by Ben-Or et al. [BOGW88].

For our notation we adopt the notation and definitions given in the supplementary materials for Cho et al. [CWB18], which we refer to throughout this supplement although we restate essential descriptions as necessary. Our protocol relies on a two-party additive secret sharing scheme where a value $x \in \mathbb{Z}_q$ is shared between CP1 and CP2, where we denote a secret sharing of x between CP1 and CP2 as $[x] \triangleq \langle [x]_1, [x]_2 \rangle$, where the notation $\langle [x]_1, [x]_2 \rangle$ means that $[x]_1$ and $[x]_2$ are shares of x in $\mathbb{Z}_q$ individually owned by CP1 and CP2, respectively, such that $x = [x]_1 + [x]_2$. Adding two secret shared values $[x]$ and $[y]$ can be done by having each party add their own shares, i.e., $\langle [x]_1 + [y]_1, [x]_2 + [y]_2 \rangle$. Adding by a public field element $a \in \mathbb{Z}_q$ can be written as $\langle [x]_1 + a, [x]_2 \rangle$. Multiplying by a public field element is also simple and can be written as $\langle a[x]_1, a[x]_2 \rangle$. Multiplying two secret shared values is more involved but, as described in Cho et al., it is possible to obtain secure protocols that generalize a tool known as Beaver multiplication triples [Bea91] for efficiently computing many multiplications, including matrix multiplications.

We use a fixed-point representation of signed real numbers that uses k total bits, of which f is the number of bits allocated to the fractional domain, referred to as the "precision." We denote a secret shared fixed-point encoding of $x \in \mathbb{R}$ as $[x]^{(f)}$. Multiplication of two fixed point numbers outputs a result with precision of $2f$ instead of f, so we use the truncation routine from Catrina and Saxena [CS10] to rescale the precision, which we denote

$$[x_{\text{trunc}}] \leftarrow \textbf{Truncate}([x], b, s)$$

as defined in Cho et al., where b is the number of bits to mask, which is chosen such

that a sufficient level of statistical security is guaranteed, and s is the number of least significant bits to truncate. We also use the data oblivious sign test (i.e., a comparison with zero) as proposed by Nishide and Ohta [NO07], which as defined in Cho et al., takes the form

$$[\mathbb{1}\{x > 0\}] \leftarrow \textbf{IsPositive}([x]^{(f)})$$

where $[\mathbb{1}\{x > 0\}]$ is a secret shared integer value equal to 1 if x is positive, and 0 otherwise. This comparison protocol requires $O(1)$ rounds of communication and $O(k)$ invocations of multiplication protocols, where k is the bit length.

C.1.2 Secure neural network computation

We are now ready to define our secure protocols for neural network training, described in the plaintext setting in Section 6.3.2. First, we implement the **GradientDescent** protocol (Algorithm 2), which takes as input $[\mathbf{X}_{\text{batch}}]^{(f)}$, $[\mathbf{y}_{\text{batch}}]^{(f)}$,

$$[\mathbf{W}]^{(f)} \triangleq [\mathbf{W}^{(1)}]^{(f)}, \ldots, [\mathbf{W}^{(L+1)}]^{(f)}, [\mathbf{W}_v^{(1)}]^{(f)}, \ldots, [\mathbf{W}_v^{(L+1)}]^{(f)}, \quad \text{and} \tag{C.1}$$

$$[\mathbf{b}]^{(f)} \triangleq [\mathbf{b}^{(1)}]^{(f)}, \ldots, [\mathbf{b}^{(L+1)}]^{(f)}, [\mathbf{b}_v^{(1)}]^{(f)}, \ldots, [\mathbf{b}_v^{(L+1)}]^{(f)}, \tag{C.2}$$

and then outputs the updated model parameters $[\mathbf{W}]^{(f)}$ and $[\mathbf{b}]^{(f)}$ after performing a single stochastic gradient update.

Multiple rounds of **GradientDescent** are invoked in which a new $\mathbf{X}_{\text{batch}}$ and corresponding $\mathbf{y}_{\text{batch}}$ are randomly sampled from the full dataset, which we repeat until reaching the max number of iterations T. Finally, after the training procedure is finished, the two collaborating entities CP1 and CP2 combine their shares to jointly reconstruct $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L+1)}$ and $\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(L+1)}$ in plaintext. If the entities wish to keep the trained model private, an alternative is to only reveal new predictions obtained by securely evaluating the model on a test set.

Notably, our model is kept private during the entire gradient descent training procedure and can remain private when making novel predictions. This is in contrast to existing approaches for privacy-preserving neural networks that are developed for

the simpler setting where the model is trained in plaintext, and only the evaluation of the trained model on new data instances is performed in a privacy-preserving manner. Such approaches are not able to make use of the higher-quality models that can be obtained by training on a much larger set of data securely pooled from multiple entities, as enabled by our framework.

The model that we report in the main text for the DrugBank and STITCH dataset experiments has hyperparameters $M_{\text{batch}} = 50$, $L = 2$, $H = 250$, $T = 20,000$, $\lambda = 0.001$, $\mu = 0.9$, and $\alpha = 0.01$, chosen based on a small-scale grid search. We

Algorithm 2: GradientDescent

Input: Mini-batch features $[\mathbf{X}_{\text{batch}}]^{(f)}$, label vector $[\mathbf{y}_{\text{batch}}]^{(f)}$, weight parameters $[\mathbf{W}]^{(f)}$, bias parameters $[\mathbf{W}]^{(f)}$, velocity parameters $[\mathbf{W}_v]^{(f)}$ and $[\mathbf{b}_v]^{(f)}$

Output: Updated parameters $[\mathbf{W}]^{(f)}, [\mathbf{b}]^{(f)}, [\mathbf{W}_v]^{(f)}, [\mathbf{b}_v]^{(f)}$

```
/* Forward propagation.  */
```

for $l = 1, \ldots, L$ **do**
  if $l = 1$ **then**
    $\left[\mathbf{Z}^{(1)}\right]^{(f)} \leftarrow \textbf{Truncate}\left([\mathbf{W}^{(1)}]^{(f)}[\mathbf{X}_{\text{batch}}]^{(f)}, k+f, f\right)$
  else
    $\left[\mathbf{Z}^{(l)}\right]^{(f)} \leftarrow \textbf{Truncate}\left([\mathbf{W}^{(l)}]^{(f)}[\mathbf{Z}^{(l-1)}]^{(f)}, k+f, f\right)$
  end
  for $i = 1, \ldots, M$ **do**
    $\left[\mathbf{Z}^{(l)}_{:,i}\right]^{(f)} \leftarrow [\mathbf{Z}^{(l)}_{:,i}]^{(f)} + [\mathbf{b}^{(l)}]^{(f)}$
  end
  $\left[\mathbb{1}\{\mathbf{Z}^{(l)} > 0\}\right] \leftarrow \textbf{IsPositive}\left([\mathbf{Z}^{(l)}]^{(f)}\right)$
  $\left[\mathbf{Z}^{(l)}\right]^{(f)} \leftarrow [\mathbf{Z}^{(l)}]^{(f)} \odot [\mathbb{1}\{\mathbf{Z}^{(l)} > 0\}]$
end

if $L = 0$ **then**
  $[\mathbf{s}]^{(f)} \leftarrow \textbf{Truncate}\left([\mathbf{W}^{(1)}]^{(f)}[\mathbf{X}_{\text{batch}}]^{(f)}, k+f, f\right)$
else
  $[\mathbf{s}]^{(f)} \leftarrow \textbf{Truncate}\left([\mathbf{W}^{(L+1)}]^{(f)}[\mathbf{Z}^{(L)}]^{(f)}, k+f, f\right)$
end
for $i = 1, \ldots, M$ **do**
  $[\mathbf{s}_i]^{(f)} \leftarrow [\mathbf{s}_i]^{(f)} + [\mathbf{b}^{(L+1)}]^{(f)}$
end

/* Loss function evaluation. */
$[-\mathbf{s} \odot \mathbf{y}_{\text{batch}}]^{(f)} \leftarrow \textbf{Truncate}\left(-[\mathbf{s}]^{(f)} \odot [\mathbf{y}_{\text{batch}}]^{(f)}, k+f, f\right)$
$[\mathbf{1} - \mathbf{s} \odot \mathbf{y}_{\text{batch}}]^{(f)} \leftarrow$ CP2: $[-\mathbf{s} \odot \mathbf{y}_{\text{batch}}]^{(f)} + \mathbf{1}$
$[\mathbb{1}\{\mathbf{1} - \mathbf{s} \odot \mathbf{y}_{\text{batch}}\}] \leftarrow \textbf{IsPositive}\left([\mathbf{1} - \mathbf{s} \odot \mathbf{y}_{\text{batch}}]^{(f)}\right)$
$\left[\frac{1}{M_{\text{batch}}}\mathbb{1}\{\mathbf{1} - \mathbf{s} \odot \mathbf{y}_{\text{batch}}\}\right]^{(f)} \leftarrow \frac{1}{M_{\text{batch}}} \cdot [\mathbb{1}\{\mathbf{1} - \mathbf{s} \odot \mathbf{y}_{\text{batch}}\}]$
$\left[\boldsymbol{\delta}^{(L+1)}\right]^{(f)} \leftarrow \textbf{Truncate}\left(-[\mathbf{y}_{\text{batch}}]^{(f)} \odot \left[\frac{1}{M_{\text{batch}}}\mathbb{1}\{\mathbf{1} - \mathbf{s} \odot \mathbf{y}_{\text{batch}}\}\right]^{(f)}, k+f, f\right)$

/* Backpropagation. */
for $l = L+1, \ldots, 1$ **do**
  if $l = 1$ **then**
    $\left[\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}}\right]^{(f)} \leftarrow \textbf{Truncate}\left([\boldsymbol{\delta}^{(1)}]^{(f)}[\mathbf{X}_{\text{batch}}^{\mathrm{T}}]^{(f)}, k+f, f\right)$
  else
    $\left[\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(l)}}\right]^{(f)} \leftarrow \textbf{Truncate}\left([\boldsymbol{\delta}^{(l)}]^{(f)}[\mathbf{Z}^{(l-1)\mathrm{T}}]^{(f)}, k+f, f\right)$
    $\left[\mathbf{W}^{(l)\mathrm{T}}\boldsymbol{\delta}^{(l)}\right]^{(f)} \leftarrow \textbf{Truncate}\left([\mathbf{W}^{(l)\mathrm{T}}]^{(f)}[\boldsymbol{\delta}^{(l)}]^{(f)}, k+f, f\right)$
    $\left[\mathbb{1}\{\mathbf{Z}^{(l-1)} > 0\}\right] \leftarrow \textbf{IsPositive}\left([\mathbf{Z}^{(l-1)\mathrm{T}}]^{(f)}\right)$
    $\left[\boldsymbol{\delta}^{(l-1)}\right]^{(f)} \leftarrow [\mathbf{W}^{(l)\mathrm{T}}\boldsymbol{\delta}^{(l)}]^{(f)} \odot [\mathbb{1}\{Z^{(l-1)} > 0\}]$
  end
  $\left[\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(l)}}\right]^{(f)} \leftarrow [\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(l)}}]^{(f)} + \textbf{Truncate}\left(\lambda[\mathbf{W}^{(l)}]^{(f)}, k+f, f\right)$
  $\left[\frac{\partial \mathcal{J}}{\partial \mathbf{b}^{(l)}}\right]^{(f)} \leftarrow [\boldsymbol{\delta}^{(l)}]^{(f)}\mathbf{1}$
end

/* Parameter updates. */
for $l = 1, \ldots, L+1$ **do**
  $[(\mathbf{W}_v^{(l)})_{\text{prev}}]^{(f)} \leftarrow [\mathbf{W}_v^{(l)}]^{(f)}$
  $[\mathbf{W}_v^{(l)}]^{(f)} \leftarrow \textbf{Truncate}\left(\mu[\mathbf{W}_v^{(l)}]^{(f)} - \alpha\left[\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(l)}}\right]^{(f)}, k+f, f\right)$
  $[\mathbf{W}^{(l)}]^{(f)} \leftarrow$
   $[\mathbf{W}^{(l)}]^{(f)} + \textbf{Truncate}\left(-\mu[(\mathbf{W}_v^{(l)})_{\text{prev}}]^{(f)} + (\mu+1)[\mathbf{W}_v^{(l)}]^{(f)}, k+f, f\right)$
  $[(\mathbf{b}_v^{(l)})_{\text{prev}}]^{(f)} \leftarrow [\mathbf{b}_v^{(l)}]^{(f)}$
  $[\mathbf{b}_v^{(l)}]^{(f)} \leftarrow \textbf{Truncate}\left(\mu[\mathbf{b}_v^{(l)}]^{(f)} - \alpha\left[\frac{\partial \mathcal{J}}{\partial \mathbf{b}^{(l)}}\right]^{(f)}, k+f, f\right)$
  $[\mathbf{b}^{(l)}]^{(f)} \leftarrow [\mathbf{b}^{(l)}]^{(f)} + \textbf{Truncate}\left(-\mu[(\mathbf{b}_v^{(l)})_{\text{prev}}]^{(f)} + (\mu+1)[\mathbf{b}_v^{(l)}]^{(f)}, k+f, f\right)$
end

return $[\mathbf{W}]^{(f)}, [\mathbf{b}]^{(f)}, [\mathbf{W}_v]^{(f)}, [\mathbf{b}_v]^{(f)}$

observed that our model performance is only minorly affected by small changes to the parameters. Although we considered all hyperparameters as fixed at the beginning of our secure protocol, additionally performing parameter selection in a secure computation framework would take only a few weeks over a WAN based on our estimates and thus remains practical. This procedure can be further sped up with parallelism or more advanced search strategies.

C.2 Experimental validation details

We obtained the compounds droloxifene, imatinib, nutlin-3, actinomycin D, and Hoescht 33258 from Abcam and the compounds GW-501516 and CHEMBL2332055 from Sigma-Aldrich, which were sent to commercial contract research organizations (CROs) for experimental validation.

Predictions for ERα, ERβ, and PgR were validated using Indigo Bioscience's nuclear receptor assay service. First, a suspension of Reporter Cells was prepared in INDIGO's Cell Recovery Medium (CRM; containing 10% Charcoal-stripped FBS). Immediately prior to assay setup, test compound master stocks were diluted in DMSO to generate solutions at 1,000x-concentration relative to each final treatment concentration. These intermediate stocks were subsequently diluted directly into INDIGO's Compound Screening Medium (CSM; containing 10% Charcoal-stripped FBS) to generate 2X-concentration treatment media. 100 µl of each prepared treatment medium was dispensed into duplicate assay wells pre-dispensed with a 100 µl suspension of Reporter Cells, thereby achieving the desired final treatment concentrations. The concentration of residual DMSO in all assay wells was 0.1%. Assay plates were incubated for 22–24 hours in a cell culture incubator (37°C, 5% CO2, 85% humidity). The reporter cell suspension was first supplemented with a 2X-EC80 concentration of the appropriate reference agonists, and then 100 µl of the cell suspension was dispensed into wells of a white 96-well assay plate. Wells were rinsed once with LCM buffer, then LCM substrate was added. Following incubation at 37°C for 30 minutes, fluorescence was measured to determine relative number of live cells per assay well. LCM Substrate

was then discarded and 100 µl/well of Luciferase Detection Reagent was added. RLUs were quantified from each assay well to determine antagonist activity.

Predictions for PARP1 were made using BPS Bioscience's PARP1 biochemical assay and screening service. The enzymatic reactions were conducted in duplicate at room temperature for 1 hour in a 96 well plate coated with histone substrate. 50 µL of reaction buffer (Tris·HCl, pH 8.0) contains NAD^+, biotinylated NAD^+, activated DNA, a PARP enzyme and the test compound. After enzymatic reactions, 50µl of Streptavidin-horseradish peroxidase was added to each well and the plate was incubated at room temperature for an additional 30 minutes. 100 µl of developer reagents were added to wells and luminescence was measured using a BioTek SynergyTM 2 microplate reader. The luminescence data were analyzed using the computer software, Graphpad Prism. In the absence of the compound, the luminescence in each data set was defined as 100% activity. In the absence of the PARP, the luminescence in each data set was defined as 0% activity. The IC50 value was determined by the concentration causing a half-maximal percent activity.

Predictions for ERBB3 and ERBB4 were validated using DiscoverX's KINOMEscan profiling service. Kinase-tagged T7 phage strains were prepared in an *E. coli* host derived from the BL21 strain. *E. coli* were grown to log-phase and infected with T7 phage and incubated with shaking at 32°C until lysis. The lysates were centrifuged and filtered to remove cell debris. The remaining kinases were produced in HEK-293 cells and subsequently tagged with DNA for qPCR detection. Streptavidin-coated magnetic beads were treated with biotinylated small molecule ligands for 30 minutes at room temperature to generate affinity resins for kinase assays. The liganded beads were blocked with excess biotin and washed with blocking buffer [SeaBlock (Pierce), 1% BSA, 0.05% Tween 20, 1 mM DTT] to remove unbound ligand and to reduce nonspecific binding. Binding reactions were assembled by combining kinases, liganded affinity beads, and test compounds in 1X binding buffer (20% SeaBlock, 0.17X PBS, 0.05% Tween 20, 6 mM DTT). Test compounds were prepared as 111X stocks in 100% DMSO. Kds were determined using an 11-point 3-fold compound dilution series with three DMSO control points. All compounds for Kd measurements are distributed by

acoustic transfer (non-contact dispensing) in 100% DMSO. The compounds were then diluted directly into the assays such that the final concentration of DMSO was 0.9%. All reactions performed in polypropylene 384-well plate. Each was a final volume of 0.02 ml. The assay plates were incubated at room temperature with shaking for 1 hour and the affinity beads were washed with wash buffer (1X PBS, 0.05% Tween 20). The beads were then re-suspended in elution buffer (1X PBS, 0.05% Tween 20, 0.5 μM non-biotinylated affinity ligand) and incubated at room temperature with shaking for 30 minutes. The kinase concentration in the eluates was measured by qPCR. Kd calculation was performed by fitting a standard Hill curve to the measured kinase concentrations.

Predictions for GRM1 and GRM5 were validated using DiscoverX's gpcrSCAN profiling service. PathHunter cell lines were seeded in a total volume of 20 μl into black-walled, clear-bottom, Poly-D-lysine coated 384-well microplates and incubated at 37°C for the appropriate time prior to testing. Assays were performed in 1X Dye Loading Buffer consisting of 1X Dye, 1X Additive A and 2.5 mM Probenecid in HBSS or 20 mM Hepes. Probenicid was prepared fresh. Cells were loaded with dye prior to testing. Media was aspirated from cells and replaced with 20 μl Dye Loading Buffer. Cells were incubated for 30-60 minutes at 37°C. For antagonist determination, cells were pre-incubated with sample followed by agonist challenge at the EC80 concentration. Intermediate dilution of sample stocks was performed to generate 3X sample in assay buffer. After dye loading, cells were removed from the incubator and 10 μl 3X sample was added. Cells were incubated for 30 minutes at room temperature in the dark to equilibrate plate temperature. Vehicle concentration was 1%. Compound antagonist activity was measured on a FLIPR Tetra (MDS). Calcium mobilization was monitored for 2 minutes and 10 μl EC80 agonist in HBSS or 20 mM Hepes was added to the cells 5 seconds into the assay.

Appendix D

Deferred details from Chapter 7

D.1 Biochemical validation details

Machine learning models were trained on all compound-kinase pairs from Davis et al. [DHH+11]. For IRAK4, c-SRC, and p110δ, we trained a GP with high uncertainty weight ($\beta = 20$) and an MLP. For PknB, the model/acquisition parameter settings were: (1) a GP without considering uncertainty ($\beta = 0$), (2) a GP with moderate uncertainty weight ($\beta = 1$), (3) a GP with high uncertainty weight ($\beta = 20$), (4) an MLP without uncertainty, (5) an MLP + GP with moderate uncertainty weight ($\beta = 1$), and (6) an MLP + GP with high uncertainty weight ($\beta = 20$). Compounds from the ZINC/Cayman dataset were featurized using the same pretrained JTNN-VAE as in the cross-validation experiment and concatenated with the feature vector for the corresponding kinase (PknB, IRAK4, c-SRC, or p110δ). Trained models were evaluated on these concatenated features. The top five predictions for each kinase from each of the above models were acquired for binding affinity determination. Predictions involving lipids only commercially available as ethanol solutions were incompatible with the binding assay, excluded from validation, and reported as not interactive.

Compounds were acquired directly from Cayman Chemical. All supplied compounds were tested to ensure $\geq 98\%$ purity. We leveraged the kinase affinity assays provided by the DiscoverX CRO. Kd determination was done using the KdELECT

assay, which measures the ability for test compounds to compete with an immobilized, active-site directed ligand using DNA-tagged kinase, where competition is measured via quantitative polymerase chain reaction (qPCR) of the DNA tag. Kinase-tagged T7 phage strains were prepared in an *Escherichia coli* (E. coli) host derived from the BL21 strain. E. coli were grown to log-phase and infected with T7 phage and incubated with shaking at 32°C until lysis. The lysates were centrifuged and filtered to remove cell debris. Streptavidin-coated magnetic beads were treated with biotinylated ligand for 30 minutes at room temperature to generate affinity resins for kinase assays. The liganded beads were blocked with excess biotin and washed with blocking buffer [SeaBlock (Pierce), 1% bovine serum albumin (BSA), 0.05% Tween 20, 1 mM dithiothreitol (DTT)] to remove unbound ligand and to reduce non-specific binding.

Binding reactions were assembled by combining kinases, liganded affinity beads, and test compounds in 1× binding buffer [20% SeaBlock, 0.17× phosphate-buffered saline (PBS), 0.05% Tween 20, 6 mM DTT]. Test compounds were prepared as 111X stocks in 100% DMSO. Kds were determined using an 11-point 3-fold compound dilution series with three DMSO control points with a top test compound concentration of 10,000 nM. All compounds for Kd measurements are distributed by acoustic transfer (non-contact dispensing) in 100% DMSO. The compounds were then diluted directly into the assays such that the final concentration of DMSO was 0.9%. All reactions performed in polypropylene 384-well plate. Each was a final volume of 0.02 mL. The assay plates were incubated at room temperature with shaking for 1 hour and the affinity beads were washed with wash buffer (1× PBS, 0.05% Tween 20). The beads were then re-suspended in elution buffer (1× PBS, 0.05% Tween 20, 0.5 μM non-biotinylated affinity ligand) and incubated at room temperature with shaking for 30 minutes. The kinase concentration in the eluates was measured by qPCR.

Kds were calculated with a standard dose-response curve using the Hill equation

$$\text{Response} = \text{Background} + \frac{\text{Signal} - \text{Background}}{1 + \frac{\text{Kd}^{\text{Hill slope}}}{\text{Dose}^{\text{Hill slope}}}}.$$

Curves were fitted using a non-linear least square fit with the Levenberg-Marquardt

algorithm. The Hill slope was set to -1; a deviation from this Hill slope in the dose-response pattern was used to identify possible aggregation, but no such deviation was observed.

D.2 Microbiological validation details

D.2.1 Mycobacterium tuberculosis model

We utilized wild-type H37Rv and H37Rv expressing an integrated copy of the luxABCDE cassette which enables mycobacteria to endogenously produce light [AZF$^+$10]; monitoring luminescence of the latter strain has been demonstrated to correlate well with the standard colony forming unit assay [BTZ$^+$17].

D.2.2 Human macrophage model

Human monocytes were isolated from human buffy coats purchased from the Massachusetts General Hospital blood bank using a standard Ficoll gradient (GE Healthcare) and subsequent positive selection of CD14+ cells (Stemcell Technologies). Selected monocytes were cultured in ultra-low-adherence flasks (Corning) for 6 days with RPMI media (Invitrogen) supplemented with hydroxyethylpiperazine ethane sulfonic acid (HEPES) (Invitrogen), L-glutamine (Invitrogen), 10% heat-inactivated fetal bovine serum (FBS) (Invitrogen) and 25 ng/mL human macrophage colony-stimulating factor (M-CSF) (Biolegend).

D.2.3 Axenic Mtb growth inhibition assay

H37Rv Mtb growth was evaluated using the resazurin viability assay (alamar blue). Mtb was grown to an optical density (OD) corresponding to early log phase (OD 0.4) and back-diluted to an optical density of 0.003 in 7H9 media supplemented with oleic albumin dextrose catalase (OADC) prior to incubation with a range of concentrations of K252a, TG101209, SU11652, and rifampicin or vehicle control in a 96 well plate with shaking at 37°C. Bacteria were incubated with drug alone for 72 hours prior to

the addition of alamar blue. After addition of alamar blue, H37Rv was incubated for an additional 48 hours and alamar blue absorbance was measured using a Tecan Spark 10M. Normalized alamar blue absorbance was calculated as

$$\frac{(o_2 \times a_1) - (o_1 \times a_2)}{(o_2 \times p_1) - (o_1 \times p_2)}$$

where $o_1 = 80586$ is the molar extinction coefficient of oxidized alamar blue at 570 nm; $o_2 = 117216$ is the molar extinction coefficient of oxidized alamar blue at 600 nm; a_1 and a_2 are the measured absorbance of the test well at 570 nm and 600 nm, respectively; and p_1 and p_2 are the measured absorbance of a positive growth control well at 570 nm and 600 nm, respectively. For each compound, we assessed bacterial growth at 1.25, 2.5, 5, 10, 25, and 50 μM to determine the MIC.

Additionally, Mycobacterium tuberculosis strain H37Rv bacteria expressing an integrated copy of the luxABCDE cassette (Andreu et al., 2010) were grown to mid-log phase and diluted to an optical density of 0.006. Mycobacteria were added to wells of a 96-well solid white polystyrene plate and incubated with a vehicle control (DMSO) or rifampicin, TG101209, or SU11652 (Cayman Chem) for 5 days. Plates were sealed with breathable film (VWR) and incubated at 37°C for 4 days with shaking. On day 5, we measured luminescence as a proxy for total bacterial burden.

D.2.4 Primary human macrophage culture

Deidentified buffy coats from healthy human donors were obtained from Massachusetts General Hospital. Peripheral blood mononuclear cells (PBMCs) were isolated from buffy coats by density-based centrifugation using Ficoll (GE Healthcare). CD14+ monocytes were isolated from PBMCs using a CD14 positive-selection kit (Stemcell). Isolated monocytes were differentiated to macrophages in RPMI 1640 (ThermoFisher Scientific) supplemented with 10% heat-inactivated fetal FBS (ThermoFisher Scientific), 1% HEPES, and 1% L-glutamine. Media was further supplemented with 25 ng/mL M-CSF (Biolegend, MCSF: 572902). Monocytes were cultured on low-adhesion tissue culture plates (Corning) for 6 days. After 6 days, macrophages were

detached using a detachment buffer of 1X Ca-free PBS and 2 mM ethylenediaminetetraacetic acid (EDTA), pelleted, and recounted. Macrophages were plated in tissue culture-treated 96-well solid white polystyrene plates at a density of 50,000 cells per well in maintenance media (RPMI, 10% heat-inactivated FBS, 1% HEPES, and 1% L-glutamine) and allowed to re-adhere overnight.

D.2.5 Intra-macrophage Mtb growth inhibition assay

H37Rv Mtb expressing the luxABCDE cassette were grown to an optical density of 0.4 and centrifuged briefly. Mtb were resuspended in pre-warmed maintenance media and filtered through a 5 μM filter to remove clumped bacteria and generate a single-cell suspension. Macrophages were infected at a multiplicity of infection of 3 bacteria to 1 macrophage in 100 μL per well and phagocytosis was allowed to proceed for 4 hours prior to washing macrophages twice with pre-warmed maintenance media to remove extracellular bacteria. Following phagocytosis and washing, cells were incubated with media containing a vehicle control (DMSO) or rifampicin, K252a, or SU11652 (Cayman Chem) for 5 days. On day 5, we measured luminescence as a proxy of intracellular bacterial burden as previously described [AZF$^+$10, BTZ$^+$17] using a high-throughput luminometer.

Appendix E

Deferred details from Chapter 8

E.1 Additional baseline method details

We benchmark our escape prediction experiments against models that try to estimate the evolutionary fitness of a viral protein based on some assumptions. Notably, viral fitness models are not equivalent to escape prediction, since mutations that preserve fitness may be neutral with respect to escape (fitness models better correspond to the "grammaticality" term in CSCS). However, in the absence of existing unsupervised models that are directly built to perform unsupervised escape prediction, viral fitness models are the most related that attempt to solve a conceptually close problem.

E.1.1 Alignment-based frequency fitness model

This baseline model for viral fitness assumes that higher mutational frequencies in a corpus correspond to higher fitness and that residue-level fitness information is independent across the viral sequence; this fitness model is widely adopted due to its simplicity [CS07, KBB+13, AAG+05, FKG+11].

We first perform MSA with the MAFFT software package (version 7.453) within the respective corpuses (influenza or HIV sequences). After sequence alignment was performed, we considered each position in the viral sequence of interest (influenza

strains `A/Perth/16/2009` or `A/WSN/1933`, or HIV strain `BG505`). At a given position, we computed the frequency of other amino acids that were aligned to that position across all other sequences in the corpus. Sequences were acquired based on the highest observed frequencies across all possible single-residue mutations.

For influenza, we found that performance (in terms of normalized AUC) improved when restricting sequence alignment to the corresponding subtype (H1 sequences for `A/WSN/1933` and H3 sequences for `A/Perth/16/2009`) For HIV, we found that performance improved when only restricting alignments to the local neighborhood of `BG505.T332N`, defined by sequences that differ by a maximum of 15 residues. In general, we found that sequence alignment is dramatically affected by the sequences that are included in the corpus. For a best-case comparison, we report the highest performance over different sequence inclusion strategies.

We also used a conceptually similar implementation of this strategy provided by the EVcouplings pipeline [HGS+19] (`https://github.com/debbiemarkslab/EVcouplings`) using default parameters. We trained the EVcouplings independent model on the same corpus of viral sequences used to train our language models.

E.1.2 Alignment-based Potts model

A common critique of the above strategy for modelling viral fitness is that the independence assumption is limiting. Biologically, two residues can co-evolve, especially if they are physically and biochemically related in the three-dimensional structure of the protein, a phenomenon referred to as "epistasis." A solution is to incorporate pairwise residue information by learning a probabilistic model in which each residue position corresponds to a random variable and pairwise potentials can encode epistatic relationships.

Hopf et al. learned such a model based on a Potts model formulation; we describe the general formulation here and leave implementation details to the original paper [HIP+17]. Given a sequence $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ where x_i comes from an alphabet $\mathcal{X}$ that is the set of all amino acids and a gap character, the model assigns an energy

score to each sequence as

$$E(\mathbf{x}; \mathbf{h}, \mathbf{J}) \triangleq \sum_{i=1}^{N} h_i x_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} J_{ij} x_i x_j.$$

This term is scaled to be a valid probability distribution

$$p(\mathbf{x}; \mathbf{h}, \mathbf{J}) = \frac{1}{Z} \exp\left\{-E(\mathbf{x}; \mathbf{h}, \mathbf{J})\right\}$$

where $Z = \sum_{\mathbf{x}'} \exp\left\{-E(\mathbf{x}'; \mathbf{h}, \mathbf{J})\right\}$. The parameters are learned by a maximum likelihood procedure using a number of critical heuristics that Hopf et al. use to allow for efficient inference and parameter regularization [HIP+17, HGS+19]. We use the pipeline provided by Hopf et al. at `https://github.com/debbiemarkslab/EVcouplings` with default parameters. We trained the EVcouplings epistatic model on the same corpus of viral sequences used to train our language models.

E.1.3 Pretrained sequence embedding models

We tested if the sequence embeddings produced by models trained on generic protein sequence corpuses [BB19, RBT+19, AKB+19] would be informative with respect to escape. We used the pretrained transformer model from Rao et al. [RBT+19] and the pretrained UniRep model from Alley et al. [AKB+19], both obtained through `https://github.com/songlab-cal/tape`. We used the pretrained model with full soft symmetric alignment and protein structure information from Bepler et al. [BB19], available through `https://github.com/tbepler/protein-sequence-embedding-iclr2019`.

Rather than training exclusively on a large viral sequence corpus, as we did, these methods trained on corpuses containing generic protein sequences. Each single-residue escape mutant was embedded using the pretrained model and mutant sequences were acquired based on the largest changes to the embedding based on the ℓ_1-distance.

E.2 Additional experimental details

E.2.1 Language model hyperparameter selection

We performed a small-scale grid search using categorical cross entropy loss after 20 training epochs on the headline and influenza datasets to select the language model architecture and hyperparameters based on a random 80%/20% cross-validation split of the training set. Hyperparameter ranges were influenced by previous applications of recurrent architectures to protein sequence representation learning [BB19]. We tested hidden unit dimensions of 128, 256, and 512. We tested architectures with one or two hidden layers. We tested three hidden-layer architectures: a densely connected neural network with access to both left and right sequence contexts, an LSTM with access to only the left context, and a BiLSTM with access to both left and right sequence contexts. We tested two Adam learning rates (0.01 and 0.001). All other architecture details described in Section 8.4.2 were fixed to reasonable defaults. In total, we tested 36 conditions and ultimately used a BiLSTM architecture with two hidden layers of 512 hidden units each, with an Adam learning rate of 0.001. We used the same architecture for all experiments. In general, we noted that increasing model capacity only served to improve performance.

Bibliography

[AAB+20] Ricard Argelaguet, Damien Arnol, Danila Bredikhin, Yonatan Deloro, Britta Velten, John C. Marioni, and Oliver Stegle. MOFA+: A statistical framework for comprehensive integration of multi-modal single-cell data. *Genome Biology*, 21(1):1–17, 2020.

[AAG+05] T. M. Allen, M. Altfeld, S. C. Geer, E. T. Kalife, C. Moore, K. M. O'Sullivan, I. DeSouza, M. E. Feeney, R. L. Eldridge, E. L. Maier, D. E. Kaufmann, M. P. Lahaie, L. Reyor, G. Tanzi, M. N. Johnston, C. Brander, R. Draenert, J. K. Rockstroh, H. Jessen, E. S. Rosenberg, S. A. Mallal, and B. D. Walker. Selective Escape from CD8+ T-Cell Responses Represents a Major Driving Force of Human Immunodeficiency Virus Type 1 (HIV-1) Sequence Diversity and Reveals Constraints on HIV-1 Evolution. *Journal of Virology*, 79(21):13239–13249, 2005.

[ABD+11] Hee Kap Ahn, Sang Won Bae, Erik D. Demaine, Martin L. Demaine, Sang Sub Kim, Matias Korman, Iris Reinbacher, and Wanbin Son. Covering points by disjoint boxes with outliers. *Computational Geometry: Theory and Applications*, 44(3):178–190, 2011.

[AHK01] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the Surprising Behavior of Distance Metrics in High Dimensional Space. 1973(27):420–434, 2001.

[AIR18] Alexandr Andoni, Piotr Indyk, and Ilya Razenshteyn. Approximate nearest neighbor search in high dimensions. *arXiv*, cs.DS:1806.09823, 2018.

[AJS12] Kathryn Twigg Arrildt, Sarah Beth Joseph, and Ronald Swanstrom. The HIV-1 Env protein: A coat of many colors. *Current HIV/AIDS Reports*, 9(1):52–63, 2012.

[Aka01] S. Akaho. A kernel method for canonical correlation analysis. In *In Proceedings of the International Meeting of the Psychometric Society*. Springer-Verlag, 2001.

[AKB+19] Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. Unified rational protein engineering

with sequence-based deep representation learning. *Nature Methods*, 16(12):1315–1322, 2019.

[ANS16] Dominique Attali, Tuong-Bach Nguyen, and Isabelle Sivignon. Epsilon-covering is NP-complete. In *European Workshop on Computational Geometry*, 2016.

[AOS+16] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv*, cs.AI:1606.06565, 2016.

[ARL+20] Kristian G. Andersen, Andrew Rambaut, W. Ian Lipkin, Edward C. Holmes, and Robert F. Garry. The proximal origin of SARS-CoV-2. *Nature Medicine*, 26(4):450–452, 2020.

[ASP+14] Khaled Ali, Dalya R. Soond, Roberto Piñeiro, Thorsten Hagemann, Wayne Pearce, Ee Lyn Lim, Hicham Bouabe, Cheryl L. Scudamore, Timothy Hancox, Heather Maecker, Lori Friedman, Martin Turner, Klaus Okkenhaug, and Bart Vanhaesebroeck. Inactivation of PI(3)K p110δ breaks regulatory T-cell-mediated immune tolerance to cancer. *Nature*, 510(7505):407–411, 2014.

[AST+17] Philipp Angerer, Lukas Simon, Sophie Tritschler, F. Alexander Wolf, David Fischer, and Fabian J. Theis. Single cells make big data: New challenges and opportunities in transcriptomics. *Current Opinion in Systems Biology*, 4:85–91, 2017.

[AV07] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[AZF+10] Nuria Andreu, Andrea Zelmer, Taryn Fletcher, Paul T. Elkington, Theresa H. Ward, Jorge Ripoll, Tanya Parish, Gregory J. Bancroft, Ulrich Schaible, Brian D. Robertson, and Siouxsie Wiles. Optimisation of bioluminescent reporters for use with mycobacteria. *PLoS ONE*, 5(5):e10777, 2010.

[Bac20] Francis Bacon. *The New Organon*. Cambridge Texts in the History of Philosophy (ed. M. Silverthorne and L. Jardine, 2000), 1620.

[BB19] Tristan Bepler and Bonnie Berger. Learning protein sequence embeddings using information from structure. *7th International Conference on Learning Representations*, arXiv(cs.LG):1902.08661, 2019.

[BBLP13] Yoshua Bengio, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. Advances in optimizing recurrent networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 8624–8628, 2013.

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[Bea91] Donald Beaver. Efficient Multiparty Protocols Using Circuit Randomization. *Crypto*, 576(814):420–432, 1991.

[BEP13] Kendrick Boyd, Kevin H. Eng, and C. David Page. Area under the precision-recall curve: Point estimates and confidence intervals. In *Lecture Notes in Computer Science*, volume 8190 LNAI, pages 451–466, 2013.

[BFW+20] Alina Baum, Benjamin O Fulton, Elzbieta Wloga, Richard Copin, Kristen E Pascal, Vincenzo Russo, Stephanie Giordano, Kathryn Lanza, Nicole Negron, Min Ni, Yi Wei, Gurinder S Atwal, Andrew J Murphy, Neil Stahl, George D Yancopoulos, and Christos A Kyratsous. Antibody cocktail to SARS-CoV-2 spike protein prevents rapid mutational escape seen with individual antibodies. *Science*, 369(6506):1014–1018, 2020.

[BGLL08] Vincent D. Blondel, Jean Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[BHEK+02] A. Buzdar, D. Hayes, A. El-Khoudary, S. Yan, P. Lønning, M. Lichinitser, R. Gopal, G. Falkson, K. Pritchard, A. Lipton, K. Wolter, A. Lee, K. Fly, R. Chew, M. Alderdice, K. Burke, and P. Eisenberg. Phase III randomized trial of droloxifene and tamoxifen as first-line endocrine treatment of ER/PgR-positive advanced breast cancer. *Breast Cancer Research and Treatment*, 73(2):161–175, 2002.

[BHS14] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *arXiv*, cs.LG:1306.6709, 2014.

[BHS+18] Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 36(5):411–420, 2018.

[BMM+19] Metin Balaban, Niema Moshiri, Uyen Mai, Xingfan Jia, and Siavash Mirarab. TreeCluster: Clustering biological sequences using phylogenetic trees. *PLoS ONE*, 14(8):e0221068, 2019.

[BOGW88] M Ben-Or, S Goldwasser, and A Wigderson. Completeness Theorems for Non-Cryptographic Fault Tolerant Distributed Computation. *Proceedings of the 20th Annual ACM Symposium on the Theory of Computing*, pages 1–10, 1988.

[BOS+10] Dan H. Barouch, Kara L. O'Brien, Nathaniel L. Simmons, Sharon L. King, Peter Abbink, Lori F. Maxfield, Ying Hua Sun, Annalena La Porte, Ambryice M. Riggs, Diana M. Lynch, Sarah L. Clark, Katherine Backus, James R. Perry, Michael S. Seaman, Angela Carville, Keith G. Mansfield, James J. Szinger, Will Fischer, Mark Muldoon, and Bette Korber. Mosaic HIV-1 vaccines expand the breadth and depth of cellular immune responses in rhesus monkeys. *Nature Medicine*, 16(3):319–323, 2010.

[BRB+15] Trevor Bedford, Steven Riley, Ian G. Barr, Shobha Broor, Mandeep Chadha, Nancy J. Cox, Rodney S. Daniels, C. Palani Gunasekaran, Aeron C. Hurt, Anne Kelso, Alexander Klimov, Nicola S. Lewis, Xiyan Li, John W. McCauley, Takato Odagiri, Varsha Potdar, Andrew Rambaut, Yuelong Shu, Eugene Skepner, Derek J. Smith, Marc A. Suchard, Masato Tashiro, Dayan Wang, Xiyan Xu, Philippe Lemey, and Colin A. Russell. Global circulation patterns of seasonal influenza viruses vary with antigenic drift. *Nature*, 523(7559):217–220, 2015.

[Bre03] P. J. Brennan. Structure, function, and biogenesis of the cell wall of Mycobacterium tuberculosis. *Tuberculosis*, 83(1-3):91–97, 2003.

[BS09] José M. Bernardo and Adrian F.M. Smith. *Bayesian Theory*. John Wiley & Sons, Ltd, 2009.

[BSB+13] Dan H. Barouch, Kathryn E. Stephenson, Erica N. Borducchi, Kaitlin Smith, Kelly Stanley, Anna G. McNally, Jinyan Liu, Peter Abbink, Lori F. Maxfield, Michael S. Seaman, Anne Sophie Dugast, Galit Alter, Melissa Ferguson, Wenjun Li, Patricia L. Earl, Bernard Moss, Elena E. Giorgi, James J. Szinger, Leigh Anne Eller, Erik A. Billings, Mangala Rao, Sodsai Tovanabutra, Eric Sanders-Buell, Mo Weijtens, Maria G. Pau, Hanneke Schuitemaker, Merlin L. Robb, Jerome H. Kim, Bette T. Korber, and Nelson L. Michael. Protective efficacy of a global HIV-1 mosaic vaccine against heterologous SHIV challenges in rhesus monkeys. *Cell*, 155(3):531–539, 2013.

[BTW+18] Dan H. Barouch, Frank L. Tomaka, Frank Wegmann, Daniel J. Stieh, Galit Alter, Merlin L. Robb, Nelson L. Michael, Lauren Peter, Joseph P. Nkolola, Erica N. Borducchi, Abishek Chandrashekar, David Jetton, Kathryn E. Stephenson, Wenjun Li, Bette Korber, Georgia D. Tomaras, David C. Montefiori, Glenda Gray, Nicole Frahm, M. Juliana McElrath, Lindsey Baden, Jennifer Johnson, Julia Hutter, Edith Swann, Etienne Karita, Hannah Kibuuka, Juliet Mpendo, Nigel Garrett, Kathy Mngadi, Kundai Chinyenze, Frances Priddy, Erica Lazarus, Fatima Laher, Sorachai Nitayapan, Punnee Pitisuttithum, Stephan Bart, Thomas Campbell, Robert Feldman, Gregg Lucksinger, Caroline Borremans, Katleen Callewaert, Raphaele Roten, Jerald Sadoff, Lorenz Scheppler,

Mo Weijtens, Karin Feddes-de Boer, Daniëlle van Manen, Jessica Vreugdenhil, Roland Zahn, Ludo Lavreys, Steven Nijs, Jeroen Tolboom, Jenny Hendriks, Zelda Euler, Maria G. Pau, and Hanneke Schuitemaker. Evaluation of a mosaic HIV-1 vaccine in a multicentre, randomised, double-blind, placebo-controlled, phase 1/2a clinical trial and in rhesus monkeys. *The Lancet*, 392(10143):232–243, 2018.

[BTZ+17] Magdalena K. Bielecka, Liku B. Tezera, Robert Zmijan, Francis Drobniewski, Xunli Zhang, Suwan Jayasinghe, and Paul Elkington. A bioengineered three-dimensional cell culture platform integrated with microfluidics to address antimicrobial resistance in tuberculosis. *mBio*, 8(1):e02073–16, 2017.

[BVW+16] Maayan Baron, Adrian Veres, Samuel L. Wolock, Aubrey L. Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K. Wagner, Shai S. Shen-Orr, Allon M. Klein, Douglas A. Melton, and Itai Yanai. A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure. *Cell Systems*, 3(4):346–360, 2016.

[CBP16] Hyunghoon Cho, Bonnie Berger, and Jian Peng. Compact Integration of Multi-Network Topology for Functional Analysis of Genes. *Cell Systems*, 3(6):P540–548.e5, 2016.

[CCR+18] Junyue Cao, Darren A Cusanovich, Vijay Ramani, Delasa Aghamirzaie, Hannah A Pliner, Andrew J Hill, Riza M Daza, Jose L McFaline-Figueroa, Jonathan S Packer, Lena Christiansen, Frank J Steemers, Andrew C Adey, Cole Trapnell, and Jay Shendure. Joint profiling of chromatin accessibility and gene expression in thousands of single cells. *Science*, 361(6409):1380–1385, 2018.

[CDCPGS09] José M. Cuevas, Pilar Domingo-Calap, Marianoel Pereira-Gómez, and Rafael Sanjuán. Experimental Evolution and Population Genetics of RNA Viruses. *The Open Evolution Journal*, 3(1):9–16, 2009.

[CDN15] Ronald Cramer, Ivan Bjerre Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, Cambridge, UK, 1 edition, 2015.

[Cha02] Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 380–388, 2002.

[Chv79] V. Chvatal. A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research*, 1979.

[CJS18] Irene Y. Chen, Fredrik D. Johansson, and David Sontag. Why is my classifier discriminatory? *Proceedings of the 32nd International*

Conference on Neural Information Processing Systems, pages 3539–3550, 2018.

[CLH+13] Murat Can Cobanoglu, Chang Liu, Feizhuo Hu, Zoltán N. Oltvai, and Ivet Bahar. Predicting drug-target interactions using probabilistic matrix factorization. *Journal of Chemical Information and Modeling*, 53(12):3399–3409, 2013.

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.

[COP+20] Junyue Cao, Diana R. O'Day, Hannah A. Pliner, Paul D. Kingsley, Mei Deng, Riza M. Daza, Michael A. Zager, Kimberly A. Aldinger, Ronnie Blecher-Gonen, Fan Zhang, Malte Spielmann, James Palis, Dan Doherty, Frank J. Steemers, Ian A. Glass, Cole Trapnell, and Jay Shendure. A human cell atlas of fetal gene expression. *Science*, 370(6518):eaba7721, 2020.

[CPG+14] Daniel K.W. Chu, Leo L.M. Poon, Mokhtar M. Gomaa, Mahmoud M. Shehata, Ranawaka A.P.M. Perera, Dina Abu Zeid, Amira S. El Rifay, Lewis Y. Siu, Yi Guan, Richard J. Webby, Mohamed A. Ali, Malik Peiris, and Ghazi Kayali. MERS coronaviruses in dromedary camels, Egypt. *Emerging Infectious Diseases*, 20(6):1049–1053, 2014.

[CR03] Thierry Calandra and Thierry Roger. Macrophage migration inhibitory factor: A regulator of innate immunity. *Nature Reviews Immunology*, 3(10):791–900, 2003.

[CRE+17] Hans Clevers, Susanne Refelski, Michael Elowitz, Allon Klein, Jay Shendure, Cole Trapnell, Ed Lein, Emma Lundberg, Matthias Uhlen, Alfonso Martinez-Aria, Joshua R. Sanes, James Eberwine, Paul Blainey, Junhyong Kim, and J. Christopher Love. What Is Your Conceptual Definition of "Cell Type" in the Context of a Mature Organism? *Cell Systems*, 4:255–259, 2017.

[CS07] John A. Capra and Mona Singh. Predicting functionally important residues from sequence conservation. *Bioinformatics*, 23(15):1875–1882, 2007.

[CS10] Octavian Catrina and Amitabh Saxena. Secure computation with fixed-point numbers. *Lecture Notes in Computer Science*, (6052 LNCS):35–50, 2010.

[CWB18] Hyunghoon Cho, David Wu, and Bonnie Berger. Secure genome-wide association analysis using multiparty computation. *Nature Biotechnology*, 36(6):547–551, 2018.

[DAW+19] Adam S. Dingens, Dana Arenz, Haidyn Weight, Julie Overbaugh, and Jesse D. Bloom. An Antigenic Atlas of HIV-1 Escape from Broadly Neutralizing Antibodies Distinguishes Functional and Structural Epitopes. *Immunity*, 50(2):520–532.e3, 2019.

[DB16] Michael B. Doud and Jesse D. Bloom. Accurate measurement of the effects of all amino-acid mutations on influenza hemagglutinin. *Viruses*, 8(6):155, 2016.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv*, cs.CL:1810.04805, 2018.

[DF08] Sanjoy Dasgupta and Yoav Freund. Random projection trees and low dimensional manifolds. In *Proceedings of the fourtieth annual ACM symposium on theory of computing*, page 537, 2008.

[DFGH+17] Pradyot Dash, Andrew J. Fiore-Gartland, Tomer Hertz, George C. Wang, Shalini Sharma, Aisha Souquette, Jeremy Chase Crawford, E. Bridie Clemens, Thi H.O. Nguyen, Katherine Kedzierska, Nicole L. La Gruta, Philip Bradley, and Paul G. Thomas. Quantifiable predictive features define epitope-specific T cell receptor repertoires. *Nature*, 547(7661):89–93, 2017.

[DHH+11] Mindy I. Davis, Jeremy P. Hunt, Sanna Herrgard, Pietro Ciceri, Lisa M. Wodicka, Gabriel Pallares, Michael Hocker, Daniel K. Treiber, and Patrick P. Zarrinkar. Comprehensive analysis of kinase inhibitor selectivity. *Nature Biotechnology*, 29(11):1046–1051, 2011.

[DJK+18] Kristofer Davie, Jasper Janssens, Duygu Koldere, Maxime De Waegeneer, Uli Pech, Łukasz Kreft, Sara Aibar, Samira Makhzami, Valerie Christiaens, Carmen Bravo González-Blas, Suresh Poovathingal, Gert Hulselmans, Katina I. Spanier, Thomas Moerman, Bram Vanspauwen, Sarah Geurs, Thierry Voet, Jeroen Lammertyn, Bernard Thienpont, Sha Liu, Nikos Konstantinides, Mark Fiers, Patrik Verstreken, and Stein Aerts. A Single-Cell Transcriptome Atlas of the Aging Drosophila Brain. *Cell*, 147(4):982–998.e20, 2018.

[DL15] Andrew M. Dai and Quoc V. Le. Semi-supervised sequence learning. *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pages 3079–3087, 2015.

[DLB18] Michael B. Doud, Juhye M. Lee, and Jesse D. Bloom. How single mutations affect viral escape from broad and narrow antibodies to H1 influenza hemagglutinin. *Nature Communications*, 9(1):1386, 2018.

[DOPB15] Steven G. Deeks, Julie Overbaugh, Andrew Phillips, and Susan Buchbinder. HIV infection. *Nature Reviews Disease Primers*, 1:15035, 2015.

[DOR+15] Tali Dekel, Shaul Oron, Michael Rubinstein, Shai Avidan, and William T. Freeman. Best-Buddies Similarity for robust template matching. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2021–2029, 2015.

[Dra91] J. W. Drake. A constant rate of spontaneous mutation in DNA-based microbes. *Proceedings of the National Academy of Sciences of the United States of America*, 88(16):7160–7164, 1991.

[DW51] J. Durbin and G. S. Watson. Testing for Serial Correlation in Least Squares Regression. II. *Biometrika*, 38(1/2):159–177, 1951.

[DY20] David DeTomaso and Nir Yosef. Identifying informative gene modules across modalities of single cell genomics. *bioRxiv*, 2020.

[EBC+10] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(1):625–660, 2010.

[EBE+09] Damian C. Ekiert, Gira Bhabha, Marc Andre Elsliger, Robert H.E. Friesen, Mandy Jongeneelen, Mark Throsby, Jaap Goudsmit, and Ian A. Wilson. Antibody recognition of a highly conserved influenza virus epitope. *Science*, 324(5924):246–251, 2009.

[Eis20] Michael Eisenstein. Active machine learning helps drug hunters tackle biology. *Nature Biotechnology*, 38(5):512, 2020.

[EJS18] Daniel Edsgärd, Per Johnsson, and Rickard Sandberg. Identification of spatial expression trends in single-cell gene expression data. *Nature Methods*, 15(5):339–342, 2018.

[EK01] Debra M. Eckert and Peter S. Kim. Mechanisms of Viral Membrane Fusion and Its Inhibition. *Annual Review of Biochemistry*, 70(1):777–810, 2001.

[FBP+15] Mary F. Fontana, Alyssa Baccarella, Nidhi Pancholi, Miles A. Pufall, De'Broski R. Herbert, and Charles C. Kim. JUNB Is a Key Transcriptional Modulator of Macrophage Activation. *The Journal of Immunology*, 194(1):177–186, 2015.

[FCE+16] Robert D. Finn, Penelope Coggill, Ruth Y. Eberhardt, Sean R. Eddy, Jaina Mistry, Alex L. Mitchell, Simon C. Potter, Marco Punta, Matloob Qureshi, Amaia Sangrador-Vegas, Gustavo A. Salazar, John Tate, and Alex Bateman. The Pfam protein families database: Towards a more sustainable future. *Nucleic Acids Research*, 44(D1):D279–D285, 2016.

[FCP19] Jennifer Furin, Helen Cox, and Madhukar Pai. Tuberculosis. *The Lancet*, 393(10181):1642–1656, 2019.

[Fir57] John Rupert Firth. *A Synopsis of Linguistic Theory, 1930-1955*. 1957.

[FJM+18] Antonio Fabregat, Steven Jupe, Lisa Matthews, Konstantinos Sidiropoulos, Marc Gillespie, Phani Garapati, Robin Haw, Bijay Jassal, Florian Korninger, Bruce May, Marija Milacic, Corina Duenas Roca, Karen Rothfels, Cristoffer Sevilla, Veronica Shamovsky, Solomon Shorser, Thawfeek Varusai, Guilherme Viteri, Joel Weiser, Guanming Wu, Lincoln Stein, Henning Hermjakob, and Peter D'Eustachio. The Reactome Pathway Knowledgebase. *Nucleic Acids Research*, 48(D1):D498–D503, 2018.

[FKG+11] Guido Ferrari, Bette Korber, Nilu Goonetilleke, Michael K.P. Liu, Emma L. Turnbull, Jesus F. Salazar-Gonzalez, Natalie Hawkins, Steve Self, Sydeaka Watson, Michael R. Betts, Cynthia Gay, Kara McGhee, Pierre Pellegrino, Ian Williams, Georgia D. Tomaras, Barton F. Haynes, Clive M. Gray, Persephone Borrow, Mario Roederer, Andrew J. McMichael, and Kent J. Weinhold. Relationship between functional profile of HIV-1 specific CD8 T cells and epitope variability with the selection of escape mutants in acute HIV-1 infection. *PLoS Pathogens*, 7(2):e1001273, 2011.

[FPT+07] Will Fischer, Simon Perkins, James Theiler, Tanmoy Bhattacharya, Karina Yusim, Robert Funkhouser, Carla Kuiken, Barton Haynes, Norman L. Letvin, Bruce D. Walker, Beatrice H. Hahn, and Bette T. Korber. Polyvalent vaccines for optimal coverage of potential T-cell epitopes in global HIV-1 variants. *Nature Medicine*, 13(1):100–106, 2007.

[FSJB+06] Pablo Fernandez, Brigitte Saint-Joanis, Nathalie Barilone, Mary Jackson, Brigitte Gicquel, Stewart T. Cole, and Pedro M. Alzari. The Ser/Thr Protein Kinase PknB Is Essential for Sustaining Mycobacterial Growth. *Journal of Bacteriology*, 188(22):7778–7784, 2006.

[GBB11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 15:315–323, 2011.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[Gen19] 10X Genomics. A New Way of Exploring Immunity – Linking Highly Multiplexed Antigen Recognition to Immune Repertoire and Phenotype, 2019. Link to document.

[GMB+16] Dominic Grün, Mauro J. Muraro, Jean Charles Boisset, Kay Wiebrands, Anna Lyubimova, Gitanjali Dharmadhikari, Maaike van den Born, Johan van Es, Erik Jansen, Hans Clevers, Eelco J.P. de Koning, and

Alexander van Oudenaarden. De Novo Prediction of Stem Cell Identity using Single-Cell Transcriptome Data. *Cell Stem Cell*, 19(2):266–277, 2016.

[GPSW17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *Proceedings of the 34th International Conference on Machine Learning*, pages 1321–1330, 2017.

[GSG+20] Allison J. Greaney, Tyler N. Starr, Pavlo Gilchuk, Seth J. Zost, Elad Binshtein, Andrea N. Loes, Sarah K. Hilton, John Huddleston, Rachel Eguia, Katharine H.D. Crawford, Adam S. Dingens, Rachel S. Nargi, Rachel E. Sutton, Naveenchandra Suryadevara, Paul W. Rothlauf, Zhuoming Liu, Sean P.J. Whelan, Robert H. Carnahan, James E. Crowe, and Jesse D. Bloom. Complete Mapping of Mutations to the SARS-CoV-2 Spike Receptor-Binding Domain that Escape Antibody Recognition. *Cell Host & Microbe*, 2020.

[GWH14] Robert C. Grande, Thomas J. Walsh, and Jonathan P. How. Sample efficient reinforcement learning with Gaussian processes. *Proceedings of the 31st International Conference on Machine Learning*, pages 1332–1340, 2014.

[GWH+17] Todd M. Gierahn, Marc H. Wadsworth, Travis K. Hughes, Bryan D. Bryson, Andrew Butler, Rahul Satija, Sarah Fortune, J. Christopher Love, and Alex K. Shalek. Seq-Well: Portable, low-cost RNA sequencing of single cells at high throughput. *Nature Methods*, 14(4):395–398, 2017.

[Ham50] R. W. Hamming. Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 29(2):147–160, 1950.

[Har54] Zellig S. Harris. Distributional Structure. *WORD*, 10(2-3):146–162, 1954.

[Hau37] Felix Hausdorff. *Set Theory*. 1937.

[HBB19] Brian Hie, Bryan Bryson, and Bonnie Berger. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. *Nature Biotechnology*, 37(6):685–691, 2019.

[HBB20] Brian Hie, Bryan D. Bryson, and Bonnie Berger. Leveraging Uncertainty in Machine Learning Accelerates Biological Discovery and Design. *Cell Systems*, 11(5):461–477, 2020.

[HCB18] Brian Hie, Hyunghoon Cho, and Bonnie Berger. Realizing private and practical pharmacological collaboration. *Science*, 362(6412):347–350, 2018.

[HCD+19] Brian Hie, Hyunghoon Cho, Benjamin DeMeo, Bryan Bryson, and Bonnie Berger. Geometric Sketching Compactly Summarizes the Single-Cell Transcriptomic Landscape. *Cell Systems*, 8(6):483–493.E7, 2019.

[HDH+18] Hugh K. Haddox, Adam S. Dingens, Sarah K. Hilton, Julie Overbaugh, and Jesse D. Bloom. Mapping mutational effects along the evolutionary landscape of HIV envelope. *eLife*, 7:e34420, 2018.

[HGS+19] Thomas A. Hopf, Anna G. Green, Benjamin Schubert, Sophia Mersmann, Charlotta P.I. Schärfe, John B. Ingraham, Agnes Toth-Petroczy, Kelly Brock, Adam J. Riesselman, Perry Palmedo, Chan Kang, Robert Sheridan, Eli J. Draizen, Christian Dallago, Chris Sander, and Debora S. Marks. The EVcouplings Python framework for coevolutionary sequence analysis. *Bioinformatics*, 35(9):1582–1584, 2019.

[HIP+17] Thomas A. Hopf, John B. Ingraham, Frank J. Poelwijk, Charlotta P.I. Schärfe, Michael Springer, Chris Sander, and Debora S. Marks. Mutation effects predicted from sequence co-variation. *Nature Biotechnology*, 35(2):128–135, 2017.

[HKR93] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J. Rucklidge. Comparing Images Using the Hausdorff Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

[HLMM18] Laleh Haghverdi, Aaron Lun, Michael Morgan, and John Marioni. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nature Biotechnology*, 36(5):421–427, 2018.

[HLN+08] Christian Hoffmann, Andrew Leis, Michael Niederweis, Jürgen M. Plitzko, and Harald Engelhardt. Disclosure of the mycobacterial outer membrane: Cryo-electron tomography and vitreous sections reveal the lipid bilayer structure. *Proceedings of the National Academy of Sciences of the United States of America*, 105(10):3963–3967, 2008.

[HMT11] Nathan Halko, Per-Gunnar Martinsson, and Joel Tropp. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Reviews*, 53(2):217–288, 2011.

[Hoe63] Wassily Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[Hot36] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[HPN+20] Brian Hie, Joshua Peters, Sarah K. Nyquist, Alex K. Shalek, Bonnie Berger, and Bryan D. Bryson. Computational Methods for Single-Cell RNA Sequencing. *Annual Review of Biomedical Data Science*, 3(1):339–364, 2020.

[HRA+19] Virginia M. Howick, Andrew J.C. Russell, Tallulah Andrews, Haynes Heaton, Adam J. Reid, Kedar Natarajan, Hellen Butungi, Tom Metcalf, Lisa H. Verzier, Julian C. Rayner, Matthew Berriman, Jeremy K. Herren, Oliver Billker, Martin Hemberg, Arthur M. Talman, and Mara K.N. Lawniczak. The malaria cell atlas: Single parasite transcriptomes across the complete Plasmodium life cycle. *Science*, 365(6455):eaaw2619, 2019.

[HSS+19] Richard Hodson, Neil Savage, Elizabeth Svoboda, Amanda Keener, Anthony King, Brian Owens, Sedeer el Showk, and Liam Drew. Nature outlook: Vaccines. *Nature*, 575:S43–S60, 2019.

[HZBB20] Brian Hie, Ellen Zhong, Bryan Bryson, and Bonnie Berger. Learning mutational semantics. *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2020.

[HZBB21] Brian Hie, Ellen Zhong, Bonnie Berger, and Bryan Bryson. Learning the language of viral evolution and escape. *Science*, 371(6526):284–288, 2021.

[IS05] John J. Irwin and Brian K. Shoichet. ZINC - A free database of commercially available compounds for virtual screening. *Journal of Chemical Information and Modeling*, 45(1):177–182, 2005.

[JBJ18] Wengong Jin, Regina Barzilay, and Tbmmi Jaakkola. Junction tree variational autoencoder for molecular graph generation. *35th International Conference on Machine Learning*, pages 2328–2337, 2018.

[JCP18] Nicole Jackson, Lloyd Czaplewski, and Laura J.V. Piddock. Discovery and development of new antibacterial drugs: Learning from experience? *Journal of Antimicrobial Chemotherapy*, 73(6):1452–1459, 2018.

[JLZ+20] Mingjian Jiang, Zhen Li, Shugang Zhang, Shuang Wang, Xiaofeng Wang, Qing Yuan, and Zhiqiang Wei. Drug-target affinity prediction using graph neural network and contact maps. *RSC Advances*, 10(35):20701–20712, 2020.

[JM61] François Jacob and Jacques Monod. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, 3(3):318–356, 1961.

[KBB+13] Björn F. Koel, David F. Burke, Theo M. Bestebroer, Stefan Van Der Vliet, Gerben C.M. Zondag, Gaby Vervaet, Eugene Skepner, Nicola S. Lewis, Monique I.J. Spronken, Colin A. Russell, Mikhail Y. Eropkin,

Aeron C. Hurt, Ian G. Barr, Jan C. De Jong, Guus F. Rimmelzwaan, Albert D.M.E. Osterhaus, Ron A.M. Fouchier, and Derek J. Smith. Substitutions near the receptor binding site determine major antigenic change during influenza virus evolution. *Science*, 342(6161):976–979, 2013.

[KBV09] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[KCC+16] Nicole L. Kallewaard, Davide Corti, Patrick J. Collins, Ursula Neu, Josephine M. McAuliffe, Ebony Benjamin, Leslie Wachter-Rosati, Frances J. Palmer-Hill, Andy Q. Yuan, Philip A. Walker, Matthias K. Vorlaender, Siro Bianchi, Barbara Guarino, Anna De Marco, Fabrizia Vanzetta, Gloria Agatic, Mathilde Foglierini, Debora Pinna, Blanca Fernandez-Rodriguez, Alexander Fruehwirth, Chiara Silacci, Roksana W. Ogrodowicz, Stephen R. Martin, Federica Sallusto, Jo Ann A. Suzich, Antonio Lanzavecchia, Qing Zhu, Steven J. Gamblin, and John J. Skehel. Structure and Function Analysis of an Antibody Recognizing All Influenza A Subtypes. *Cell*, 166(3):596–608, 2016.

[KLJ+11] Craig Knox, Vivian Law, Timothy Jewison, Philip Liu, Son Ly, Alex Frolkis, Allison Pon, Kelly Banco, Christine Mak, Vanessa Neveu, Yannick Djoumbou, Roman Eisner, An Chi Guo, and David S. Wishart. DrugBank 3.0: A comprehensive resource for 'Omics' research on drugs. *Nucleic Acids Research*, 39(D1):D1035–D1041, 2011.

[KLR+15] Adam J. Kucharski, Justin Lessler, Jonathan M. Read, Huachen Zhu, Chao Qiang Jiang, Yi Guan, Derek A.T. Cummings, and Steven Riley. Estimating the Life Course of Influenza A(H3N2) Antibody Responses from Cross-Sectional Data. *PLoS Biology*, 13(3):e1002082, 2015.

[KMF+19] Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-Ru Loh, and Soumya Raychaudhuri. Fast, sensitive, and accurate integration of single cell data with Harmony. *Nature Methods*, 16(12):1289–1296, 2019.

[KMT+05] Wael Khazen, Jean Pierre M'Bika, Céline Tomkiewicz, Chantal Benelli, Charles Chany, Ammar Achour, and Claude Forest. Expression of macrophage-selective markers in human and rodent adipocytes. *FEBS Letters*, 579(25):5631–5634, 2005.

[Knu97] Donald E. Knuth. *The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms*. Addison Wesley Longman Publishing Co., Inc., USA, 1997.

[Kra19] Florian Krammer. The human antibody response to influenza A virus infection and vaccination. *Nature Reviews Immunology*, 19(6):383–397, 2019.

[KS13] Kazutaka Katoh and Daron M. Standley. MAFFT multiple sequence alignment software version 7: Improvements in performance and usability. *Molecular Biology and Evolution*, 30(4):772–780, 2013.

[KWW18] Hyunsuh Kim, Robert G. Webster, and Richard J. Webby. Influenza Virus: Dealing with a Drifting and Shifting Pathogen. *Viral Immunology*, 31(2):174–183, 2018.

[LBB18] Jonathan W. Lehmann, Daniel J. Blair, and Martin D. Burke. Towards the generalized iterative synthesis of small molecules. *Nature Reviews Chemistry*, 2(2):1–20, 2018.

[LBC+20] MD Luecken, M Büttner, K Chaichoompu, A Danese, M Interlandi, MF Mueller, DC Strobl, L Zappia, M Dugas, M Colomé-Tatché, and FJ Theis. Benchmarking atlas-level data integration in single-cell genomics. *bioRxiv*, 2020.

[LCS+02] Albert T. Liao, May B. Chien, Narmada Shenoy, Dirk B. Mendel, Gerald McMahon, Julie M. Cherrington, and Cheryl A. London. Inhibition of constitutively active forms of mutant kit by multitargeted indolinone tyrosine kinase inhibitors. *Blood*, 100(2):585–593, 2002.

[LEZ+19] Juhye M. Lee, Rachel Eguia, Seth J. Zost, Saket Choudhary, Patrick C. Wilson, Trevor Bedford, Terry Stevens-Ayers, Michael Boeckh, Aeron C. Hurt, Seema S. Lakdawala, Scott E. Hensley, and Jesse D. Bloom. Mapping person-to-person variation in viral mutations that escape polyclonal serum targeting influenza hemagglutinin. *eLife*, 27(8):e49324, 2019.

[LGB+17] Nathan Lawlor, Joshy George, Mohan Bolisetty, Romy Kursawe, Lili Sun, V. Sivakamasundari, Ina Kycia, Paul Robson, and Michael L. Stitzel. Single-cell transcriptomes identify human islet cell signatures and reveal cell-type-specific expression changes in type 2 diabetes. *Genome Research*, 27(2):208–222, 2017.

[LKB+18] Raymond H.Y. Louie, Kevin J. Kaczorowski, John P. Barton, Arup K. Chakraborty, and Matthew R. McKay. Fitness landscape of the human immunodeficiency virus envelope protein that is targeted by antibodies. *Proceedings of the National Academy of Sciences of the United States of America*, 115(4):E564–E573, 2018.

[LMF+03] Lin Leng, Christine N. Metz, Yan Fang, Jing Xu, Seamas Donnelly, John Baugh, Thomas Delohery, Yibang Chen, Robert A. Mitchell, and

Richard Bucala. MIF Signal Transduction Initiated by Binding to CD74. *The Journal of Experimental Medicine*, 197(11):1467–1476, 2003.

[LOS+11] Kathryn E.A. Lougheed, Simon A. Osborne, Barbara Saxty, David Whalley, Tim Chapman, Nathalie Bouloc, Jasveen Chugh, Timothy J. Nott, Dony Patel, Vicky L. Spivey, Catherine A. Kettleborough, Justin S. Bryans, Debra L. Taylor, Stephen J. Smerdon, and Roger S. Buxton. Effective inhibitors of the essential kinase PknB and their potential as anti-mycobacterial agents. *Tuberculosis*, 91(4):277–286, 2011.

[LOSC20] Haitao Liu, Yew Soon Ong, Xiaobo Shen, and Jianfei Cai. When Gaussian Process Meets Big Data: A Review of Scalable GPs. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6402–6413, 2017.

[LZZ+17] Yunan Luo, Xinbin Zhao, Jingtian Zhou, Jinglin Yang, Yanqing Zhang, Wenhua Kuang, Jian Peng, Ligong Chen, and Jianyang Zeng. A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nature Communications*, 8(1):573, 2017.

[MDG+16] Mauro J. Muraro, Gitanjali Dharmadhikari, Dominic Grün, Nathalie Groen, Tim Dielen, Erik Jansen, Leon van Gurp, Marten A. Engelse, Francoise Carlotti, Eelco J.P. de Koning, and Alexander van Oudenaarden. A Single-Cell Transcriptome Atlas of the Human Pancreas. *Cell Systems*, 3(4):385–394, 2016.

[MH18] Leland McInnes and John Healy. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv*, stat.ML:1802.03426, 2018.

[MKY+13] Jian Ping Mei, Chee Keong Kwoh, Peng Yang, Xiao Li Li, and Jie Zheng. Drug-target interaction prediction by learning from local information and neighbors. *Bioinformatics*, 29(2):238–245, 2013.

[MLB06] Eric F. Morand, Michelle Leech, and Jürgen Bernhagen. MIF: A new cytokine link between rheumatoid arthritis and atherosclerosis. *Nature Reviews Drug Discovery*, 5(5):399–411, 2006.

[MMWC12] Anand Murugan, Thierry Mora, Aleksandra M. Walczak, and Curtis G. Callan. Statistical inference of the generation probability of T-cell receptors from sequence repertoires. *Proceedings of the National Academy of Sciences of the United States of America*, 109(40):16161–16166, 2012.

[MOO+11] Shingo Matsushima, Naoyuki Okita, Misako Oku, Wataru Nagai, Masaki Kobayashi, and Yoshikazu Higami. An Mdm2 antagonist, Nutlin-3a, induces p53-dependent and proteasome-mediated poly(ADP-ribose) polymerase1 degradation in mouse fibroblasts. *Biochemical and Biophysical Research Communications*, 407(3):557–561, 2011.

[MRDJ17] Jonas Mueller, David N. Reshef, George Du, and Tommi Jaakkola. Learning optimal interventions. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1039–1047, 2017.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pages 3111–3119, 2013.

[MW16] Kenneth Murphy and Casey Weaver. *Janeway's Immunobiology*. W. W. Norton & Company, 2016.

[MYD+11] Daphna Meroz, Sun Woo Yoon, Mariette F. Ducatez, Thomas P. Fabrizio, Richard J. Webby, Tomer Hertz, and Nir Ben-Tal. Putative amino acid determinants of the emergence of the 2009 influenza A (H1N1) virus in the human population. *Proceedings of the National Academy of Sciences of the United States of America*, 108(33):13522–13527, 2011.

[MZ17] Payman Mohassel and Yupeng Zhang. SecureML: A System for Scalable Privacy-Preserving Machine Learning. *Proceedings - IEEE Symposium on Security and Privacy*, pages 19–38, 2017.

[NBC21] Ashwin Narayan, Bonnie Berger, and Hyunghoon Cho. Assessing single-cell transcriptomic variability through density-preserving data visualization. *Nature Biotechnology*, 2021.

[Nea12] Radford M Neal. *Bayesian learning for neural networks*. Springer Science & Business Media, 2012.

[Ng] Andrew Ng. CS229 Lecture Notes.

[NHP+16] Sonia Nestorowa, Fiona K. Hamey, Blanca Pijuan Sala, Evangelia Diamanti, Mairi Shepherd, Elisa Laurenti, Nicola K. Wilson, David G. Kent, and Berthold Göttgens. A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. *Blood*, 128(8):e20–e31, 2016.

[NIW+13] Valeria Nikolaenko, Stratis Ioannidis, Udi Weinsberg, Marc Joye, Nina Taft, and Dan Boneh. Privacy-preserving matrix factorization. *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications security*, pages 801–812, 2013.

[NO07] Takashi Nishide and Kazuo Ohta. Multiparty Computation for Interval, Equality, and Comparison Without Bit-Decomposition Protocol. *Public Key Cryptography*, pages 343–360, 2007.

[NYC15] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.

[OLA+14] Corrie Ortega, Reiling Liao, Lindsey N. Anderson, Tige Rustad, Anja R. Ollodart, Aaron T. Wright, David R. Sherman, and Christoph Grundner. Mycobacterium tuberculosis Ser/Thr Protein Kinase B Mediates an Oxygen-Dependent Replication Switch. *PLoS Biology*, 12(1):e1001746, 2014.

[Oli07] Travis E Oliphant. SciPy: Open source scientific tools for Python. *Computing in Science and Engineering*, 9:10–20, 2007.

[ÖÖO18] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. DeepDTA: Deep drug-target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.

[OSM+09] Shao En Ong, Monica Schenone, Adam A. Margolin, Xiaoyu Li, Kathy Do, Mary K. Doud, D. R. Mani, Letian Kuai, Xiang Wang, John L. Wood, Nicola J. Tolliday, Angela N. Koehler, Lisa A. Marcaurelle, Todd R. Golub, Robert J. Gould, Stuart L. Schreiber, and Steven A. Carr. Identifying the proteins to which small-molecule probes and drugs bind in cells. *Proceedings of the National Academy of Sciences of the United States of America*, 106(12):4617–4622, 2009.

[PAG+15] Franziska Paul, Ya'Ara Arkin, Amir Giladi, Diego Adhemar Jaitin, Ephraim Kenigsberg, Hadas Keren-Shaul, Deborah Winter, David Lara-Astiaso, Meital Gury, Assaf Weiner, Eyal David, Nadav Cohen, Felicia Kathrine Bratt Lauridsen, Simon Haas, Andreas Schlitzer, Alexander Mildner, Florent Ginhoux, Steffen Jung, Andreas Trumpp, Bo Torben Porse, Amos Tanay, and Ido Amit. Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors. *Cell*, 163(7):1663–1677, 2015.

[PBM16] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 78(5):947–1012, 2016.

[PNI+18] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. *Proceedings of NAACL-HLT*, pages 2227–2237, 2018.

[Pop59] Karl Popper. *The Logic of Scientific Discovery*. Routledge Classics, 1959.

[PPY+19] Krzysztof Polański, Jong-Eun Park, Matthew D Young, Zhichao Miao, Kerstin B Meyer, and Sarah A Teichmann. BBKNN: Fast Batch Alignment of Single Cell Transcriptomes. *Bioinformatics*, page btz625, 2019.

[PSGG+19] Blanca Pijuan-Sala, Jonathan A. Griffiths, Carolina Guibentif, Tom W. Hiscock, Wajid Jawaid, Fernando J. Calero-Nieto, Carla Mulas, Ximena Ibarra-Soria, Richard C.V. Tyser, Debbie Lee Lian Ho, Wolf Reik, Shankar Srinivas, Benjamin D. Simons, Jennifer Nichols, John C. Marioni, and Berthold Göttgens. A single-cell molecular map of mouse gastrulation and early organogenesis. *Nature*, 566(7745):490–495, 2019.

[PV11] Fabian Pedregosa and G Varoquaux. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[QMM20] Xin Qiu, Elliot Meyerson, and Risto Miikkulainen. Quantifying Point-Prediction Uncertainty in Neural Networks via Residual Estimation with an I/O Kernel. *Eighth International Conference on Learning Representations*, arXiv(cs.LG):1906.00588, 2020.

[RA17] Mostafa Rahmani and George K. Atia. Spatial Random Sampling: A Structure-Preserving Data Sketching Tool. *IEEE Signal Processing Letters*, 24(9):1398–1402, 2017.

[Ram12] Sephra N. Rampersad. Multiple applications of alamar blue as an indicator of metabolic function and cellular health in cell viability bioassays. *Sensors*, 12(9):12347–12360, 2012.

[Ran71] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[RBT+19] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating Protein Transfer Learning with TAPE. *Proceedings of the 33nd International Conference on Neural Information Processing Systems*, pages 9686–9698, 2019.

[Rea14] Sarah Reardon. Pharma firms join NIH on drug development. *Nature*, 2014.

[RH10] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

[RKK01] Michael J. Root, Michael S. Kay, and Peter S. Kim. Protein design of an HIV-1 entry inhibitor. *Science*, 291(5505):884–888, 2001.

[Rou87] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[RSG+19] Samuel G Rodriques, Robert R Stickels, Aleksandrina Goeva, Carly A Martin, Evan Murray, Charles R Vanderburg, Joshua Welch, Linlin M Chen, Fei Chen, and Evan Z Macosko. Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Science*, 363(6434):1463–1467, 2019.

[RTL+17] Aviv Regev, Sarah A. Teichmann, Eric S. Lander, Ido Amit, Christophe Benoist, Ewan Birney, Bernd Bodenmiller, Peter Campbell, Piero Carninci, Menna Clatworthy, Hans Clevers, Bart Deplancke, Ian Dunham, James Eberwine, Roland Eils, Wolfgang Enard, Andrew Farmer, Lars Fugger, Berthold Göttgens, Nir Hacohen, Muzlifah Haniffa, Martin Hemberg, Seung Kim, Paul Klenerman, Arnold Kriegstein, Ed Lein, Sten Linnarsson, Emma Lundberg, Joakim Lundeberg, Partha Majumder, John C. Marioni, Miriam Merad, Musa Mhlanga, Martijn Nawijn, Mihai Netea, Garry Nolan, Dana Pe'er, Anthony Phillipakis, Chris P. Ponting, Stephen Quake, Wolf Reik, Orit Rozenblatt-Rosen, Joshua Sanes, Rahul Satija, Ton N. Schumacher, Alex Shalek, Ehud Shapiro, Padmanee Sharma, Jay W. Shin, Oliver Stegle, Michael Stratton, Michael J.T. Stubbington, Fabian J. Theis, Matthias Uhlen, Alexander Van Oudenaarden, Allon Wagner, Fiona Watt, Jonathan Weissman, Barbara Wold, Ramnik Xavier, and Nir Yosef. The Human Cell Atlas. *eLife*, 6:e27041, 2017.

[RUC+10] Jane B. Reece, Lisa A. Urry, Michael L. Cain, Steven A. Wasserman, Peter V. Minorsky, and Robert B. Jackson. *Campbell Biology*. Pearson, 2010.

[RW05] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2005.

[RWC+19a] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[RWC+19b] Philipp Rentzsch, Daniela Witten, Gregory M. Cooper, Jay Shendure, and Martin Kircher. CADD: Predicting the deleteriousness of variants throughout the human genome. *Nucleic Acids Research*, 47(D1):D886–D894, 2019.

[RWLP03] Douglas D. Richman, Terri Wrin, Susan J. Little, and Christos J. Petropoulos. Rapid evolution of the neutralizing antibody response to HIV type 1 infection. *Proceedings of the National Academy of Sciences of the United States of America*, 100(7):4144–4149, 2003.

[SBZ+18] Mikhail Shugay, Dmitriy V. Bagaev, Ivan V. Zvyagin, Renske M. Vroomans, Jeremy Chase Crawford, Garry Dolton, Ekaterina A. Komech, Anastasiya L. Sycheva, Anna E. Koneva, Evgeniy S. Egorov, Alexey V. Eliseev, Ewald Van Dyk, Pradyot Dash, Meriem Attaf, Cristina Rius, Kristin Ladell, James E. McLaren, Katherine K. Matthews, E. Bridie Clemens, Daniel C. Douek, Fabio Luciani, Debbie Van Baarle, Katherine Kedzierska, Can Kesmir, Paul G. Thomas, David A. Price, Andrew K. Sewell, and Dmitriy M. Chudakov. VDJdb: A curated database of T-cell receptor sequences with known antigen specificity. *Nucleic Acids Research*, 46(D1):D419–D427, 2018.

[SC17] Yuriy Sverchkov and Mark Craven. A review of active learning approaches to experimental design for uncovering biological networks. *PLoS Computational Biology*, 13(6):e1005466, 2017.

[SC20] Enrique Santiago and Armando Caballero. The value of targeting recombination as a strategy against coronavirus diseases. *Heredity*, 125:69–172, 2020.

[SG08] Ajit P. Singh and Geoffrey J. Gordon. Relational learning via collective matrix factorization. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 650–658, 2008.

[SGH+20] Tyler N. Starr, Allison J. Greaney, Sarah K. Hilton, Daniel Ellis, Katharine H.D. Crawford, Adam S. Dingens, Mary Jane Navarro, John E. Bowen, M. Alejandra Tortorici, Alexandra C. Walls, Neil P. King, David Veesler, and Jesse D. Bloom. Deep Mutational Scanning of SARS-CoV-2 Receptor Binding Domain Reveals Constraints on Folding and ACE2 Binding. *Cell*, 182(5):1295–1310.e20, 2020.

[SGP+18] Laksshman Sundaram, Hong Gao, Samskruthi Reddy Padigepati, Jeremy F. McRae, Yanjun Li, Jack A. Kosmicki, Nondas Fritzilas, Jörg Hakenberg, Anindita Dutta, John Shon, Jinbo Xu, Serafim Batzloglou, Xiaolin Li, and Kyle Kai How Farh. Predicting the clinical impact of human mutation with deep neural networks. *Nature Genetics*, 50(8):1161–1170, 2018.

[SHM+16] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[SHNB20] Rohit Singh, Brian Hie, Ashwin Narayan, and Bonnie Berger. Metric learning enables synthesis of heterogeneous single-cell modalities. *bioRxiv*, 2020.

[SHS+17] Marlon Stoeckius, Christoph Hafemeister, William Stephenson, Brian Houck-Loomis, Pratip K. Chattopadhyay, Harold Swerdlow, Rahul Satija, and Peter Smibert. Simultaneous epitope and transcriptome measurement in single cells. *Nature Methods*, 14(9):865–868, 2017.

[SLZ+10] Sampa Santra, Hua Xin Liao, Ruijin Zhang, Mark Muldoon, Sydeaka Watson, Will Fischer, James Theiler, James Szinger, Harikrishnan Balachandran, Adam Buzby, David Quinn, Robert J. Parks, Chun Yen Tsao, Angela Carville, Keith G. Mansfield, George N. Pavlakis, Barbara K. Felber, Barton F. Haynes, Bette T. Korber, and Norman L. Letvin. Mosaic vaccines elicit CD8+ T lymphocyte responses that confer enhanced immune coverage of diverse HIV strains in monkeys. *Nature Medicine*, 16(3):324–328, 2010.

[SMW+18] Arpiar Saunders, Evan Z. Macosko, Alec Wysoker, Melissa Goldman, Fenna M. Krienen, Heather de Rivera, Elizabeth Bien, Matthew Baum, Laura Bortolin, Shuyu Wang, Aleksandrina Goeva, James Nemesh, Nolan Kamitaki, Sara Brumbaugh, David Kulp, and Steven A. McCarroll. Molecular Diversity and Specializations among the Cells of the Adult Mouse Brain. *Cell*, 174(4):1015–1030.e16, 2018.

[SNC+10] Rafael Sanjuán, Miguel R. Nebot, Nicola Chirico, Louis M. Mansky, and Robert Belshaw. Viral Mutation Rates. *Journal of Virology*, 84(19):9733–9748, 2010.

[SOAC+18] Genevieve L. Stein-O'Brien, Raman Arora, Aedin C. Culhane, Alexander V. Favorov, Lana X. Garmire, Casey S. Greene, Loyal A. Goff, Yifeng Li, Aloune Ngom, Michael F. Ochs, Yanxun Xu, and Elana J. Fertig. Enter the Matrix: Factorization Uncovers Knowledge from Omics. *Trends in Genetics*, 34(10):790–805, 2018.

[SPE+16] Åsa Segerstolpe, Athanasia Palasantza, Pernilla Eliasson, Eva-Marie Andersson, Anne-Christine Andréasson, Xiaoyan Sun, Simone Picelli, Alan Sabirsh, Maryam Clausen, Magnus K. Bjursell, David M. Smith, Maria Kasper, Carina Ämmälä, and Rickard Sandberg. Single-Cell Transcriptome Profiling of Human Pancreatic Islets in Health and Type 2 Diabetes. *Cell Metabolism*, 24(4):593–607, 2016.

[SS19] Tim Stuart and Rahul Satija. Integrative single-cell analysis. *Nature Reviews Genetics*, 20(5):257–272, 2019.

[SSV+16] Damian Szklarczyk, Alberto Santos, Christian Von Mering, Lars Juhl Jensen, Peer Bork, and Michael Kuhn. STITCH 5: Augmenting protein-

chemical interaction networks with tissue and affinity data. *Nucleic Acids Research*, 44(D1):D380–D384, 2016.

[STD+13] Weijun Shen, Matthew S. Tremblay, Vishal A. Deshmukh, Weidong Wang, Christophe M. Filippi, George Harb, You Qing Zhang, Anwesh Kamireddy, Janine E. Baaten, Qihui Jin, Tom Wu, Jonathan G. Swoboda, Charles Y. Cho, Jing Li, Bryan A. Laffitte, Peter McNamara, Richard Glynne, Xu Wu, Ann E. Herman, and Peter G. Schultz. Small-molecule inducer of β cell proliferation identified by high-throughput screening. *Journal of the American Chemical Society*, 135(5):1669–1672, 2013.

[STS18] Valentine Svensson, Sarah A. Teichmann, and Oliver Stegle. SpatialDE: Identification of spatially variable genes. *Nature Methods*, 15(5):343–346, 2018.

[SWLO06] Manish Sagar, Xueling Wu, Sandra Lee, and Julie Overbaugh. Human Immunodeficiency Virus Type 1 V1-V2 Envelope Loop Sequences Expand and Add Glycosylation Sites over the Course of Infection, and These Modifications Affect Antibody Neutralization Sensitivity. *Journal of Virology*, 80(19):9586–9598, 2006.

[SYS+20] Jonathan M. Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M. Donghia, Craig R. MacNair, Shawn French, Lindsey A. Carfrae, Zohar Bloom-Ackerman, Victoria M. Tran, Anush Chiappino-Pepe, Ahmed H. Badran, Ian W. Andrews, Emma J. Chory, George M. Church, Eric D. Brown, Tommi S. Jaakkola, Regina Barzilay, and James J. Collins. A Deep Learning Approach to Antibiotic Discovery. *Cell*, 180(4):688–702, 2020.

[TCG+14] Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J. Lennon, Kenneth J. Livak, Tarjei S. Mikkelsen, and John L. Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology*, 32(4):381–386, 2014.

[TCwC+07] Adi L. Tarca, Vincent J. Carey, Xue wen Chen, Roberto Romero, and Sorin Drăghici. Machine learning and its applications to biology. *PLoS Computational Biology*, 3(6):e116, 2007.

[THI+20] Kelvin Kai-Wang To, Ivan Fan-Ngai Hung, Jonathan Daniel Ip, Allen Wing-Ho Chu, Wan-Mui Chan, Anthony Raymond Tam, Carol Ho-Yan Fong, Shuofeng Yuan, Hoi-Wah Tsoi, Anthony Chin-Ki Ng, Larry Lap-Yip Lee, Polk Wan, Eugene Tso, Wing-Kin To, Dominic Tsang, Kwok-Hung Chan, Jian-Dong Huang, Kin-Hang Kok, Vincent Chi-Chung Cheng, and Kwok-Yung Yuen. COVID-19 re-infection by a phylogenetically distinct SARS-coronavirus-2 strain confirmed by whole genome sequencing. *Clinical Infectious Diseases*, page ciaa1275, 2020.

[TKD+16] Dustin Tran, Alp Kucukelbir, Adji B. Dieng, Maja Rudolph, Dawen Liang, and David M. Blei. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv*, stat.CO:1610.09787, 2016.

[TM17] Ellen Tedford and Glenn McConkey. Neurophysiological changes induced by chronic Toxoplasma gondii infection. *Pathogens*, 6(2):19, 2017.

[TVRWM04] Giuseppe Tocchini-Valentini, Natacha Rochel, Jean Marie Wurtz, and Dino Moras. Crystal Structures of the Vitamin D Nuclear Receptor Liganded with the Vitamin D Side Chain Analogues Calcipotriol and Seocalcitol, Receptor Agonists of Clinical Importance. Insights into a Structural Basis for the Switching of Calcipotriol to a Receptor. *Journal of Medicinal Chemistry*, 47(8):1956–1961, 2004.

[TWvE19] V. A. Traag, L. Waltman, and N. J. van Eck. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1):1–12, 2019.

[VBC+19] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[vdMH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[VEB10] Nx Vinh, J Epps, and J Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:2837–2854, 2010.

[WBB+06] Rudolf Waelchli, Birgit Bollbuck, Christian Bruns, Thomas Buhl, Jörg Eder, Roland Feifel, Rene Hersperger, Philipp Janser, Laszlo Revesz, Hans Günter Zerwes, and Achim Schlapbach. Design and preparation of 2-benzamido-pyrimidines as inhibitors of IKK. *Bioorganic and Medicinal Chemistry Letters*, 16(1):108–112, 2006.

[WBD+10] Chih Jen Wei, Jeffrey C. Boyington, Kaifan Dai, Katherine V. Houser, Melissa B. Pearce, Wing Pui Kong, Zhi Yong Yang, Terrence M. Tumpey,

and Gary J. Nabel. Cross-neutralization of 1918 and 2009 influenza viruses: Role of glycans in viral evolution and vaccine design. *Science Translational Medicine*, 2(24):24ra21, 2010.

[WC53] James D. Watson and Francis H.C. Crick. Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, 171:737–738, 1953.

[WDW+03] Xiping Wei, Julie M. Decker, Shuyi Wang, Huxiong Hui, John C. Kappes, Xiaoyun Wu, Jesus F. Salazar-Gonzalez, Maria G. Salazar, J. Michael Kilby, Michael S. Saag, Natalia L. Komarova, Martin A. Nowak, Beatrice H. Hahn, Peter D. Kwong, and George M. Shaw. Antibody neutralization and escape by HIV-1. *Nature*, 422(6929):307–312, 2003.

[WE07] L. F. Wang and B. T. Eaton. Bats, civets and the emergence of SARS. *Current Topics in Microbiology and Immunology*, 315:325–344, 2007.

[Wei88] David Weininger. SMILES, a Chemical Language and Information System: 1: Introduction to Methodology and Encoding Rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.

[WID09] Deric L. Wheeler, Mari Iida, and Emily F. Dunn. The Role of Src in Solid Tumors. *The Oncologist*, 14(7):667–678, 2009.

[Wil17] Menaka Wilhelm. Big Pharma Buys Into Crowdsourcing for Drug Discovery. *Wired*, 2017.

[Wor19] World Health Organization. *World Malaria Report 2019*. 2019.

[WOT+20] Nicholas C. Wu, Jakub Otwinowski, Andrew J. Thompson, Corwin M. Nycholat, Armita Nourmohammad, and Ian A. Wilson. Major antigenic site B of human influenza H3N2 viruses has an evolving local fitness landscape. *Nature Communications*, 11(1):1–10, 2020.

[WPT+20] Alexandra C. Walls, Young Jun Park, M. Alejandra Tortorici, Abigail Wall, Andrew T. McGuire, and David Veesler. Structure, Function, and Antigenicity of the SARS-CoV-2 Spike Glycoprotein. *Cell*, 181(2):281–292.e6, 2020.

[WWS+09] Zhulun Wang, Holger Wesche, Tracey Stevens, Nigel Walker, and Wen-Chen Yeh. IRAK-4 Inhibitors for Inflammation. *Current Topics in Medicinal Chemistry*, 8(9):724–737, 2009.

[WYZL14] Wenhui Wang, Sen Yang, Xiang Zhang, and Jing Li. Drug repositioning by integrating target information through a heterogeneous network model. *Bioinformatics*, 30(20):2923–2930, 2014.

[XEK+10] Rui Xu, Damian C. Ekiert, Jens C. Krause, Rong Hai, James E. Crowe, and Ian A. Wilson. Structural basis of preexisting immunity to the 2009 H1N1 pandemic influenza virus. *Science*, 328(5976):357–360, 2010.

[XTR+20] Yuan Xue, Terence C. Theisen, Suchita Rastogi, Abel Ferrel, Stephen R. Quake, and John C. Boothroyd. A single-parasite transcriptional atlas of toxoplasma gondii reveals novel control of antigen expression. *eLife*, 9:e54129, 2020.

[XWZW10] Zheng Xia, Ling-Yun Wu, Xiaobo Zhou, and Stephen T C Wong. Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces. *BMC Systems Biology*, 4(S6):1–12, 2010.

[YAA+16] Andrew Yates, Wasiu Akanni, M. Ridwan Amode, Daniel Barrell, Konstantinos Billis, Denise Carvalho-Silva, Carla Cummins, Peter Clapham, Stephen Fitzgerald, Laurent Gil, Carlos García Girón, Leo Gordon, Thibaut Hourlier, Sarah E. Hunt, Sophie H. Janacek, Nathan Johnson, Thomas Juettemann, Stephen Keenan, Ilias Lavidas, Fergal J. Martin, Thomas Maurel, William McLaren, Daniel N. Murphy, Rishi Nag, Michael Nuhn, Anne Parker, Mateus Patricio, Miguel Pignatelli, Matthew Rahtz, Harpreet Singh Riat, Daniel Sheppard, Kieron Taylor, Anja Thormann, Alessandro Vullo, Steven P. Wilder, Amonida Zadissa, Ewan Birney, Jennifer Harrow, Matthieu Muffato, Emily Perry, Magali Ruffier, Giulietta Spudich, Stephen J. Trevanion, Fiona Cunningham, Bronwen L. Aken, Daniel R. Zerbino, and Paul Flicek. Ensembl 2016. *Nucleic Acids Research*, 44(D1):D710–D716, 2016.

[YAG+08] Yoshihiro Yamanishi, Michihiro Araki, Alex Gutteridge, Wataru Honda, and Minoru Kanehisa. Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, 24(13):i232–i240, 2008.

[YDDB15] Y. William Yu, Noah M. Daniels, David Christian Danko, and Bonnie Berger. Entropy-Scaling Search of Massive Biological Data. *Cell Systems*, 1(2):130–140, 2015.

[YLD+20] Rui Yin, Emil Luusua, Jan Dabrowski, Yu Zhang, and Chee Keong Kwoh. Tempel: time-series mutation prediction of influenza A viruses via attention-based recurrent neural networks. *Bioinformatics*, page btaa050, 2020.

[ZDMZ13] Xiaodong Zheng, Hao Ding, Hiroshi Mamitsuka, and Shanfeng Zhu. Collaborative matrix factorization with multiple similarities for predicting drug-target interactions. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1025–1033, 2013.

[ZHL+18] Amit Zeisel, Hannah Hochgerner, Peter Lönnerberg, Anna Johnsson, Fatima Memic, Job van der Zwan, Martin Häring, Emelie Braun, Lars E. Borm, Gioele La Manno, Simone Codeluppi, Alessandro Furlan, Kawai Lee, Nathan Skene, Kenneth D. Harris, Jens Hjerling-Leffler, Ernest Arenas, Patrik Ernfors, Ulrika Marklund, and Sten Linnarsson. Molecular Architecture of the Mouse Nervous System. *Cell*, 174(4):999–1014.e22, 2018.

[ZTB+17] Grace X.Y. Zheng, Jessica M. Terry, Phillip Belgrader, Paul Ryvkin, Zachary W. Bent, Ryan Wilson, Solongo B. Ziraldo, Tobias D. Wheeler, Geoff P. McDermott, Junjie Zhu, Mark T. Gregory, Joe Shuga, Luz Montesclaros, Jason G. Underwood, Donald A. Masquelier, Stefanie Y. Nishimura, Michael Schnall-Levin, Paul W. Wyatt, Christopher M. Hindson, Rajiv Bharadwaj, Alexander Wong, Kevin D. Ness, Lan W. Beppu, H. Joachim Deeg, Christopher McFarland, Keith R. Loeb, William J. Valente, Nolan G. Ericson, Emily A. Stevens, Jerald P. Radich, Tarjei S. Mikkelsen, Benjamin J. Hindson, and Jason H. Bielas. Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8(1):14049, 2017.