Visual analytics at the atlas scale for multimodal and spatial single-cell data

A DISSERTATION PRESENTED
BY
MARK S. KELLER
TO
THE DIVISION OF MEDICAL SCIENCES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE SUBJECT OF
BIOMEDICAL INFORMATICS

HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS

APRIL 2025

© 2025 – Mark S. Keller
This dissertation is licensed under a Creative Commons Attribution 4.0 International Public License.

Dissertation Advisor: Nils Gehlenborg Mark S. Keller

Visual analytics at the atlas scale for multimodal and spatial single-cell data

Abstract

Scientific measurements at the resolution of individual cells – single-cell experiments – are central to biology because the cell is the fundamental unit of life. In the past decade, advancements in sequencing and bioimaging technologies have enabled cellular measurements to be made at high-throughput, forging the field of single-cell biology. Single-cell experiments are now being applied in large scale, driven by concerted efforts from funding institutions and consortia worldwide. The resulting datasets are being compiled into single-cell *atlas* resources: collections of cellular-resolution maps intended to summarize and communicate single-cell data through hierarchical and spatial organization and inclusion of multiple biosamples, tissue types, organs, and/or organisms from one or more experimental conditions. Single-cell atlases are intended to facilitate downstream usage in biology and medicine via their establishment as gold-standard sets of measurements that can serve as common points of reference. Current information visualization systems are not equipped to adequately deal with the scale, complexity, and heterogeneity of single-cell atlas data, nor are they tailored to the needs of target user audiences.

This thesis investigates how visual analytics systems can be employed for interactive visualization of multimodal and spatial single-cell datasets, addressing challenges at the scale of individual experiments to whole atlases. The first chapter provides an

overview of the landscape of single-cell data visualizations, including interactive systems. The second chapter builds on preliminary work on a framework for interactive data visualization for single-cell data, including for spatial, imaging, and multimodal data. The third chapter builds upon this framework to develop a system tailored to cross-experiment comparisons, for example between single-cell data from case and control groups, informed through interviews with individuals from its intended audience of biologists, pathologists, and clinicians. The fourth chapter explores how algorithms for identification of spatial domains with relevance to biological function can be integrated with interactive visualizations. The fifth chapter explores how to integrate chromatin accessibility measurements into the existing framework for single-cell data visualization.

By combining and extending concepts from bioinformatics, information visualization, human-computer interaction, and software engineering, this thesis pioneers approaches for exploring, understanding, and communicating foundational single-cell atlas resources.

Contents

Title Page **i**

Copyright **ii**

Abstract **iii**

Table of Contents **v**

Listing of figures **viii**

1 Introduction **xiii**
- 1.1 Preface . . . xiii
- 1.2 Motivation . . . xiv
- 1.3 Thesis statement . . . xv
- 1.4 Visualization Background . . . xv
- 1.5 Biological Background . . . xviii
- 1.6 Biomedical Insights from Single-Cell Techniques . . . xxxi
- 1.7 Single-Cell Atlasing Consortia . . . xxxii
- 1.8 Data Structures for Single-Cell Analysis . . . xxxv
- 1.9 Computational Methods for Single-Cell Data Analysis . . . l
- 1.10 Evaluation of Interactive Systems for Single-Cell Data Visualization . . . lvi
- 1.11 Contributions . . . lx

2 The State of Single-Cell Atlas Data Visualization in the Biological Literature **lxiii**
- 2.1 Preface . . . lxiii
- 2.2 Introduction . . . lxiv
- 2.3 Survey of Static Figures from the Biological Literature . . . lxvii
- 2.4 Analysis of Plotting Toolkits . . . xcvi
- 2.5 Discussion . . . xcix
- 2.6 Research Outlook . . . ciii

2.7 Conclusion . . . civ

3 Interactive visual analytics framework for single-cell data **cvi**
3.1 Preface . . . cvi
3.2 Abstract . . . cvii
3.3 Introduction . . . cviii
3.4 Related Work . . . cviii
3.5 Vitessce: integrative visualization of multimodal and spatially-resolved single-cell data . . . cix
3.6 Discussion . . . cxvii
3.7 Conclusion . . . cxxii

4 Cross-experiment comparisons **cxxiv**
4.1 Preface . . . cxxiv
4.2 Introduction . . . cxxv
4.3 Related Work . . . cxxvi
4.4 Polyphony: an Interactive Transfer Learning Framework for Single-Cell Data Analysis . . . cxxxv
4.5 Compasce: Comparative Visualization for Single-Cell Resolution Tissue Atlas Data . . . cliii
4.6 Discussion . . . clxxxix
4.7 Conclusion . . . cxciii

5 Exploration of interpretable spatial domains **cxcv**
5.1 Preface . . . cxcv
5.2 Introduction . . . cxcvii
5.3 Related Work . . . cxcvii
5.4 Viv: Multiscale Visualization of High-Resolution Multiplexed Bioimaging Data on the Web . . . cci
5.5 Interactive visualization of kidney micro-compartmental segmentations and associated pathomics on whole slide images . . . cciii
5.6 Interactive visualization of spatial query results . . . ccxvi
5.7 Discussion . . . ccxxii
5.8 Conclusion . . . ccxxv

6 Visual analysis of single-cell chromatin accessibility **ccxxvii**
6.1 Preface . . . ccxxvii
6.2 Introduction . . . ccxxviii
6.3 Related Work . . . ccxxx
6.4 Kidney Chromatin Landscape Browser . . . ccxxxi
6.5 Extension of Vitessce for visualization of chromatin accessibility data ccxxxii

6.6 Cistrome Explorer: an interactive visual analysis tool for large-scale epigenomic data . . . ccxxxv
6.7 Discussion . . . ccxxxviii
6.8 Conclusion . . . ccxl

7 Conclusion ccxli
7.1 Thesis summary . . . ccxli
7.2 Research outlook . . . ccxliii

Appendix A Use-Coordination ccxlvi
A.1 Preface . . . ccxlvi
A.2 Abstract . . . ccxlvii
A.3 Introduction . . . ccxlviii
A.4 Design Goals . . . ccxlix
A.5 Related Work . . . ccli
A.6 The Use-Coordination Model and Grammar . . . ccliii
A.7 Software Implementation . . . cclv
A.8 Use Cases . . . cclviii
A.9 Discussion . . . cclx
A.10 Conclusion . . . cclxi

Appendix B Supplementary Information for Use-Coordination cclxii
B.1 Preface . . . cclxii
B.2 JSON representation . . . cclxiii
B.3 Hierarchy of coordination . . . cclxviii
B.4 Usage guidance . . . cclxxiv

Appendix C Bibliometric Analysis of Dimensionality Reduction Algorithm Usage cclxxviii
C.1 Preface . . . cclxxviii
C.2 Background . . . cclxxix
C.3 Bibliometric Analysis Across Subject Areas . . . cclxxx
C.4 Code Availability . . . cclxxxvi
C.5 Supplementary Figures . . . cclxxxvii

Appendix D Zarr: Multidimensional Array Storage Format ccxcix
D.1 Preface . . . ccxcix
D.2 Zarr . . . ccc
D.3 Zarr in R . . . ccc

Appendix E Supplementary Information for Chapter 1 cccii

E.1 Preface cccii
E.2 Other single-cell topics ccciii

Appendix F Supplementary Information for Chapter 2 cccv
F.1 Preface cccv
F.2 Query Strings cccvi
F.3 Paper Filtering and Annotation Processes cccvi
F.4 Code Descriptions cccvii

Appendix G Supplementary Information for Chapter 3 cccxvii
G.1 Preface cccxvii
G.2 Extended Data Figures for Vitessce cccxviii
G.3 Using Vitessce cccxxiv
G.4 Multimodal configuration cccxxvii
G.5 Data organization cccxxviii
G.6 Coordinated multiple views cccxxx
G.7 JavaScript implementation cccxxxi
G.8 Processing of data for use cases cccxxxiii
G.9 Software Architecture cccxxxiv
G.10 Planned designs cccxlvi

Appendix H Supplementary Information for Chapter 4 cccxlvii
H.1 Formative Interview Questions cccxlvii
H.2 Formative Interview Results cccl
H.3 Comparative Analysis Survey ccclv
H.4 Comparative Visualization Implementation Details ccclvii
H.5 Comparative Analysis Metadata Format ccclviii
H.6 Planned designs ccclix

Appendix I Supplementary Information for Chapter 5 ccclxii
I.1 Preface ccclxii
I.2 Methods for Section 5.5 ccclxiii
I.3 Proposal for using OME-NGFF image format to store point-based data ccclxviii

References cdxxxix

Listing of figures

1.1 Tabular representations for single-cell datasets. xxxvii

2.1 Methodology used for literature search, paper identification, and filtering. lxx
2.2 Distribution of cell count per publication year. lxx
2.3 Histogram of codes per subfigure. lxxii
2.4 Top codes among all subfigures. lxxiii
2.5 Top codes across publications. lxxiv
2.6 Reproduced figure. lxxv
2.7 Reproduced figure. lxxvi
2.8 Reproduced figure. lxxvi
2.9 Reproduced figure. lxxvii
2.10 Reproduced figure. lxxvii
2.11 Reproduced figure. lxxviii
2.12 Reproduced figure. lxxviii
2.13 Reproduced figure. lxxix
2.14 Reproduced figure. lxxx
2.15 Reproduced figure. lxxx
2.16 Reproduced figure. lxxxii
2.17 Reproduced figure. lxxxiv
2.18 Reproduced figure. lxxxv
2.19 Reproduced figure. lxxxvi
2.20 Reproduced figure. lxxxvii
2.21 Reproduced figure. lxxxviii
2.22 Reproduced figure. lxxxviii
2.23 Reproduced figure. xc
2.24 Reproduced figure. xc
2.25 Reproduced figure. xci
2.26 Reproduced figure. xci
2.27 Reproduced figure. xciii
2.28 Reproduced figure. xciii
2.29 Reproduced figure. xcv

2.30 Screenshot of survey browser website. xcvi
2.31 Plotting function usage statistics. xcviii
2.32 Plotting function usage by code. xcix

3.1 Vitessce can be used in multiple settings. cx
3.2 Sample use cases for Vitessce. cxvi

4.1 Polyphony interface. cxxxvi
4.2 Background on batch effects. cxxxvii
4.3 Interactive transfer learning steps in Polyphony. cxl
4.4 Comparison view in Polyphony. cxliv
4.5 Earlier design of comparison view in Polyphony. cxlvi
4.6 Marker view in Polyphony. cxlvii
4.7 Survey of HuBMAP consortium members about comparisons of interest. clxiv
4.8 Biomarker selection interface. clxvi
4.9 Application layout. clxvii
4.10 Designs for simultaneous visualization of quantitative and categorical information with dimensionality reduction results. clxxi
4.11 Text label positioning for dimensionality reduction contour plot. clxxii
4.12 Comparative dot plots and violin plots. clxxiii
4.13 Dot plots and violin plots. clxxiii
4.14 Violin plots displaying aggregate expression values. clxxiv
4.15 Volcano plot, case versus control, within cell type. clxxv
4.16 Volcano plot, one cell type versus rest. clxxv
4.17 Gene set enrichment bar plot. clxxvi
4.18 Volcano plot and tabular view. clxxvii
4.19 Cell type composition bar plot. clxxviii
4.20 Treemap plot, colored by cell type. clxxviii
4.21 Treemap plot, colored by sample group. clxxix
4.22 Comparative analysis data processing pipeline. clxxxii
4.23 Comparative analysis data processing results. clxxxiii

5.1 Spatially-resolved analysis methods. cxcix
5.2 Pipeline and data representation for segmentations and pathomic features. ccx
5.3 Screenshot of visualization of segmentations of functional tissue units . ccxiii
5.4 Visualization of Slide-seq dataset in HuBMAP Portal. ccxix
5.5 SpatialQuery and Vitessce in a Jupyter notebook. ccxix
5.6 SpatialQuery implementation diagram. ccxxi

6.1 Kidney Chromatin Landscape Browser. ccxxxiii
6.2 Multiome Visualization in HuBMAP Portal. ccxxxiv
6.3 Cistrome Explorer. ccxxxvii

A.1 Example visual analytics system integrating `use-coordination`. . . . ccxlvii
A.2 Extensions of `use-coordination`. cclvi
A.3 Use cases for `use-coordination`. cclvi

C.1 Rankings based on papers citing dimensionality reduction methods. cclxxxiii
C.2 Counts of papers citing dimensionality reduction methods. cclxxxvi
C.3 Overview of record counts in citation network database. cclxxxvii
C.4 Distribution of number of subject area assignments per paper for papers citing dimensionality reduction methods. cclxxxvii
C.5 Comparison of subject area assignment methods. cclxxxviii
C.6 Counts of papers assigned to sets of one or more subject areas. . . . cclxxxix
C.7 Counts of papers in the Semantic Scholar Academic Graph stratified by subject area. ccxc
C.8 Count of papers citing dimensionality reduction methods, stratified by subject area. ccxci
C.9 Count of papers citing dimensionality reduction methods, stratified by subject area. ccxcii
C.10 Percentage of papers citing dimensionality reduction methods, stratified by subject area. ccxciii
C.11 Percentile rankings of dimensionality reduction methods over time based on citation counts. ccxciv
C.12 Percentile rankings of dimensionality reduction methods over time based on citation counts. ccxcv
C.13 Proportional usage of dimensionality reduction methods over time based on citation counts. ccxcvi

G.1 Visualization of single-molecule fluorescence in situ hybridization (smFISH) data. cccxviii
G.2 Visualization of a 10x Genomics Visium dataset. cccxix
G.3 Visualization of a 3D multimodal mass spectrometry imaging dataset. . cccxx
G.4 Visualization of a CODEX dataset. cccxxi
G.5 Visualization of a CITE-seq dataset. cccxxii
G.6 Visualization of a 10x Genomics Multiome dataset. cccxxiii
G.7 Visualization of a MALDI IMS dataset processed multiple ways. . . . cccxxiv
G.8 Comparison of CSV- and Zarr-based file sizes. cccxxx
G.9 Software architecture. cccxxxiv
G.10 Configuration of multimodal visualization. cccxxxvii
G.11 React component architecture for Vitessce. cccxxxviii
G.12 Design of sidebar for Vitessce. cccxlvi

H.1 Design of spatial representative samples view for Vitessce. ccclx

H.2 Design of comparative spatial representative samples view for Vitessce. ccclx
H.3 Design of spatial small multiples view for Vitessce. cccLxi

I.1 Proposal for representation of point data using OME-NGFF. ccclxx

1

Introduction

1.1 Preface

Portions of this chapter have been adapted from "The State of Single-Cell Atlas Data Visualization in the Biological Literature," which is under review at the time of writing[229].

1.2 Motivation

The cell is the fundamental unit of life, responsible for structure, function, and division in organisms[298,359]. The disruption of cellular processes can lead to dysfunction and disease at the tissue, organ, or organism level. Thus, it is of great importance to disentangle and measure cellular processes to improve our understanding of both basic biology and disease. In recognition, researchers have set out to measure biological processes across millions of cells from many different contexts (*e.g.*, organisms, diseases) and publish them in the form of single-cell *atlas* datasets[148]. Life scientists envision single-cell atlases as reference resources of human cells, including characteristic features (biomarkers), locations, and abundances[359]. More comprehensive than any individual experiment, atlases combine data from multiple experiments to provide more power to generate and test hypotheses[279].

Atlas datasets are emerging rapidly but remain daunting to analyze and interpret[199]. One of the biggest obstacles is the high dimensionality of the data, as each of the millions of cells can have tens of thousands of measured features. Moreover, single-cell data brings challenges that do not necessarily emerge in other high-dimensional settings. Challenges include the presence of associated spatially-resolved and genomic data[295,201], batch effects[279], and concerns about misuse of dimensionality reduction[163,118,109,254,449]. Other problems include the contextualization of atlas information with prior biological literature[374], connection of atlases that adhere to differing organizing principles[139], and alignment of data to differing coordinate frameworks[366,438]. In light of these challenges, the domain requires visualizations that integrate multiple data types (*e.g.*, gene expression matrices, genomic coordinates, and microscopy images) and facilitate tasks such as relation and comparison among them.

1.3 Thesis statement

The development and deployment of visual analytics systems for atlas-scale multimodal and spatial single-cell data requires experiment-tailored visualization configurations, coordinated multiple views, efficient and standardized data representations, interpretable data transformation techniques, and interactive visualizations to facilitate comparison, all of which are informed through systematic review of single-cell biology literature and qualitative analysis of need-finding interviews with biologists, pathologists, and clinicians.

1.4 Visualization Background

Data visualization serves as a tool for analysis, communication, and recording of information[44]. Visualization for analysis of information facilitates exploration and supports reasoning. Such exploratory visualization enables hypothesis generation and identification of patterns or outliers. In contrast, the purpose of visualization for communication is to explain, narrate, persuade, or collaborate. Examples of visualizations for recording information are blueprints and seismographs, whereby the recording of information does not necessarily happen prior to visualization, but instead the processes are one and the same. Visualization leverages the capacity of the human visual system and cognition to surface insights and disseminate information – an important category of information being scientific findings.

Data visualizations can be static or interactive. Thus, there are two key components to the scientific discipline of visualization: representation and interaction[457]. The study of representation concerns human perception and computer graphics. On the other hand, the study of interaction for visualization is a topic within the field of

human-computer interaction (HCI). The field of **visual analytics** concerns the study of visual analytics systems, which are software tools that provide users with interactive visualization capabilities. Keim *et al.* define visual analytics as the combination of "automated [data] analysis techniques with interactive visualizations for effective understanding, reasoning and decision making on the basis of very large and complex data sets,"[226]. Visual analytics is therefore highly interdisciplinary, integrating both human factors and data science concepts to develop tools that help users to make sense of large-scale, heterogeneous, and interconnected datasets, including those which arise in bioinformatics contexts.

The single-cell field has acknowledged that visualization is key to the success of atlas interpretation. However, identifying the best visual encoding(s) and interactivity techniques to display such data remain unsolved problems. When outlining the goals and guiding principles of the Human Cell Atlas, the authors state, "...how best to capture and represent this information requires serious attention,"[359]. More recently, the field has acknowledged the difficulty of visualization as atlas datasets become more complex: "...as atlases increase in complexity, their visualization and interpretation also do so. Workflows should, therefore, be adopted to ease interpretation and visualization of atlases spanning tissue resolutions and omics layers,"[207]. In discussing the challenges of single-cell experiments in neuroscience, Bonev *et al.* include a section titled, "Data sharing, dissemination and visualization," which articulates a need to focus resources on visualization of spatial single-cell 'omics datasets to promote transparency, reproducibility, and improved understanding of biology[83].

Single-cell atlasing efforts present interesting challenges for visualization research (as well as computer science more broadly). First, a diverse group of individuals constitute the *users* who aim to analyze and interpret single-cell atlases, which human-

computer interaction and visualization collaborators must take into account. These users span from wet lab biologists performing experiments, to clinician-scientists involved in tissue donor recruitment and disease-specific atlas interpretation, to high school students learning about cell biology – none of whom necessarily have programming expertise. On the computationally-oriented side, users may span the spectrum of computational biologists, bioinformaticians, software engineers, and visualization experts. Van den Brandt *et al.* find that even among these latter four user personas (in the related area of genomics visualization), there are clear differences in how they approach their work, including problems of interest, level of automation used during analysis, and intended audience for authored visualizations[417]. Care must then be taken not to generalize user-centered findings from one user persona to all others.

Next, single-cell atlas generation and analysis are data-intensive. For this reason, the single-cell field has, from its onset, embraced data science and machine learning techniques[324]. However, the number of cells that can be profiled in a single experiment continues to grow each year, with individual experiments today already capable of profiling over one million cells[404,403]. Imaging-based experiments, in particular those that image in 3D, now routinely produce volumes whose storage requires terabytes of disk space[455]. The number and type of features that can be profiled per cell also continues to increase, resulting in heterogeneous datasets that are not trivial to align with each other[61,53]. Unlike other data-intensive fields such as atmospheric science or astronomy, data generation efforts for single-cell atlases are largely de-centralized, with many labs performing experiments and generating data in the form of collections of files. These files are challenging to merge and catalog not only due to differences in the sets of measured features and presence of experimental variations[279], but also due to the lack of a global coordinate system for alignment and

the heterogeneous nature of the data produced by cutting-edge instruments and experiments. There is also rich prior knowledge to leverage, including biological literature, ontologies, genome-wide association studies, and findings from model organism studies. While the problem of heterogeneous data within the contexts of biology and bioinformatics is not new[374], single-cell atlasing efforts are both accentuating this problem and have resulted in several proposed solutions[426,129]. Further, a focus on millions of cells as data points highlights longstanding visualization challenges (*e.g.*, overplotting), but also provides opportunities for establishing scalable domain-specific solutions.

1.5 Biological Background

Computational analysis – including visualization – of single-cell atlas data requires familiarity with the underlying experimental methods. In addition, it is beneficial to understand the goals and logistics of large atlasing efforts and how they compare to individual single-cell experiments.

1.5.1 Atlasing goals

The high-level goals of cell atlasing efforts have been thoroughly outlined by Regev *et al.*[359], Elmentaite *et al.*[148], Rood *et al.*[365], and Hrovatin *et al.*[207]. For instance, Elmentaite *et al.* discuss how cell atlases both reinforce and redefine our understanding of cell type composition and function across human tissue types[148]. Both Rood and Elmentaite articulate goals of surveying cell types and their functions across the human body, identifying cell types (and their interactions) that are relevant to disease, and linking these findings to genetics using existing genome-wide association study

(GWAS) data. Goals related to therapeutics and medicine are to use single-cell data to stratify patients, to compare drug delivery routes or predict pathogen entry routes, to diagnose disease, and to monitor disease or drug safety. These goals can be categorized into several groups, which span from basic biology to clinical translation (Table 1.1):

- Atlas Representation and Organization.
- Atlas Evaluation and Experimental Design.
- Cellular Taxonomy and Identification.
- Development, Trajectories, and States.
- Gene Regulation and Mechanism.
- Cell-Cell and Ligand-Receptor Interaction.
- Cross-Organ and Cross-Species Comparisons.
- Disease Diagnosis and Prognosis.
- Drug Development and Target Identification.
- Therapy Resistance Mechanisms.

We must consider these high-level goals when developing or evaluating low-level experimental techniques and computational methods, including visualizations and visual analytics systems. Yet there are also differing perspectives that have been raised, especially in relation to atlas representation and organization. Regev *et al.* propose the construction of an atlas with two 'branches': one focused on cataloging cellular

properties and the other focused on spatial organization of cells[359]. Domcke *et al.* call for the field to settle on a primary organizing principle for atlases to better facilitate projection onto an established reference in a manner similar to the human reference genome[139]. Adler *et al.* point out several ways of looking at cellular organization with underpinnings in complex systems theory, such as primary-supportive, instructive, and supplier-consumer[48]. Fischer *et al.* note that there is a wide gap between classic systems biology approaches and current utility of human single-cell atlas resources, noting sampling constraints and emphasizing that discoveries at a given spatiotemporal scale must be validated through experimental models that reflect this scale[158]. Nevertheless, the outlines of high-level goals serve as primers on the direction of the field and useful reference points.

1.5.2 Why 'omics?

Since the invention of the microscope, researchers have been able to observe cells and, by extension, record measurements at single-cell resolution. Such observations can be said to be measurements of *phenotype* (observable characteristics), which are the results of processes that occur at the molecular level within and among cells and their environment. In other words, processes that occur at the molecular scale are causal for the observed phenotype of each cell. This causal relationship then explains why we are interested in molecular biology in the first place: this is where interventions (*e.g.*, therapies or perturbations[138]) can be made that affect phenotype. (Note that in other contexts, all processes downstream of the genome are considered to be part of cell phenotype, including, for instance, the transcription of mRNA molecules). Owing to the complexity of cell biology, the development of such interventions today heavily depends on our ability to make measurements at the molecular scale. As there are

Goal category	Goal
Atlas Representation and Organization	Features can be viewed at any level of magnification, and high-dimensional information collapsed into simpler views (Regev et al. 2017)
	Enable atlas-guided analysis of new data (Hrovatin et al. 2024)
Atlas Evaluation and Experimental Design	Use for power analyses, that is, to estimate the number of cells, samples or donors that need to be profiled to answer specific questions (Hrovatin et al. 2024)
	Assess how different technical parameters affect the data to improve the design of future experiments (Hrovatin et al. 2024)
	Understand which cell types, diseases, demographics or other categories are understudied in the current data and need to be better captured in the future (Hrovatin et al. 2024)
Cellular Taxonomy and Identification	A comprehensive reference catalog of cells based on their stable properties and transient features, as well as their locations and abundances (Regev et al. 2017)
	A system of coordinates on which one can represent and harmonize concepts at many levels (Regev et al. 2017)
	Additional coordinates or annotations to represent histological and anatomical information (e.g., a cell's location, morphology, or tissue context), temporal information (e.g., the age of the individual or time since an exposure), and disease status (Regev et al. 2017)
	Identify cell types using a taxonomy based on morphological, physiological, and molecular differences (Regev et al. 2017)
	Spatially map cell types based on expression signatures to see how they relate to each other and connect them with cell types defined by morphology or function (Regev et al. 2017)
	Characterize the full range of normal cellular physiology: appropriate physiological behaviors and potential responses by a cell in presence of stimuli (Regev et al. 2017)
	Look up in which cell types a gene of interest is expressed and at which level (Regev et al. 2017)
	Derive expression signatures that uniquely identify cell types (Regev et al. 2017)
	Survey primary and supportive cell types and cell states across multiple organs (Elmentaite et al. 2022)
	Reveal cellular subtypes that are unique in particular locations (Rood et al. 2022)
	Reveal cellular subtypes that are unique in particular disease contexts (Rood et al. 2022)
	Explore marker genes, gene programs and the effects of biological and technical factors on cell types (Hrovatin et al. 2024)
	Evaluate newly identified or previously proposed marker gene signatures (including for tissue staining, cell sorting, or probe design) (Hrovatin et al. 2024)
	Establish consensus cell type annotations (standardize nomenclature and resolution) (Hrovatin et al. 2024)
	Reveal rare cell types that are often missed when analyzing individual datasets (Hrovatin et al. 2024)
Development, Trajectories, and States	Order cells with respect to cyclic processes, transient responses, and plastic states: derive gene signatures that reflect each phase and tease apart interactions among parallel processes (Regev et al. 2017)
	Reconstruct development as trajectories in high-dimensional space (Regev et al. 2017)
Gene Regulation and Mechanism	Resolve tissue-specific disease mechanisms by combining genome-wide association study (GWAS) and single-cell gene expression data across tissues (Elmentaite et al. 2022)
	Identify cell-type specific disease networks (Elmentaite et al. 2022)
	Identify alternative targets within disease-relevant pathways (Elmentaite et al. 2022)
	Identify the specific cell types, states, programs, and contexts where disease-implicated genes act (Rood et al. 2022)
	Identify gene–gene relationships and multicellular programs (groups of genes co-regulated across different cell types) (Hrovatin et al. 2024)
	Understand the molecular characteristics of phenotypes, such as disease, age or sex by analyzing associated changes in gene expression and cell-type composition (Hrovatin et al. 2024)
	Bring our understanding of anatomy, development, physiology, pathology, intracellular regulation, and intercellular communication to a new level of resolution (Regev et al. 2017)
	Understand the molecular mechanisms that control cell type, differentiation, responses and states - with cells, between cells, and between cells and their tissue matrix (Regev et al. 2017)
Cell-Cell and Ligand-Receptor Interaction	Understand interactions between abnormal cells and all other cells in a tissue ecosystem in promoting or suppressing disease processes (e.g., between malignant cells and the tumor microenvironment) (Regev et al. 2017)
	Identify cell-cell interaction networks relevant to cell therapies (Elmentaite et al. 2022)
Cross-Organ and Cross-Species Comparisons	What are the tissue-specific adaptations of omnipresent supportive lineages (for example, subtypes of fibroblasts) across organs? (Elmentaite et al. 2022)
	Study organism-wide cell function, development, organoid protocol design and evolution across species (Hrovatin et al. 2024)
	Use cross-condition differences to aid in selecting preclinical models (e.g., mouse models) (Hrovatin et al. 2024)
	Resolve commonalities between different subsets of specialized cells (Elmentaite et al. 2022)
	What are the core shared expression profiles within highly specialized lineages (for example, subtypes of neurons)? (Elmentaite et al. 2022)
	Resolve tissue-specific programmes of shared cells (Elmentaite et al. 2022)
	Reveal cellular subtypes that are shared across tissues (Rood et al. 2022)
	Identify conserved cell states among organisms by comparing across species atlases (Regev et al. 2017)
Disease Diagnosis and Prognosis	Provide markers, signatures and tools for basic research (facilitating detection, purification and genetic manipulation of every cell type) and clinical applications (including diagnosis, prognosis and monitoring response to therapy) (Regev et al. 2017)
	Use single-cell data to stratify patients on the basis of their cellular responses to disease (Elmentaite et al. 2022)
	Design molecular diagnostics based on precise cellular markers (Elmentaite et al. 2022)
	Identify cell of origin in adult and pediatric tumors (Rood et al. 2022)
	Develop diagnostic tools based on single-cell profiling (Rood et al. 2022)
	Predict drug toxicity (Rood et al. 2022)
	Predict drug efficacy and monitor the impact and state of cellular therapy in human patients (Rood et al. 2022)
	Clinical classification of patients based on their single-cell profile (Hrovatin et al. 2024)
Drug Development and Target Identification	Understand how both compositional and cell-intrinsic expression changes can be coordinated across multiple cell types, resulting in shifts in multicellular communities in disease. (Rood et al. 2022)
	Identify cell types that will be responsive to available treatments (Elmentaite et al. 2022)
	Predict organ of action and best drug delivery route (Elmentaite et al. 2022)
	Design cell therapies based on tissue-resident immune cell profiles (Elmentaite et al. 2022)
	Use shared molecular characteristics across conditions to identify drug repurposing opportunities (Hrovatin et al. 2024)
	Perform high-resolution phenotypic screens for desired cell states (Rood et al. 2022)
	Understand the cells in which disease-risk-associated variants are expressed and act (Rood et al. 2022)
	Aid in answering pressing biomedical questions, understanding disease mechanisms, developing new treatments, improving model systems and advancing disease prognosis or diagnosis (Hrovatin et al. 2024)
Therapy Resistance Mechanisms	Identify therapy resistance mechanisms, including cell communities that may predict response to therapies such as checkpoint inhibitors (Rood et al. 2022)
	Predict drug resistance mechanisms, both intrinsic and acquired (Rood et al. 2022)

Table 1.1: High-level goals of cell atlasing efforts, based on reviews and perspectives in the domain[359,148,365,207]. These are roughly categorized; in some cases a goal could be related to more than one category.

several types of such measurements (*e.g.*, for different types of biomolecules), the term 'omics refers to this general class of measurement techniques.

The information in this section and the section that follows is intended to provide sufficient biological background to understand the contents of this thesis; these sections are primarily geared toward readers from the computer science and visualization domains.

1.5.3 Which 'omics?

A cell's genome encodes (in DNA) a blueprint for its function. However, this blueprint encodes function much like a circuit or a state machine, whose operation depends on a number of inputs, including surrounding cells and other environmental stimuli, which all then contribute to the observed output (*i.e.*, phenotype). Notably, at any given moment, each cell is actively executing only a subset of its encoded instructions. Thus, knowledge of a cell's **genomic** information alone, while providing a base level of insight into potential programs it could execute, is not a sufficient readout of its operations or its state. By extension, this in part explains why completion of the Human Genome Project was not sufficient to understand all of human biology.

There is a continuum of 'omics techniques – hereafter referred to as *experimental modalities* – that target different types of biomolecules at different stages along the path from blueprint to phenotype in cells. While genomic measurements at single-cell resolution are possible, single-cell genome sequencing is not widely used outside of the studies of cancer and somatic mosaicism because multiple cells from the same individual are otherwise generally assumed to harbor the same genomic DNA sequence[272]. Instead, within the single-cell domain, the most widely-applied experimental modality to date has been **transcriptomics**. The transcriptome, or the set of RNA transcripts

in a cell, sits further along the continuum between DNA and phenotype, as the DNA in the genome encodes genes which must first be *transcribed* into RNA transcripts. This RNA can then (potentially) be *translated* into proteins. In other words, the transcriptome of a cell is dynamic and provides more information (compared to DNA) about the gene programs that are currently being executed. Single-cell transcriptomics have been widely applied due to several factors, including cost and maturity of experimental and statistical methods, but are not necessarily the superior readout of cell state. Transcript abundances do not have perfect correlation with abundances of the corresponding proteins (for instance, due to transcript degradation), a shortcoming which can be addressed through the application of alternative experimental modalities along the continuum.

In between the genome and transcription, there are complex regulatory processes that govern whether a given gene will be transcribed in the first place. These regulatory processes are tightly intertwined with the question of cell state (recall the state machine analogy) and can be measured via **epigenomics** modalities (a subset of which are termed **nucleomics**, measuring the organization, folding, and dynamics of DNA in the cell nucleus). Downstream of transcription, processes of protein translation or abundance of the resulting proteins (or post-translational modifications) can be profiled via **translatomics** (*e.g.*, ribosome profiling[422]) and **proteomics** techniques. Further along the continuum is **metabolomics**, which measures metabolites, the components (of cellular or environmental origin) of chemical reactions that occur within cells (or upon or outside, extracellularly). Within the field of metabolomics, there are subtypes such as lipidomics and glycomics, which measure lipid (fat) and glycan (sugar) molecules, respectively. These methods, while providing complementary (or richer) perspectives into cellular operations (compared to transcriptomics),

are envisioned to become more broadly applied as they increase in resolution and decrease in cost.

Finally, the -omics suffix is sometimes used to refer to phenotypic-level measurements, for instance in the fields of **connectomics** and **pathomics**, which extract morphological measurements at ultrastructural and microscopic scales, respectively, using imaging-based techniques[77,99]. While these modalities differ from traditional 'omics in that they do not profile biomolecules directly, they share other characteristics, including that such measurements can be performed at high-throughput and at single-cell resolution. In some cases, statistical and computational methods can be reused for both molecular and non-molecular modalities, and researchers often envision relating data between modalities.

Notes on Terminology

Individual 'omics refer to classes of experimental techniques that measure particular types of features. Specific experimental techniques within these classes may be referred to by name or abbreviation. Many experimental techniques involve sequencing DNA (regardless of whether genomic DNA is the feature of interest; in many cases, techniques rely on reverse transcription) and are named accordingly. To denote that a technique is performed at single-cell resolution, often the prefix 'sc' is used. For example, transcriptomic experiments are typically performed using RNA sequencing, abbreviated as 'RNA-seq', with 'scRNA-seq' referring to 'single-cell RNA sequencing'. While 'transcriptomics' has become synonymous with 'RNA-seq', this is not necessarily the case, especially for other modalities such as epigenomics which encompass several experimental techniques that are used for different purposes.

There are a number of 'omes for which there does not exist a one-to-one relation-

ship with an 'omics class of methods. The -ome suffix in these cases is used to refer to the totality of features of a particular category. For example, the *microbiome* and *matrisome* refer to the measurement of features that characterize microbes (*e.g.*, fungi, bacteria, and viruses) and the extracellular matrix, respectively[318]. While single-cell methods (*e.g.*, scRNA-seq) can be used to study these 'omes, these topics have traditionally been addressed in standalone studies.

1.5.4 Multiple Experimental Modalities

In the previous subsection, we described how differing experimental modalities can provide different views into cellular operations. A logical next step would be to combine multiple of these techniques in the same cell to obtain richer snapshots of cellular circuits. Indeed, single-cell biologists have developed **multimodal** experimental techniques in recent years, typically able to profile two or three modalities in the same cells. Currently, not all experimental techniques are compatible with one another (*e.g.*, destructive to the tissue sample), and as such, multimodal techniques are a major area of active research. We refer to Vandereyken *et al.* for a comprehensive review of multimodal experimental techniques published between 2015 and 2022, including spatially-resolved techniques[420].

Until the robustness, feasibility, and cost-effectiveness of multimodal methods increase, the majority of available single-cell data will remain **unimodal**. As a consequence, computational approaches are being developed to bridge this gap. There are two main classes of approach, the first being the use of one modality to predict another (*i.e.*, cross-modality prediction)[125], and the second being the learning of shared low-dimensional representations (*i.e.*, diagonal data integration)[61,453]. In some cases, dual-modality experimental data can serve as a molecular "bridge" to enable compari-

son and combined analysis of unimodal datasets[188].

1.5.5 Why Single-Cell?

For centuries, biologists have studied cellular phenomena, which 'omics techniques have facilitated at high-throughput for decades. Yet only in recent years has it become possible to preserve cellular identity from start-to-finish in 'omics experiments via the use of microfluidics[286] (or high-resolution imaging). In other words, readouts from 'omics methods have historically not contained a matching of biomolecules to their cells of origin. Biologists now refer to the previous techniques as 'bulk' (to distinguish from single-cell). Bulk is to single-cell as a fruit smoothie is to a fruit salad: in the former, we can get a vague sense of its components, but our understanding is limited by our inability to distinguish the individual original entities within[324].

In a bulk experiment, one cannot distinguish whether a difference in gene expression (between two cell groups, say health and disease) is due to a differing cell type composition (*i.e.*, number of cells of each class which are executing their normal state machine programs) versus due to altered gene expression (*i.e.*, same distribution over cell classes but a change in the state machine programs they are executing)[350]. Single-cell methods resolve this discrepancy, as, for instance in transcriptomics, every observed transcript can be mapped back to its cell of origin. Similarly, a "rare" cell type (which may be relevant to disease) that produces an atypical distribution of transcripts may effectively appear as noise in a bulk experiment. Rood *et al.* [365] review several examples, including one study which found cell type composition to be predictive of therapeutic response in Crohns disease, while in contrast, another study found coordinated shifts in gene expression across multiple cell types to be predictive of risk for ulcerative colitis. These reasons among others help to explain the enthusi-

asm around single-cell biology and great interest in profiling individual cells in large scale.

1.5.6 Spatially-Resolved Methods

It may not be apparent so far that many of the aforementioned 'omics methods do not preserve information about the spatial organization of cells (in technical terms, such experiments require *dissociation* of cells). Continuing the fruit analogy, non-spatial is to spatially-resolved as a fruit salad is to a fruit tart as a fruit salad is to a fruit tart[324]. In the latter, we observe not only the individual entities, but also how they are organized and their spatial relationships with one another. Spatially-resolved single-cell measurements can provide insights into signaling interactions among cells and characterize cellular neighborhoods or communities that organize into functional tissue units (FTUs)[334].

This preservation of spatial context, like preservation of cellular identity, has been a recent advancement, and remains an active area of research, termed 'spatial omics' (and sometimes also 'spatial biology'). Spatial 'omics comes with its own challenges including the development of spatial *and* multimodal experimental techniques and considerations of spatial resolution, tissue size & 3D profiling.

The advantages of 3D over 2D spatial data may not be readily apparent. In biology, it is common for structures within a tissue to extend beyond a 2D plane (*e.g.*, vasculature or nerves). Similarly, the distribution of cells sampled by a 2D slice may be significantly different than the distribution of cells in a larger 3D volume, potentially missing rare phenomena[1]. However, the increased complexity of both the experimental and the data analysis processes require justification for consideration of the additional dimension. As pathologists have traditionally analyzed 2D tissue slices,

pathological interpretation of 3D tissue volumes remains challenging[394]. These reasons explain the current prevalence of 2D spatial single-cell data and increasing interest in overcoming the challenges of 3D single-cell profiling.

1.5.7 Temporally-Resolved Methods

Cells are dynamic, responding to internal and external stimuli over time. Understanding these dynamics is critical but challenging, as many single-cell experimental methods are destructive, preventing repeated measurements of the same cells. In developmental biology, cells from different organisms are often profiled at various developmental timepoints, providing useful but limited snapshots due to differences between cells and coarse temporal resolution.

There are several alternative approaches that are designed for measuring cellular phenomena over time, each with different trade-offs and thus tailored to different types of biological questions. Live cell imaging uses time-lapse microscopy images of cells located in multi-well plates (typically tens to hundreds of wells)[354]. However, cells in multi-well plates are removed from their tissue context, limiting the conclusions that can be drawn from their study *in vitro*. In contrast, intravital microscopy is a form of *in vivo* imaging that can be performed over time using model organisms such as mice[352,236]. These imaging-based methods can be used to answer questions at short timescales but, as with other imaging-based single-cell techniques, they can be limited in terms of multiplexing (the number of different biomarkers they can measure). Sequencing-based readouts allow for high-throughput and highly-multiplexed single-cell profiles that either include temporal information directly (*e.g.*, in the form of DNA barcodes) or can be used to infer temporal relationships (termed pseudotime)[430,62].

On the other hand, algorithmic techniques enable inference of dynamics from standard single-cell 'omics experiments that would otherwise be viewed as snapshots. Pseudotime inference algorithms construct minimum spanning trees upon the manifold of high-dimensional measurements obtained from many cells, assuming similar cells in high-dimensional space to be closer developmentally. RNA velocity algorithms model the observed ratios of spliced to unspliced RNA transcripts to predict whether a cell is in a steady state vs. progressing to a different state, a process that occurs on a timescale of hours[246]. In the coming years, approaches that enable comprehensively profiling cells both temporally and spatially at multiple scales will become increasingly important and could serve as ground truth for the development of virtual cell models[100].

1.5.8 Experiment versus Atlas

A single-cell *experiment* refers to any procedure that involves quantitatively profiling multiple features of individual cells, typically via sequencing, imaging, or mass spectrometry-based techniques, in which individual cells can be distinguished from one another. An *atlas* undoubtedly refers to something larger and more complex than a single experiment, but what exactly?

Despite an abundance of biological reviews and perspectives on the topic of single-cell atlases, this term has not been well defined. 'Atlas' has been used in the single-cell literature to refer to different ideas, ranging from one to many single-cell studies profiling thousands to millions of cells, obtained from three to three hundred tissue samples, using one or more experimental modalities (including both spatial and non-spatial). In some contexts, a singular 'atlas' is described as a 'cell census' and explained using analogies like 'Google Maps', 'Wikipedia', & the periodic table (in this

context, there is presumably a single atlas)[2], while in other contexts the plural 'atlases' is used to refer to separate publications or data resources (in this context, there are multiple standalone atlases for different purposes). The initial outline for the Human Cell Atlas project called for publication of intermediate 'draft' atlases which build toward a 'full' human atlas, with 'sister' atlases for model organisms[359]. More recently, the notion of a single-cell 'atlas' has been framed as a foundation model[367]. Not only is the term 'atlas' used in the field without clear definition, the singular and plural forms are often used interchangeably by authors, especially in reviews and perspectives. We note that in geography, the term "atlas" is defined as a "collection of maps" (most commonly maps of Earth), in which geographic information and political boundaries may also be accompanied by statistics.

We define a "human single-cell atlas" as a collection of maps consisting of data from **multiple single-cell experiments from multiple tissue donors that are intended to summarize and communicate this data through hierarchical organization** (*e.g.*, cell → region → sample → tissue donor, as well as via ontology- or metadata-based hierarchy) **and spatial arrangement** (in physical space or latent space), **which may also be accompanied by statistics at each hierarchy level or at different spatial partitions**. There are multiple dimensions along which we might further classify an atlas, including whether it comprises one versus multiple organs, one versus multiple experimental modalities, physical versus only latent space organization, temporal versus not, and the count of cells profiled. Without the "human" qualifier, a more general "single-cell atlas" may map cells from one or more species, allowing an additional type of hierarchical organization via phylogeny and evolutionary relationships. We focus on human to limit the scope of this thesis.

This thesis delves into atlas-scale data, as this brings the potential for specific

types of within-dataset comparisons that would not otherwise be possible (*i.e.*, for an individual single-cell experiment that profiles the same number of cells from a single tissue sample from a single donor). Atlas generation efforts require overcoming important logistical challenges[148] which also represent opportunities for visualization research. Namely, collection of tissue samples must occur prior to single-cell profiling experiments, which requires coordination among different institutions and researchers, with metadata and provenance tracking required at multiple stages. Experiments may then be performed on the same tissue sample at multiple labs, complicating logistics, in particular when such experiments must be performed in a specific order that considers chemical compatibility and incidental ablation. As atlases comprise cells from multiple tissue donors, they can provide insights into inter-individual variability at cellular resolution, but must consider ancestral diversity of tissue donors if used to draw such conclusions[58]. During data analysis, logistical challenges include the usage of ontologies or otherwise consistent nomenclature, challenges of data integration in the presence of multiple experimental modalities, and uniformity of data processing pipelines used (if data from individual experiments are processed independently prior to data integration)[199].

1.6 Biomedical Insights from Single-Cell Techniques

Large-scale cell atlasing projects are often exploratory and descriptive in nature, with the understanding that comprehensive profiling and organization of cells will facilitate findings related to cell-cell interactions, drug target identification, disease diagnosis and prognosis, and improved understanding of gene regulation (Table 1.1). Indeed, data from single-cell atlases are beginning to contribute to translational findings.

Rood *et al.*[365] and Fischer *et al.*[158] discuss challenges in medicine which are amenable to investigation using cell atlas data and highlight recent studies in which human cell atlas data have been used to understand disease mechanisms. For example, analysis of adult and pediatric kidney tumor cells through comparison to non-tumor cells from a reference atlas revealed distinct disease mechanisms and cell types of origin which suggest different targeting strategies (targeting of Wilms' cells in pediatric tumors versus a convoluted proximal tubular cell subtype in adult tumors)[460]. Spatially-resolved single-cell profiling techniques have also contributed to recent translational advances, such as one study which used spatial proteomics to identify treatment with JAK inhibitor for a fatal skin condition[321]. A number of other studies have used single-cell atlas data to identify molecular, cellular, and cell-cell interaction biomarkers for validation in follow-up experiments, with relevance to cancer[217,214], COVID-19[317,476,442], lupus[348], and heart failure[238].

1.7 Single-Cell Atlasing Consortia

Cell atlas data portals aim to streamline and standardize data sharing and access. Data portals have been developed by funding institutions, atlasing consortia, and individual research groups. Many of these data portals contain integrations with interactive visualization tools. In some cases, these tools are tightly coupled with a data portal's infrastructure, while in others, a data portal embeds or links to a standalone interactive tool to enable visualization.

We highlight 19 atlas data portals mentioned by either Hemberg *et al.*[199] or Börner *et al.*[85], two recent reviews of single-cell atlasing efforts (Table 1.2). These are associated with 13 different consortia (11 NIH-funded, plus the Human Cell Atlas and

Single Cell Atlas consortia). Of these 19 data portals, we found that 12 contained embedded visualizations of single-cell data, while another 3 contained links to other tools for the purpose of single-cell data visualization. The remaining 4 data portals either did not contain any visualizations or did not contain single-cell resolution visualizations (for example, visualizing only file trees or high-level metadata). Of the 15 data portals containing embedded or linked visualizations, we found that 6 portals implemented custom visualizations that were tightly coupled to the portal, while the remaining 9 either completely or partially relied on an integration with a reusable tool such as Vitessce (Chapter 3), Cellxgene[303], UCSC Cell Browser[396], Minerva[357], UCSC Xena[174], or ShinyCell[331].

The majority of reviewed data portals did contain interactive visualizations of their single-cell data. However, the landscape of supported visualization features differed significantly across portals, with some offering highly interactive plots while others offered only carousels of static plot images. Further, within portals, the sets of available visualizations often differed among datasets. Separate visualization types were often split among multiple linked but independent tools. There is room for atlas data portals to offer more seamless navigation among multiple visualization tools and more consistent user experiences. We concur with Hrovatin *et al.*[207] that there is a need to develop interactive tools that are tailored to a broader range of user needs, as well as a need for the community to develop standards for storage and sharing of additional types of downstream atlas analysis results to facilitate interoperability of such tools (with each other and with data portal infrastructure).

Table 1.2: Atlasing efforts and data portals.

Data Portal	Source
CZ CELLxGENE Discover[129]	199
Single Cell Portal[406]	199
Single Cell Expression Atlas[167]	199
Human BioMolecular Atlas Program (HuBMAP)[210]	199
Human Cell Atlas (HCA)[359,371]	199
Allen Brain Cell (ABC) Atlas[137]	199
Tumor Immune Single Cell Hub 2 (TISCH2)[186]	199
Deeply Integrated Single-Cell Omics (DISCO)[265]	199
Single Cell Atlas[337]	199
BRAIN Initiative Cell Census Network (BICCN)[135]	85
Genotype-Tissue Expression (GTEx)[273]	85
ATLAS-D2K[302,332]	85
Gut Cell Atlas[149]	85
Human Tumor Atlas Network (HTAN)[372]	85
Kidney Precision Medicine Project (KPMP)[132]	85
LungMAP[60]	85
Stimulating Peripheral Activity to Relieve Conditions (SPARC)[328]	85
The Cancer Genome Atlas (TCGA)[301]	85

1.8 Data Structures for Single-Cell Analysis

In this section, we provide an overview of the characteristics of single-cell datasets. We map biological concepts (Section 1.5) to data abstractions to help researchers link single-cell experimental outputs to data science concepts. For example, we describe the typical shapes of matrix & tabular data structures and discuss contexts in which data structures such as trees, networks, images, and text are used.

1.8.1 Matrices and Tensors

The cell-by-gene matrix has emerged as a central data structure in single-cell transcriptomics. Across ecosystems of computational tools for single-cell data, similar 2D matrices are commonly used for inputs and outputs.

Observation by Feature Matrix

Single-cell transcriptomics measures the transcripts that are expressed by each cell. Upon sequencing of transcripts, the "raw" data consists of a collection of nucleotide sequences in which the cell identifier (in the form of a unique nucleotide barcode) is concatenated to the transcript sequence which together provide enough information to resolve both the cell and gene of origin for each transcript. Therefore, the "processed" form of a single-cell transcriptomics experiment is most often stored in a so-called "cell-by-gene" two-dimensional matrix in which rows are cells, columns are genes, and values are the per-cell per-gene transcript counts that follow from an analysis step called *transcript quantification*. For most downstream analyses, the transcript sequences themselves are not required and are assumed to be redundant (with the exceptions of RNA velocity[246] and copy number aberration[164] analyses).

As transcriptomics is not the only experimental modality, the cell-by-gene matrix is a special case of an "observation-by-feature" matrix, where the observations correspond to cells and the features correspond to genes. For instance, special cases of this matrix data structure are used to store chromatin accessibility ("cell-by-peak") and protein abundance ("cell-by-protein") measurements. The observation-by-feature abstraction allows for reuse of data analysis and visualization tools across experimental modalities.

1.8.2 Tables

Tabular data representations are widely used in the single-cell domain, as in many other fields. A notable difference however is the importance of alignments between certain dimensions of matrices and tables. For example, the observation (*i.e.*, cell) axis of the observation-by-feature matrix is commonly aligned to the rows of a per-observation table (*i.e.*, one row per cell, with an arbitrary number of columns) (`T-C` in Figure 1.1). The feature (*i.e.*, gene) axis of the matrix is commonly aligned to the rows of a per-feature table (*i.e.*, one row per gene, with arbitrary columns) (`T-G` or `T-P` in Figure 1.1). The data structures used in the domain enforce these alignments (*e.g.*, ensuring that table and matrix row indices match)[427,94]. These tables facilitate annotation of cells and genes with derived metadata such as cluster membership or alternative identifiers.

Derived quantitative attributes may be stored in per-observation or per-feature tables. For example, during quality control of the observation by feature matrix, the sum of transcript counts per cell may be stored in a `total_counts` column of the per-observation table. Categorical attributes derived from the observation by feature matrix include unsupervised cluster memberships, cell type annotations, or sample

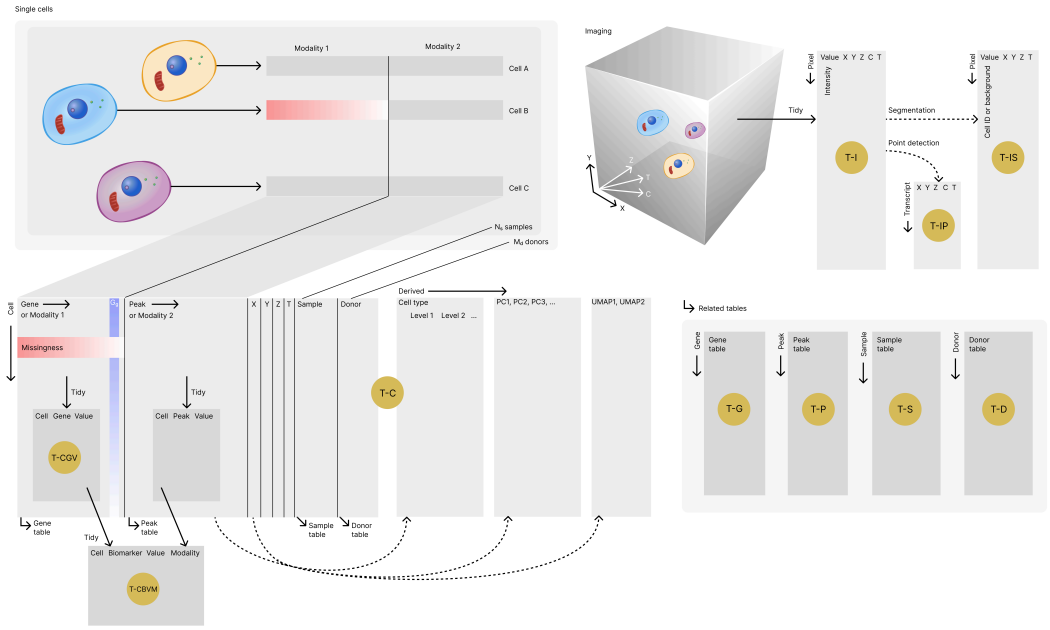

Figure 1.1: Tabular representations for single-cell datasets. Each yellow circle indicates a table in which rows are conceptually distinct. For example, in `T-C`, rows correspond to cells, while in `T-CGV` rows correspond to `(cell, gene, value)` tuples.

identifiers. These values are often stored in strings or categorical columns of the per-observation table.

Related Tables

There are a number of associated tables which do not align with an axis of the observation by feature matrix, for example containing metadata about samples or donors (`T-S` or `T-D` in Figure 1.1). These tables are often linked to the per-observation table via an identifier column, analogous to a foreign key relationship in a database system.

1.8.3 Spatially-Resolved Data

As noted in Section 1.5.6, spatially-resolved data is of increasing importance. There are multiple ways to represent spatially-resolved single-cell data, depending on both experimental and data analysis factors. The data analysis ecosystem for spatially-resolved experiments is less mature and lacks consensus around best practices (compared to non-spatial transcriptomics).

Imaging-Based versus Sequencing-Based

There are two main classes of experimental techniques which produce spatially-resolved data: imaging-based and sequencing-based. Imaging-based techniques include transcriptomics (imaging of transcript sequences along with their spatial coordinates) and proteomics (imaging of antibodies bound to proteins)[136]. Sequencing-based techniques typically incorporate barcodes containing spatial coordinates with RNA transcripts prior to high-throughput sequencing, with barcodes arranged or deposited on the tissue sample using a predetermined regular grid of spots or beads. The results of imaging-based transcriptomics techniques consist of $(x, y, \texttt{gene})$ tuples representing points identified via the image, while for imaging-based proteomics techniques the full grid of image pixels are more often used: $(x, y, \texttt{protein}, \texttt{pixel intensity})$. The results of sequencing-based techniques consist of the full grid of $(x, y, \vec{s})$ where $\vec{s}$ is a vector of transcript counts for all genes in the transcriptome captured in the spot. Emons *et al.* emphasize that data analysts should primarily be concerned with the data resolution and representation: point pattern (*e.g.*, transcript coordinates) versus regular lattice (*e.g.*, grid of spots) versus irregular lattice (*e.g.*, cell outlines)[150].

Images

Any microscopy image in which individual cells can be distinguished from one another may be considered spatially-resolved single-cell data. There are two main classes of single-cell imaging data: red-green-blue (RGB) and multi-channel ("multiplexed"). RGB images include brightfield and histology (*e.g.*, staining-based) techniques and contain three channels which can be represented and visualized in the same way as a natural image produced by a camera. In contrast, multiplexed images contain one or more channels and must be stored using formats that support an arbitrary number of channels. A single channel of a multiplexed image can be visualized as a grayscale image, or multiple channels can be visualized simultaneously by mapping each to a different color and blending to produce a pseudocolored image[437]. Data representations for bioimages differ from representations for natural images in several ways: consideration of five dimensions (X, Y, Z, C for channel, T for temporal) as opposed to three (X, Y, RGB), consideration of experimental metadata, and consideration of multiple resolutions (*i.e.*, image pyramid support). Images are the central representation for certain experimental techniques, including antibody-based imaging of proteins, in contrast with those where the central representation is points or spots.

Points

Techniques which image RNA transcripts often involve nucleotide encoding schemes and multiple rounds of imaging, such that the original imaging data is not utilized beyond quality control. Instead, the raw images are used to derive points (x, y coordinates) which correspond to localized transcripts[112]. Point-based imaging methods such as MERFISH may also be performed in 3D, localizing transcripts to (x, y, z) co-

ordinates (Section 1.8.3). Only the spatial coordinates and transcript identity (*i.e.*, corresponding gene) are recorded; the physical size of such points is generally not considered. Downstream of point identification, data analysis may include point pattern ("segmentation-free") analysis[150,287] or cell segmentation[349] (facilitating subsequent cell-centric analysis).

Spots

In contrast to point-based techniques which typically use imaging, spot-based techniques use sequencing to profile gene expression along a regular grid of observations (spots). This regular grid of observations differs from both traditional image pixels and the irregularly-spaced nature of point-based data. Historically, the resolution of spot-based techniques was approximately 55 μm, often capturing more than one cell per spot (and with gaps between adjacent spots). This led to the development of computational "spot deconvolution" methods to decompose each spot into proportions of predicted cell type categories[264]. Today, spot-based techniques can achieve 2 μm (*i.e.*, subcellular) resolution (and without gaps)[326], introducing the opposite problem: how to aggregate multiple spots which are predicted to correspond to a single cell[353].

Points correspond to an instance of a feature (*e.g.*, an RNA transcript) and may lack a recorded physical size. Spots, on the other hand, correspond to an observation vector of feature counts (*e.g.*, count of transcripts per gene), have a physical size that is associated with the experimental technique or instrument (*e.g.*, 2 μm for Visium HD), and are often arranged in a regular grid. While both points and spots can be visualized using circular visual marks positioned at their spatial coordinate, the distinctions between these data representations should ideally be reflected in their visual

encoding.

Segmentations

Cell segmentation is a crucial data analysis step for all spatially-resolved experiments which achieve subcellular resolution, including those represented by images, points, or subcellular spots. The process of cell segmentation entails annotation of data points (*e.g.*, individual pixels) with unique cellular identifiers. Cell segmentation is a form of instance segmentation (in contrast to semantic segmentation methods which produce a binary annotation such as cell versus non-cell). Segmentation results may either remain as annotations (*e.g.*, mapping of pixels/voxels/points to cell IDs) or may be aggregated or otherwise simplified (*e.g.*, polygon/mesh per cell). When a segmentation algorithm output is represented using pixels, it is called a "segmentation bitmask" also known as a "label image". In such a bitmask, pixels with value 0 represent background, while pixels with value i represent the ith cell. In contrast, polygon-based segmentations store cellular boundaries using polygon geometries (accompanied by a mapping between polygons and cell identifiers). The 3D analog of a bitmask is a label volume in which voxels indicate cells, while the 3D analog of polygon-based segmentation is mesh-based segmentation.

Current widely used algorithms for cell segmentation include CellPose[400], Mesmer[179], and Baysor[349]. CellPose and Mesmer are deep learning-based methods which operate on imaging data, while Baysor is a probabalistic method that operates on point data. Despite these algorithms being considered state-of-the-art, it remains challenging for current cell segmentation algorithms to generalize across different imaging techniques and tissue types, especially those in which cell morphology varies[284]. A recent analysis by Hartman and Satija demonstrates that novel metrics are needed

to evaluate cell segmentation methods, such as quantification of gene pairs that are observed to be co-expressed based on segmentations despite exhibiting mutually exclusive patterns in non-spatial experiments[191]. The development of cell segmentation methods remains an active area of research, and visualization is an important tool for qualitative evaluation of segmentation quality.

While segmentation is a challenging task, it is valuable in that it facilitates the conversion of pixel- or point-based representations to a cell-based representation: a cell-by-feature matrix. Once cell boundaries have been computed, it is straightforward to quantify features on a per-cell basis. For instance, the cell boundary facilitates counting of transcript points corresponding to gene j that are contained within the boundary of cell i, matrix entry c_{ij}. Thus, with the observation-by-feature matrix, the hundreds of tools[463] designed for analysis of this matrix (despite not necessarily intended for spatially-resolved data, specifically) become possible to reuse and apply. Similarly, insights derived on a segmented-cell level are more interpretable and easier to validate than those that remain in terms of pixels or points, for instance with validation via non-spatial experiments or interpretation through relation to existing biological literature that describes phenomena in terms of cell types.

In addition to segmentation being performed within a data analysis workflow, in experimental procedures such as deep visual proteomics, segmentation is performed in real-time during the experiment[368]. In deep visual proteomics, cells under a microscope are segmented and the resulting cell boundaries are used to excise the cells from the tissue using laser microdissection. The excised cells fall into a multi-well plate where they can be prepared for proteomics performed using time-of-flight mass spectrometry.

2D versus 3D

In prior sections, we mentioned cases in which spatially-resolved experiments produce three-dimensional data, such as volumes or z-axis coordinates. While not all experimental techniques are compatible with intact tissue blocks, in such cases multiple aligned 2D slices can be "registered" to produce a 3D dataset. We therefore distinguish between "true 3D" datasets (in which the z-axis is equally as dense as the x- and y- axes) and so-called "2.5D" datasets (in which the z-axis is more sparse than either the x- or y- axes) that originate from physical slicing and subsequent registration. Representation and visualization techniques for 2.5D and 3D single-cell datasets are open areas of research[200,314].

Coordinate Systems

Unlike in geography, the single-cell domain does not have a single established coordinate system upon which to align and analyze spatially-resolved data. This also stands in contrast to genomics, where the coordinate system of a reference genome enables alignment. There is no consensus on how to align datasets from multiple spatially-resolved 'omics experiments onto a common coordinate system (at the organ or whole-organism scales) despite multiple proposed approaches[366,438,465,219,146]. However, the field is beginning to establish standards for the representation of coordinate transformations, including for affine transformations between imaging fields of view and for nonlinear transformations (*e.g.*, for registration of adjacent 2D slices)[311,291].

Regions of Interest

During analysis of spatially-resolved data, an analyst may identify regions of interest (ROIs). Data representations for regions of interest will vary by the type of annotation, but the aforementioned data representations still apply: bitmasks, polygons, and points (potentially with associated quantitative or categorical information). Notably, the difference in semantics for a region of interest may influence the visualization design (*e.g.*, different visual encoding for bitmasks representing ROIs versus segmentations). Visualizations of regions of interest are relevant to quality control, pathology, and narrative contexts.

1.8.4 Graphs and Networks

Graph-based approaches are widespread throughout computational biology[477]. In single-cell contexts, graph-based representations may store pairwise relationships between observations (*i.e.*, nodes corresponding to cells) or between features (*i.e.*, nodes corresponding to features such as genes or proteins). Edges may be directed or undirected and may be associated with categorical labels or quantitative values (*i.e.*, edge weights). Nearest neighbor graphs of observations are commonly used, with edge weights corresponding to distances computed in high-dimensional space or in physical space (for spatially-resolved data). Common types of graphs of features include protein-protein interaction networks and gene regulatory networks. While the aforementioned graphs can be stored using standard graph data structures (*e.g.*, adjacency matrices or edge lists), a challenge is that the high-dimensional nature of the data allows for attributed graphs with thousands of attributes, which can be difficult to store and analyze.

Trees, Hierarchies, and Ontologies

Tree-structured data in this domain include ontologies, hierarchies, phylogenetic trees, and trajectories. Relevant ontologies include cell ontology (describing cell types), gene ontology (describing gene function), experimental factor ontology, medical concept ontologies (*e.g.*, MONDO), and anatomical structure ontologies (*e.g.*, UBERON). These ontologies can be represented using directed acyclic graph (DAG) data structures, and may include inter-ontology linkages. Other trees include the results of hierarchical clustering algorithms. Trees or graphs may store hierarchical representations of observations, for example, to indicate multiscale spatial relationships among cells and organelles, vessels, nerves, or functional tissue unit(s).

1.8.5 Genome-Mapped Data

We refer to Nusrat *et al.*[325] for an overview of genome-mapped data abstractions for visualization, yet highlight those that arise in single-cell contexts specifically. As noted in Section 1.5.3, single-cell genomics (*i.e.*, sequencing of genomic DNA) is not often the focus of single-cell studies; experimental modalities such as transcriptomics, epigenomics, proteomics, and metabolomics are typically the focus. However, for all modalities that use sequencing as a measurement technique, genome-mapped data will be produced (despite not necessarily being the data representation used for the majority of analyses). For example, the sequencing reads corresponding to transcripts in transcriptomics studies would be considered genome-mapped data, but a standard initial analysis step is to transform raw sequencing data into an cell-by-gene matrix of transcript counts. A cell-by-gene matrix would not be considered genome-mapped directly, as genes are an abstraction for genomic coordinates (but could be considered

indirectly genome-mapped via the coordinates of each gene in a reference genome). In contrast, studies of epigenomic modalities (including chromatin accessibility, also known as scATAC-seq) often involve direct analysis of the genome-mapped data (*i.e.*, retaining the genomic coordinates of data points during analysis). It is common to reduce the approximately 3 billion base pairs of the human genome into fewer features, either using fixed-width bins of hundreds-to-thousands of nucleotides or performing "peak-calling" to identify variable-width regions (*i.e.*, peaks) of enriched signal[451,176]. Regardless, these bin- or peak-based features remain directly associated with genomic coordinates, as unlike genes they are analysis-derived and experiment-specific, as opposed to being pre-established units of biological function. Single-cell chromatin accessibility data is therefore typically represented using a cell-by-bin or cell-by-peak matrix (both special cases of the observation-by-feature matrix and simultaneously genome-mapped).

1.8.6 Temporally-Resolved Data

Temporally-resolved single-cell data may be represented by a temporal data axis, for example, using the T axis of a bioimage format. Pseudotemporal orderings may be stored in a column of the per-observation table. RNA velocity vectors may be stored using an observation-by-feature matrix or a graph data structure. We note that outside of these cases, the standard data structures in the domain do not yet comprehensively support temporally-resolved data.

1.8.7 Experimental and Clinical Metadata

Experimental metadata consists of information about the experimental protocol and instrument used, for example. Sample-level metadata consists, for example of the physical dimensions of a tissue sample and the tissue preservation method. If multiple iterations of tissue sectioning are performed, there may be sample-level metadata for each tissue section, or for larger (pre-sectioning) tissue blocks. Sample-level data may also include pathological annotations if the samples have been reviewed by a pathologist. Finally, there may be clinical metadata at the tissue donor level. Publicly accessible clinical data associated with single-cell data is typically high-level to protect donor privacy, and may consist of variables such as age, sex, race, and body mass index (BMI), smoking history, and history of cancer[210]. For disease-focused atlas efforts, disease-specific (yet still high-level) clinical variables may be available, such as disease subtypes[132]. Experimental and clinical metadata are typically represented using associated tables[202] (Section 1.8.2) or knowledge graphs[397,477].

1.8.8 Text-Based Data

Text-based data in the single-cell atlas context may include descriptions of biological function for genes, cell types, or pathways. It could also include text for narration or storytelling purposes, for example in tools like Minerva[357]. Large language model-based chat interfaces are also gaining interest, for example CellWhisperer[379] and PathChat (in the related area of digital pathology)[276]. Recent work in the visualization field has found that audiences prefer visualizations with an abundance of text annotations[399] resulting in a call for treating text and visualization co-equally[197].

1.8.9 Multimodal Data

Data resulting from multimodal experimental techniques (Section 1.5.4) can often be represented by multiple cell-by-feature matrices, aligned on their cell axis. For example, SNARE-seq[114] enables profiling both the transcriptome and chromatin accessibility in the same cells, producing both cell-by-gene and cell-by-peak matrices. Data structures such as MuData[94] operate as a container for multiple such cell-by-feature matrices, with options for enforcing alignments among different matrix dimensions. The question of how to store the results of experiments that produce both spatially-resolved data *and* multimodal data remains open, yet data structures such as SpatialData[291] are beginning to address this. For visualization, it is paramount that data structures represent (and facilitate queries of) the correspondences between cells and features across modalities to enable analysis tasks involving relation[386].

1.8.10 Dataset Size

Single-cell transcriptomics studies have been profiling increasing numbers of cells over time, with cell counts increasing exponentially[404,403]. However, these prior surveys of single-cell studies have been focused on the transcriptomics modality and on individual experiments (as opposed to atlases). Hemberg *et al.* recently summarized the numbers of cells contained in a set of single-cell atlas data portals, finding cell counts ranging from 4 million (in Allen Brain Cell Atlas) to 84.3 million (in CZ CELLxGENE Discover)[199]. An important consideration is how many cells are required to answer a given biological question. Power analysis approaches have been proposed, including for spatial 'omics experiments[64]. More research is needed to establish power analysis approaches for additional types of biological questions and experimental

modalities, as well as to integrate power analyses into interactive tools.

1.8.11 Data Access Patterns

The large-scale nature of atlas data means that it can be prohibitive to load an entire atlas dataset during analysis. Especially for visualization, it can be beneficial to load subsets of data on-demand. Interactive visualization tools can take advantage of data access patterns to ensure low latency and high performance rendering.

Common data access patterns for visualization include column-wise access of the observation-by-feature matrix (*i.e.*, loading all data for a feature of interest), column-wise access of per-observation tables (*i.e.*, loading all data for a column of interest), and access of row subsets of matrices and tables (*i.e.*, loading data for a subset of observations, for example a certain cell type or spatial region).

For spatial and imaging data, patterns include access to a spatially-defined subset of data. The user's zoom level and current field of view dictate the minimum amount of data to load. Tile-based approaches facilitate loading only the data for a given field of view. For data which is straightforward to aggregate, including image pixels and voxels, pyramid-based data structures (image pyramids) enable pre-aggregation at several different resolutions (typically based on powers of two). In cases of multiplexed imaging, it is common for users to be interested in a subset of image channels. Further, it is only feasible to blend a handful of colors at a time[437]. Thus, it makes sense to be able to access the data for each image channel independently. Similarly, users may be interested in viewing individual Z- or T-slices of volumetric or temporal data. For genomic data, similar approaches (tiled and multi-resolution data structures) can be used (based on genomic coordinates as opposed to spatial coordinates).

1.8.12 File Formats

In the single-cell domain, data analysts use specialized file formats. For bioimaging data, the OME-TIFF and OME-NGFF formats are used[312,311,313]. For unimodal data, AnnData objects are used in-memory and stored on-disk using either the HDF5 or Zarr formats[427]. For non-spatial multimodal data, MuData objects are used in-memory and can also be stored on-disk using HDF5 and Zarr[94]. The SpatialData object is a container data structure for AnnData objects as well as OME-NGFF-based bioimages and label (segmentation bitmask) images, polygons, and points. SpatialData objects are stored on-disk using a combination of the Zarr and Parquet formats[291]. In the R data analysis ecosystem, the Seurat, SingleCellExperiment, and SpatialExperiment formats are commonly-used alternatives to AnnData, MuData, and SpatialData[187]. Appendix D provides more information about the Zarr format and its usage in the R ecosystem.

For large-scale data, it can be useful to work directly with these formats during visualization (rather than performing conversion), as this enables interoperability and avoids computational overhead. We emphasize that this domain has benefited from visualization data access patterns being considered when establishing file format standards. Thus, many of the aforementioned formats facilitate efficient partial reading operations in parallel.

1.9 Computational Methods for Single-Cell Data Analysis

The large-scale and high-dimensional nature of single-cell datasets has necessitated interdisciplinary collaborations between the fields of computer science and single-cell biology. As the scRNA-tools database shows, there are now over one thousand soft-

ware tools and computational methods for single-cell transcriptomics data alone[464]. Here, we provide background on key stages of single-cell data analysis and note which tools and data structures commonly arise at each stage. For more comprehensive analysis best practices, we refer to Luecken *et al.*, Amezquita *et al.*, and Heumos *et al.*[278,201,57].

1.9.1 Quality Control and Normalization of Observation by Feature Matrices

Initial processing steps ("preprocessing") applied to an observation-by-feature matrix include those of quality control, normalization, and standardization. Due to a proliferation of differing approaches to preprocessing, it can be challenging to determine how a given observation by feature matrix was processed and how to avoid introducing biases when aiming to merge matrices ("data integration") that have been processed differently. Sarfraz *et al.* therefore recommend the inclusion of metadata and provenance information to facilitate re-analysis and data integration[376]. In the context of visualization, we emphasize that the choices of normalization and standardization approaches can dramatically affect distributions of feature values (*e.g.*, gene expression values) and therefore the perception of the shape of such distributions upon visual encoding. Visualizations of feature values should be annotated with provenance metadata, and in interactive settings, the results of different normalization methods may be offered for user selection.

Quality Control

For sequencing-based transcriptomics data, there are a number of quality control strategies which aim to remove low-quality observations corresponding to damaged cells or erroneous profiling of two cells (doublets)[278,57,206]. The level of tolerance which is applied when filtering observations will depend on both experimental conditions (*e.g.*, some tissue or cell types may be differently prone to different phenomena) and intended downstream analyses. Strategies for quality control can be divided into those where a user specifies a filtering threshold directly (*e.g.*, discard observations in which zero transcripts were identified for G or more genes) or indirectly via posterior probabilities or median absolute deviations[278,57,206]. In some cases, thresholds will be manually specified upon visualization of quality control metric distributions (*i.e.*, human in the loop), while in fully automated data processing pipelines (*e.g.*, implemented within large atlasing consortia) such thresholds may be hard-coded[210]. For spatial transcriptomics, imaging, and other experimental modalities, different quality control metrics must be considered[201,413,65]. Downstream processing steps such as cell segmentation may require their own quality control procedures[191].

Normalization and Standardization

Sequencing-based transcriptomics data analyses often require normalization of the raw counts of transcripts in order to correct for technical variability (*i.e.*, noise) that could confound biological variability (*i.e.*, signal). Components of technical variation may include stochastic differences in cell lysis and the efficiency of the reactions prior to sequencing, which can impact the total number of sequenced transcripts per cell ("sequencing depth"). Hafemeister and Satija articulate two main goals following nor-

malization: expression of a gene within a cell should not be correlated with the sequencing depth of the cell, and variance of a gene's expression among all cells should be independent of the gene's mean expression level and sequencing depth[181]. A number of statistical models have been proposed for normalization and correction of undesired technical artifacts[414,181,253]. While there are many benchmarks of normalization methods for single-cell transcriptomics data, Ahlmann-Eltze and Huber state that the optimal method depends on the biological question(s) of interest, "There is no context-free measure of success for a preprocessing method, as it is contingent on the objectives of the subsequent analysis," but note that a common objective is to understand "...the variety of cell types and states in terms of a lower-dimensional mathematical structure...,"[50]. For imaging or non-transcriptomics data modalities, the characteristics of data distributions are different and necessitate alternative normalization approaches[315,305,293,385]. Normalization is sometimes followed by a standardization (z-scoring) procedure to center the expression vector for each gene at a mean of zero (with standard deviation of one)[376].

1.9.2 Dimensionality Reduction and Embedding

Dimensionality reduction (DR) is commonly applied to make sense of the high-dimensional observation-by-feature matrices that emerge from single-cell experiments. The (often implicit) hypothesis that underlies the usage of DR is that the biological processes of interest are significantly lower-dimensional than the number of genes (or other features). Thus, DR can meaningfully preserve important structures from the high-dimensional space and facilitate interpretation (via visualization) or streamline analysis (via reduced computational complexity).

The idea that high-dimensional single-cell data exists on a low-dimensional man-

ifold is closely related to the developmental landscape model proposed by Waddington in 1940 (predating the emergence of single-cell data)[428]. The controversy that would arise from widespread application of such a model was predicted by Haldane, "Like other symbols in biology, I expect that this one will prove valuable for a time but may later be rather misleading,"[183,355]. Application of DR to single-cell data has generated controversy among computational biologists and biostatisticians. Namely, there are concerns about misinterpretation of structure that emerges in two dimensions and performing statistical tests on groups of data points identified via the low-dimensional space without accounting for having performed DR ("double dipping")[431,163,118]. However the prevailing perspective is that DR can be useful for hypothesis generation and statistical testing, assuming findings are validated through other means and the proper test corrections are applied[237,254].

There are several takeaways for visualization researchers. Single-cell biologists seek interpretable low-dimensional representations of high-dimensional data matrices, using both linear and nonlinear methods. Single-cell biologists most often perform dimensionality reduction along the feature axis (as opposed to along the observation axis[423]), with the output being stored in a matrix with shape (N cells $\times$ D latent features). The single-cell domain is active in usage and development of DR methods[72,310,319] and seeks ways to validate DR-based findings. Appendix C contains a bibliometric analysis of dimensionality reduction algorithm usage not only in biology and medicine, but across application domains.

1.9.3 Comparative Data Analysis Methods

The notion of comparative data analysis is vague, however in this section we mean analyses which involve comparisons across conditions. For example, case-versus-control

style, such as treatment versus placebo or disease versus non-disease, or comparisons of a quantitative variable such as age. There are many statistical and computational methods designed to perform these types of comparative analyses of single-cell datasets. Commonly-used methods include differential gene expression analysis, compositional analysis, gene set enrichment or pathway analysis, and perturbation modeling. We refer to the Single-cell best practices resource for an overview of these techniques[278,201] and further elaborate on them in Chapter 4.

1.9.4 Spatial Data Analysis Methods

The development of methods for analyzing spatial and imaging data represents a highly active research area within computational biology. There are currently approaches which span multiple scales, from analysis of ligand-receptor interactions, to understanding which cell types co-localize, to segmentation of larger functional tissue unit (FTU) structures. While some approaches draw upon geospatial statistical techniques, others use graph theoretical or computer vision techniques. We provide further background on spatial data analysis methods in Chapter 5.

1.9.5 Data Integration

As noted in Section 1.8.1, single-cell data exhibits variability from both biological and technical factors. Technical factors (*e.g.*, experimental protocol or sequencing instrument differences) can be especially pronounced when analyzing data from multiple experiments or experimental modalities. Data integration is the process of combining data from multiple experiments or modalities and therefore involves correction for technical variability while preserving biological signal[61,119]. Integration method

outputs generally fall into one of three categories: graph-based, joint embeddings, or corrected observation-by-feature matrices[279].

Data integration is an umbrella term for four distinct subtypes: shared-observations, shared-features, mosaic, and diagonal[61]. For example, when the same set of cells have been profiled using multiple experimental modalities, shared-observations integration is used (*i.e.*, the matrices can be aligned along the observation axis). On the other hand, when different sets of cells ("batches") have been profiled using a single experimental modality, shared-features integration is used (sometimes called batch correction). Mosaic integration refers to usage of a patchwork of shared-observations and shared-features[471]. Finally, when different sets of cells are profiled using different experimental modalities, diagonal integration is required.

1.10 Evaluation of Interactive Systems for Single-Cell Data Visualization

As this dissertation involves the development of software and interactive visualization systems, here we provide background on several evaluation approaches that can be applied in the context of research that involves visualization and software tool development.

A key method from the information visualization domain is the Nested Model for visualization design and validation proposed by Munzner[316]. This model consists of four levels:

- Domain problem characterization.
- Data abstraction design.

- Visual encoding and interaction technique design.
- Algorithm design.

At the domain problem characterization stage, the goal is to understand the target user audience and the questions they ask of their data (in domain terms). Data abstraction design entails translation of problems and data types from domain-specific language into the language of computer science (*e.g.*, Is the data tabular? Do table columns contain quantitative versus ordered versus categorical information?). This stage can also involve determination of how best to transform the data to facilitate downstream visualization. In the visual encoding and interaction design stage, the visualization designer decides which data fields to map to particular visual marks, visual channels, or interactions, with Munzner deferring to existing theories for comprehensive coverage of these topics. Finally, in the algorithm design stage, the developer uses principles from computer science to ensure that the system can efficiently perform the visual encodings and interactions that resulted from the previous stage. Munzner emphasizes that the model not only serves as a guide during the design process, but also serves to guide the evaluation of a visualization system, as validation can be performed at each level.

We can alternatively use the Design Arguments framework proposed by Glassman* to motivate (or evaluate) the need for a tool or method (especially software, but not necessarily). This framework requires answering the following questions:

- What is the **need**?
    - Who are the user(s), in what situation(s)?

*Glassman E., Harvard CS279R, Fall 2022

  - What goal(s) do they currently have, or might they have if the obstacles were removed?
  - What obstacle(s) are they encountering or would they encounter on their way to these goal(s)?
  - What evidence is there for this need?

- What is the proposed **approach**?
  - What are the characteristics of the approach?
  - How does each characteristic address one or more obstacles?
  - What evidence is there that this would be an effective approach?

- What is the **novelty**?
  - What are the 'delta's? (*i.e.*, how would this work differ from its nearest neighbors in the literature?)

In this framework, we decouple the need argument from the approach argument, avoiding premature selection of any particular approach and facilitating comparison of alternatives. We are forced to articulate the intended user audience and usage setting. By enumerating the obstacles and subsequently mapping characteristics of the approach to individual obstacles, we can easily determine which obstacles our proposed approach will address. The design arguments framework is a helpful first step to take when embarking on a research project that involves tool development because it can be used in the absence of user interviews or user studies and in the absence of the proposed software tool.

Once the intended user audience and usage settings have been articulated, we can consider motivation and evaluation approaches that involve humans. We may know that some target user audience has a certain goal, but it may be unclear whether this audience is encountering obstacles in achieving this goal. Alternatively, we might have a hunch that users are encountering certain obstacles and thus have a particular need, and we might be interested to gather evidence that can confirm or reject our hunch. In these cases, we can perform a *need-finding* study, which typically involves semi-structured formative interviews with individuals from the specified user audience. The semi-structured interview format entails a predefined set of interview questions, with the potential to deviate or ask follow-up questions based on interviewee responses. For example, Chang *et al.*[107] performed a need-finding study with information workers to better understand challenges in web browser tab management (which subsequently informed the development of a web browser extension addressing the identified challenges[108]). These types of semi-structured interviews primarily produce qualitative data, for instance, in the form of interview transcripts and notes. To derive more structured insights from these interviews, we draw upon the field of qualitative data analysis[168]. The results and insights from a need-finding study can be valuable research contributions on their own (*i.e.*, even in the absence of solutions to address any identified obstacles and needs). Such contributions serve as evidence and justification that can be referenced in follow-up literature and can motivate and inform a broader set of researchers who may engage in addressing the identified obstacles.

Following an initial implementation of a tool to address some need, there are multiple evaluation options (including, but not limited to, user studies). Ko *et al.* provide a practical guide covering ten considerations when designing studies that involve human participants for evaluation of software tools: recruitment, selection (of participants

according to inclusion criteria), consent, procedure (*e.g.*, script for the investigator), demographic measurements, group assignment, participant training, participant tasks, outcome measurements, and debrief and compensation[235]. In their perspective, "Usability Evaluation Considered Harmful (Some of the Time)," Greenberg and Buxton discuss how user studies are in many cases a relevant evaluation approach, but not always, and when they are used they should be complemented by alternative approaches[178]. For example, if a user study is performed prematurely (*e.g.*, on a sketch or prototype), users may be prone to focus on early bugs such that a researcher declines to pursue this design avenue that might otherwise provide user benefits. Greenberg and Buxton also caution that usability evaluation approaches provide limited insight into how novel software systems might be adopted by users over time and incorporated into everyday activities.

In contrast to a user study, in a case study form of evaluation for a tool, we present a usage scenario and walk through how our tool would support a user in this scenario. Such a case study may demonstrate achievement of a goal that was articulated in a need-finding interview, or instead may demonstrate that some prior finding in the literature that previously required some painstaking approach can be more easily reproduced using the tool. This thesis has primarily made use of design arguments, need-finding studies, and case studies for evaluation of contributed software tools.

1.11 Contributions

This dissertation supports the thesis statement (Section 1.3) through the contribution of new visual analytics systems and techniques for representation and interaction of single-cell experiment and atlas datasets. Chapter 2 presents a survey of

visualizations for single-cell atlas data found in the biological literature. This survey provides a reference point for the current state of atlas data visualization in the single-cell domain, upon which improved techniques can be developed. Chapter 3 introduces Vitessce, a framework for developing web-based interactive single-cell data visualizations. This framework addresses key challenges, including the need to tailor visualizations to the data types generated by different experimental techniques and the need to coordinate interactions across multiple related visualizations. Further, Vitessce addresses logistical challenges, enabling usage of interactive visualizations in different computational environments, facilitating deployment and sharing of visualizations, and reducing data conversion requirements by supporting multiple file formats that are commonly used to store single-cell data. Chapter 4 builds upon the Vitessce framework to support visual comparisons of single-cell data originating from multiple experiments. In this chapter, we develop interactive visualizations to address two separate comparative scenarios: the first being query dataset versus reference atlas comparison, geared towards the task of reference-based cell type annotation, and the second being case-versus-control comparison, for contrasting a pair of sample groups. Chapter 5 presents multiple visualization approaches for exploration of spatially-resolved single-cell data. This chapter first discusses the Viv software library for efficient web-based visualization of large multiplexed bioimaging datasets. Then, the chapter explores an extension of Vitessce which enables the simultaneous visualization of imaging data with overlaid segmentations of functional tissue units (FTUs) and derived attributes. Finally, the chapter discusses visualization of the outputs of a statistical method which identifies spatially-enriched co-occurring sets of cell types. Chapter 6 explores multiple approaches for visualization of genome-mapped data, including epigenomics and single-cell chromatin accessibility data. Concluding

chapter 7 presents a thesis summary and an outlook for future research efforts that might address limitations of my research, pursue underexplored topics, and follow-up on promising directions in this domain.

2

The State of Single-Cell Atlas Data Visualization in the Biological Literature

2.1 Preface

Contents of this chapter have been adapted from "The State of Single-Cell Atlas Data Visualization in the Biological Literature," which is under review at the time of writ-

ing[229]. I conceptualized and led this project, performing data analysis and writing the manuscript. I thank multiple others who contributed to project discussions and editing of the manuscript: Eric Mörth, Thomas C. Smits, Simon Warchol, Grace Guo, Qianwen Wang, Robert Krueger, Hanspeter Pfister, and Nils Gehlenborg.

2.2 Introduction

Large-scale single-cell profiling efforts are underway with the aim to generate 'atlas' resources that catalog cellular archetypes including biomarkers and spatial locations. While the problem of visualization of cellular data is not new, the increasing size, resolution, and heterogeneity of single-cell atlas datasets presents challenges and opportunities for visualization researchers. In this chapter, we survey the usage of visualization to interpret single-cell atlas datasets by assessing over 1,800 figure panels from 45 biological publications. We intend for this survey to serve as a foundational resource for the visualization community as single-cell techniques and atlas-scale datasets are emerging rapidly with aims of advancing our understanding of biological function in health and disease. Here, we contribute the following:

- A systematic review of over 1,800 visualizations contained in figures within single-cell atlas publications from the biological literature.
- A research outlook, presenting challenges and opportunities at the intersection of visualization and single-cell biology.

2.2.1 Related Work

To the best of our knowledge, all prior reviews related to single-cell atlas data analysis have been written for an audience of biological researchers, and without a focus on

visualization.

Reviews and Perspectives from Biology and Medicine

In Chapter 1, we pointed to reviews and perspectives from the biomedical community that articulate goals for single-cell atlasing efforts (Table 1.1). These goals articulated by cell biologists can be viewed as user goals within the context of visual analytics. However, in these reviews and perspectives, the goals are often articulated at a high level such that translation to domain-agnostic visualization tasks remains difficult[251]. Nevertheless, they serve as primers on the direction of the field and useful reference points during the analysis of lower-level goals found within atlas publications and tasks facilitated by single-cell visualizations.

Visualization Surveys from Biology and Medicine

Two survey papers discuss the landscape of visualization tools for single-cell data. In 2020, Cakir *et al.* analyzed 13 interactive tools for single-cell transcriptomics data visualization[102]. In 2021, analysis of the scRNA-tools database[463] revealed that 437 of the 1,059 cataloged tools (41%) were annotated with the `visualization` category (and 89 (8%) were annotated as `interactive`). These two surveys are written for audiences of biologists & bioinformaticians and are limited in scope (to the transcriptomics modality). While Cakir *et al.* profiled tool performance (processing time and memory) for usage scenarios with between five thousand and two million cells, addressing questions of scalability, neither survey considers aspects of visualization design (*i.e.*, which plot types or visual encodings are available in different tools).

State of the Art Reports for Visualization Researchers

Within visualization, there are State of the Art Reports (STARs) on related topics under the umbrella of biology and medicine. The most closely related STARs are on live cell imaging[354] and connectomics[77], as these two domains deal with cellular-resolution data, but these areas differ in that they lack the high dimensionality and heterogeneity that single-cell atlases present. A recent STAR on visualization for physiology[165] contains sections on cellular-scale and tissue-scale data, but focuses on spatially- and temporally-resolved experimental modalities such as imaging (as opposed to sequencing-based), with the authors acknowledging, "We found comparatively few cell- and tissue-scale visualization works,"[165]. STARs for genomics[325] and molecular structures[242] are also closely related, but the primary entities analyzed in these domains are not cells, but rather sequences and atoms, respectively. There are no reviews focusing on the single-cell domain from a visualization research perspective, even though visualization is a key part of single-cell data analysis. The importance of this topic and the aforementioned challenges motivate a systematic review of current practices and the need to chart a path forward for visualization research applied to problems in the single-cell atlas domain.

Previous Calls to Action

Prior efforts have been made to highlight opportunities to apply visualization within the single-cell biology domain. I presented at an Application Spotlight at IEEE VIS 2023, "Visualization for spatial single-cell atlases: Identifying challenges and opportunities in complex biomedical data at multiple scales and modalities,"[12]. The 2021 and 2024 BioMedVis challenges were related to single-cell data visualization[13]. De-

spite these efforts, there remains a need to comprehensively survey the current state of visualization in the single-cell domain.

2.3 Survey of Static Figures from the Biological Literature

Single-cell atlas research results have primarily been communicated via publications containing static figures and plots. We conducted a systematic review of static figures that appear in single-cell atlas publications. Such figures represent the current state of visualization in the field, providing insight into the findings that authors consider most interesting or consequential to peers, the journal audience, and the scientific community. Through our contribution of this comprehensive and unbiased analysis, an audience of visualization researchers can be confident in relating current practices in the domain to prior best practices from visualization. We conducted this analysis in pursuit of answers to the following research questions (RQ):

- **RQ**: Which visualization designs are used in single-cell atlas publications?
- **RQ**: Which visualization designs are used only with certain data types, experimental techniques, or purposes?

2.3.1 Methodology

We used a keyword search to identify single-cell atlas publications and performed manual qualitative analysis of all subfigures within the main text of all publications meeting the inclusion criteria. We aimed to capture publications describing recent large-scale single-cell atlas efforts of human cells. We conducted a search of the Web of Science database using the advanced query string found in Listing 9. We filtered the query results based on the following eligibility criteria:

- EC1: Include only published 2020 to 2024
- EC2: Include only peer-reviewed (*i.e.*, exclude preprints)
- EC3: Exclude methods, benchmarking, and databases
- EC4: Exclude reviews, perspectives, and roadmaps
- EC5: Exclude non-human studies (*e.g.*, murine and primate)
- EC6: Exclude non-biology (*e.g.*, chemistry)
- EC7: Exclude duplicate entries
- EC8: Exclude cancer
- EC9: Exclude bulk (*i.e.*, non-single cell resolution)
- EC10: Exclude studies profiling less than 50,000 cells

The above eligibility criteria have the following rationale. Due to the recent nature of the single-cell field, studies published in the last five years (EC1) have generally profiled larger numbers of cells with more advanced experimental techniques (including spatial and multimodal). As the term "atlas" to describe single-cell data is not well-defined, we set the 50,000 cell threshold (EC9) to ensure that we captured publications which had to grapple with challenges of visualization scalability such as overplotting. The median number of cells profiled by studies published in 2022 was 50,000, thus this threshold roughly focuses our survey on the upper half of the study distribution during these years[403]. Because our database query was agnostic to paper types, we manually filtered to include only papers describing large atlas datasets (EC3, EC4) as we hypothesize that such papers are aiming to communicate insights

such as discovery of cell types or biomarkers, as opposed to communicating findings such as model performance (as would be expected within papers describing methods, benchmarking, or databases). We aimed to capture papers focusing on a single organism, human (EC5), as we expect that papers which focus on model organisms or cross-species comparison would introduce different types of analyses and therefore a different distribution of figure types (*e.g.*, related to evolutionary relationships). Similarly, we expect that studies related to cancer will include a very different set of figure types, for instance related to mutation (EC8). While we considered non-human and cancer atlases out of scope for this work, these are important topics for follow-up surveys. In line with our focus on single-cell data, we excluded all studies describing bulk (EC9) experiments (*i.e.*, non-single cell). To ensure a high-quality survey, we restricted our analysis to peer-reviewed publications (EC2) and filtered out duplicate records (EC7). While this eligibility criteria limits the scope of our analysis, follow-up surveys could expand their inclusion criteria to evaluate the generalization of our claims and our hypotheses of differing distributions of the space of visualization usage (*e.g.*, for cancer-related studies).

We performed the keyword search on August 12, 2024. An initial set of 221 matching database records was filtered down to 45 publications based on the eligibility criteria (Figure 2.1; Table 2.1). Eligibility was assessed based on metadata (for year and peer review criteria), title, abstract, and introduction. In the time since we performed this search (five months), re-running the search returns 34 additional records (an unknown number of which would pass the criteria), exemplifying the rapid pace at which the single-cell field is moving.

Next, we examined figures and sub-figures within each of the 45 publications. We performed qualitative analysis of each subfigure with the AtlasTI software using open

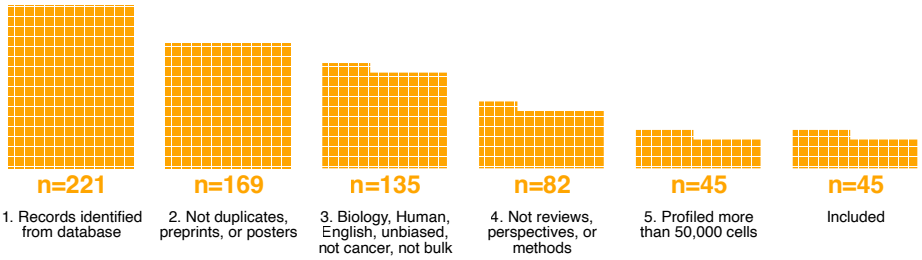

Figure 2.1: Methodology used for literature search, paper identification, and filtering. An initial set of 221 records was filtered down to 45 records matching the inclusion criteria. These included 45 papers contained 1,846 total sub-figures.

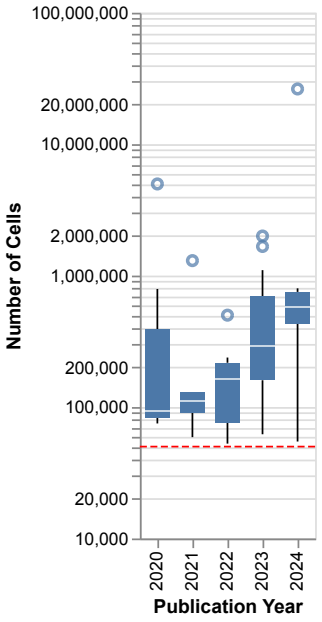

Figure 2.2: Distribution of cell count (log-scaled) per publication year for included atlas publications. Red dashed line denotes the 50,000 cell threshold for inclusion.

Table 2.1: Publications included in survey of single-cell atlas publication figures.

First Author	Year	# of Cells
Ament [56]	2023	168,719
Bocchi [82]	2021	96,789
Braun [92]	2023	1,665,937
Cao [103]	2020	4,979,593
Chen, Tao [115]	2024	54,658
Chen, Xinyue [116]	2024	26,300,000
Chen, Duo [111]	2023	62,063
Deprez [134]	2020	77,969
Domcke [140]	2020	790,957
Eze [154]	2021	59,000
Goh [173]	2023	169,798
Gray [177]	2022	52,681
He, Peng [193]	2022	71,752
He, Shuai [194]	2020	84,363
Hu [208]	2021	125,253
Jones [220]	2022	500,000
Kumar [245]	2023	714,331
Lai [249]	2024	387,000
Lake [250]	2023	400,000
Lee, Nicole Yee Shin [257]	2023	673,750
Lee, Woochan [258]	2023	95,170
Lengyel [261]	2022	146,538
Li [266]	2023	1,100,000
Liang [268]	2023	400,000
Massoni-Badosa [296]	2024	556,000
Monavarfeshan [309]	2023	151,000
Nicin [320]	2022	88,536
Orozco [327]	2020	100,055
Park [339]	2020	255,901
Reed [358]	2024	800,000
Su [401]	2022	224,464
Terekhova [410]	2023	2,000,000
Travaglini [415]	2020	75,000
van Zyl [419]	2022	195,248
Waelchli [429]	2024	606,380
Wang, Zeshuai [435]	2023	293,095
Wang, Zheng [436]	2021	130,000
Wesley [443]	2022	237,978
Williams [445]	2021	88,000
Winkler [446]	2022	181,388
Xu [450]	2023	180,000
Yan [452]	2020	85,000
Zhang, Bao [466]	2023	125,955
Zhang, Kai [467]	2021	1,300,000
Zhao [472]	2022	64,993

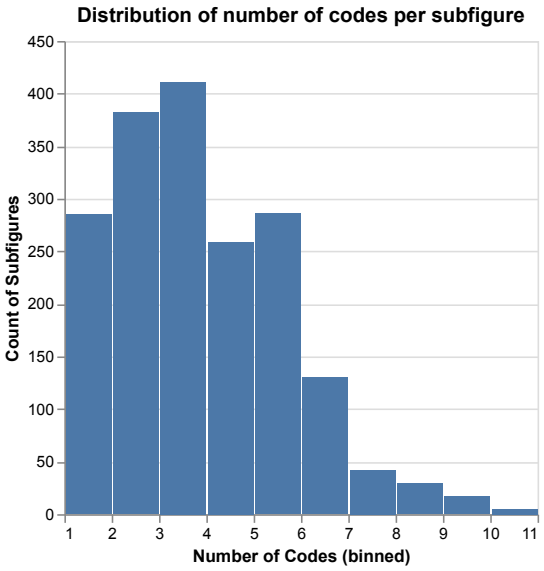

Figure 2.3: Histogram of codes per subfigure.

coding as described by Gibbs[168]. The resulting codes could be grouped into **plot types** (*e.g.*, "pie chart"), **visual properties** (*e.g.*, "highlighting"), **data types** (*e.g.*, "pseudotime"), **experimental techniques and modalities** (*e.g.*, "10x Visium" or "chromatin accessibility"), and **purpose** (*e.g.*, "comparison"). We define each code in Appendix F.4.

2.3.2 Results

We applied 85 codes to the 45 paper sources, across 1,846 subfigure quotations, for a total of 6,259 code applications. On average, each subfigure was annotated by 3.39 codes (min: 1, max: 11, median 3; Figure 2.3). Each unique code was applied to a median of 23 subfigures and mean of 73.63 (min: 1, max: 777). The distribution of subfigures per code was right-skewed, with 12 codes being applied to over 100 subfigures each (summing to 4,406 code applications) and the remaining 73 codes being

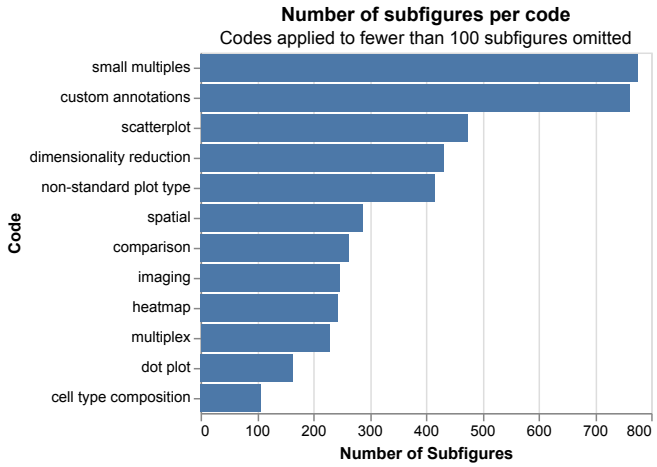

Figure 2.4: Top codes among all subfigures. Data only shown for codes applied to at least 100 subfigures.

applied to fewer than 100 subfigures each (summing to 1,853 code applications).

We first examined which codes were most-used among all subfigures. The top 12-applied codes were (from most to least): small multiples, custom annotations, scatterplot, dimensionality reduction, non-standard plot type, spatial, comparison, imaging, heatmap, multiplex, dot plot, and cell type composition (Figure 2.4).

We next examined how codes were used across the 45 publications. Limiting analysis to unique papers per code, the distribution was less skewed, with 19 codes each being applied to over half (more than 22) of the papers: scatterplot, small multiples, non-standard plot type, custom annotations, dimensionality reduction, heatmap, cell type composition, comparison, dot plot, spatial, stacked bar plot, imaging, experimental design, multiplex, violin plots, statistical significance, multi-scale, bar plot, gsea (Figure 2.5).

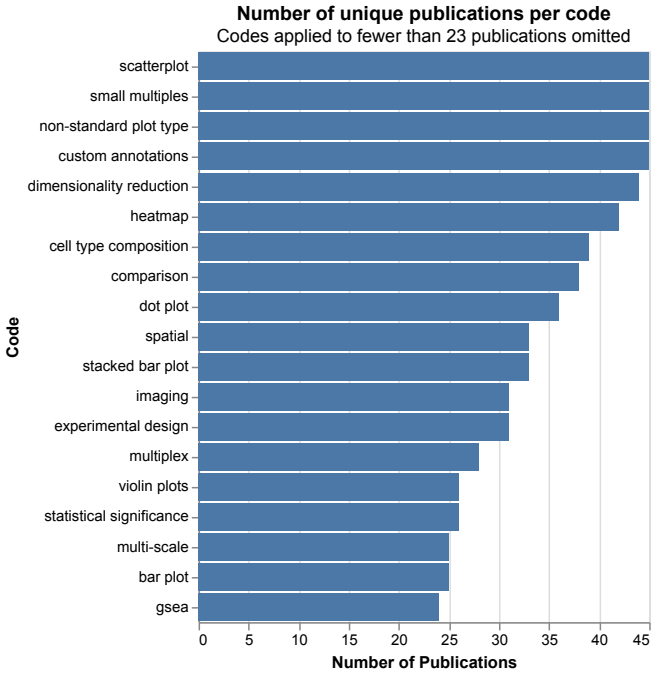

Figure 2.5: Top codes across publications. Data only shown for codes applied in more than half of the 45 publications.

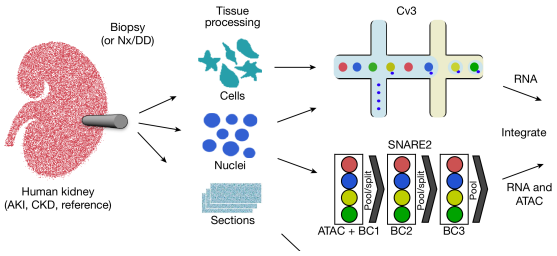

Figure 2.6: Example of code "experimental design" applied to subfigure 1a from Lake *et al.*[250]. Reprinted from an open-access publication.

Experimental design diagrams were used in the majority of publications

We applied the code "experimental design" to 43 subfigures (2.33% of subfigures) within 31 publications (68.89% of publications). These diagrams often appeared early within publications to communicate experimental strategies to readers, typically using a combination of flow charts, schematics, and illustrations (Figure 2.6). In some cases, these diagrams included some quantitative information such as numbers of samples per condition or embedded plots. We considered experimental design diagrams to be distinct from "anatomical diagram" subfigures, of which there were 41 subfigures (2.22%) from 22 publications (48.89%). Anatomical diagrams were often used to illustrate the organization and localization of cell types and larger functional tissue units within an organ, providing context for references to such entities in the text or in other figures (Figure 2.7).

Four codes appeared in all included publications

The following codes were applied to at least one subfigure in all 45 single-cell atlas publications: small multiples, custom annotations, scatterplot, and non-standard plot

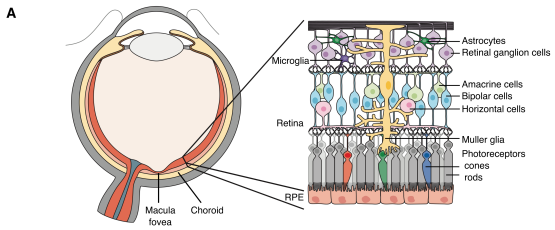

Figure 2.7: Example of code "anatomical diagram" applied to subfigure 1a from Orozco *et al.*[327]. Reprinted from an open-access publication.

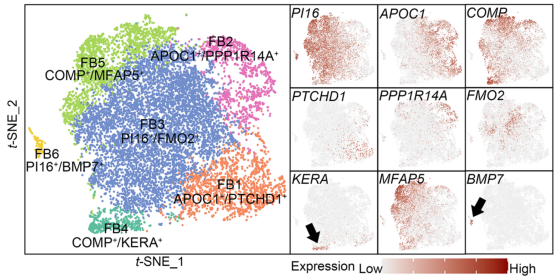

Figure 2.8: Example of code "small multiples" applied to subfigure 2a from Zhao *et al.*[472]. Reprinted from an open-access publication.

type (Figure 2.5). While we began with the previous section due to the typical early appearance of the subfigures within publications, the sections that follow are organized based on descending code occurrences.

Small multiples. "Small multiples" was the most-applied code, with 777 subfigure applications (42.09% of subfigures). We define "small multiples" as repeated instances of the same plot type, either within a single subfigure (Figures 2.8 and 2.10) or among subfigure instances within a parent figure (Figure 2.9). The top co-occurring codes were custom annotations (400 cooccurrences; 51.48% of primary occurrences), scatterplot (254; 32.69%), dimensionality reduction (230; 29.60%), spatial (229; 29.47%), and imaging (206; 26.51%).

Small multiples were often used to simultaneously visualize the distribution of

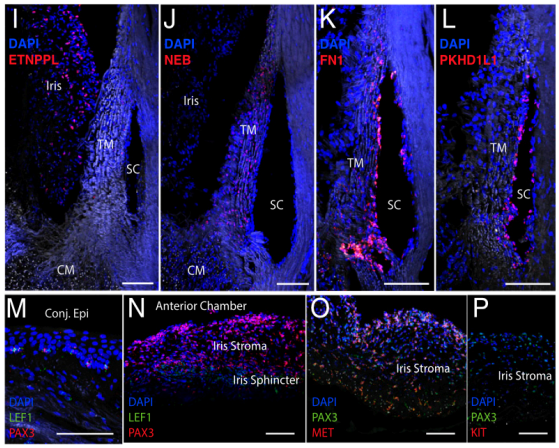

Figure 2.9: Example of code "small multiples" applied to multiple subfigures (5i-5p) from van Zyl *et al.*[419]. Reprinted from an open-access publication.

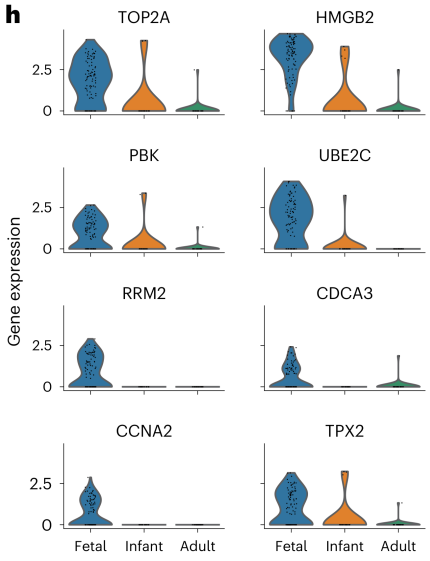

Figure 2.10: Example of code "small multiples" applied to subfigure 3h from Chen *et al.*[116]. Reprinted from an open-access publication.

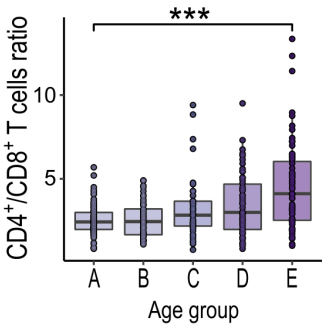

Figure 2.11: Example of code "non-standard plot type" applied to subfigure 1e from Terekhova *et al*.[410]. Reprinted from an open-access publication.

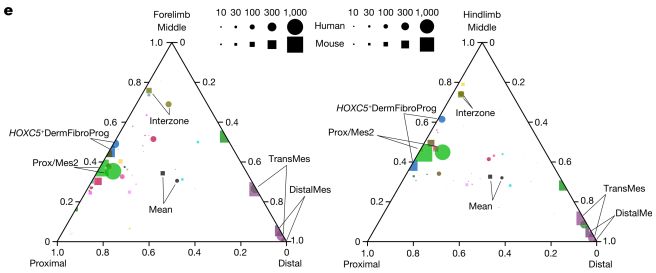

Figure 2.12: Example of code "non-standard plot type" applied to subfigure 6e from Zhang *et al*.[466]. Reprinted from an open-access publication.

abundance for a subset of biomarkers across all cells (Figure 2.8). A similar frequently-observed usage was to visualize multiple image channels (corresponding to different biomarkers), enabling reuse of colors for separate channels within the same imaging field of view and avoiding color blending issues (Figure 2.13).

Custom annotations. The code "custom annotations" was the second-most applied code. It was applied to 762 subfigures (41.28%), including at least one subfigure within all 45 analyzed publications. We define "custom annotations" as any non-standard annotation to a plot. This includes text and region annotations on scatterplots and images, metadata bar plots alongside heatmaps, and arrows indicating data

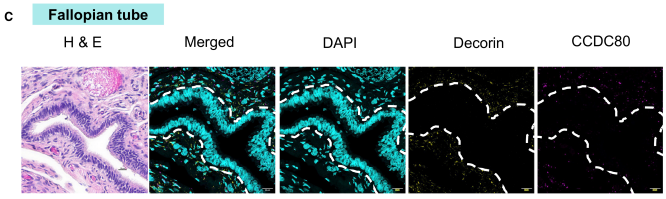

Figure 2.13: Example of code "custom annotations" applied to subfigure 3c from Lengyel *et al.*[261]. Reprinted from an open-access publication. Here, custom annotations take the form of outlined regions of interest.

points of interest. The top codes that co-occurred with this code were small multiples (400; 52.49% of primary occurrences; Figure 2.13), scatterplot (307; 40.29%; Figure 2.16), dimensionality reduction (278; 36.48%), spatial (216; 28.35%; Figure 2.13), and imaging (194; 25.46%).

Through further coding efforts, this code could be stratified, for instance into text-based annotations, polygon- or region-based annotations, arrow annotations, and metadata bar plot annotations. We observe that text annotations on scatterplots to indicate cell type categories are often used in lieu of a legend that would define the color per category (Figure 2.16). We hypothesize that custom annotations are widespread in static publication figures in order to communicate or reinforce the authors' intended takeaway message.

Scatterplot. Scatterplot usage was widespread, with this code applied to 474 subfigures (25.68%). While the majority of scatterplot subfigures displayed dimensionality reduction results (Section 2.3.2), other occurrences of scatterplots included gating scatterplots (23; Figure 2.14) and volcano plots (16; Figure 2.15).

Gating scatterplots facilitate identification of specific subpopulations of cells. The points correspond to cells and the X and Y axes correspond to two different biomarkers. Points are often colored to indicate point density, sometimes with density con-

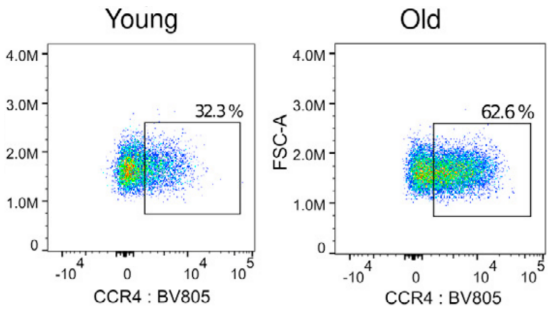

Figure 2.14: Example of code "gating scatterplot" applied to subfigure 3n from Terekhova *et al.*[410]. Reprinted from an open-access publication.

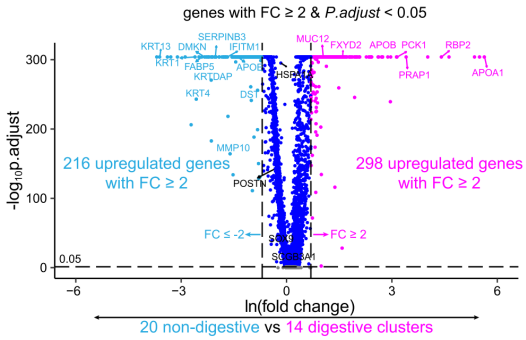

Figure 2.15: Example of code "volcano plot" applied to subfigure 5d from He *et al.*[194]. Reprinted from an open-access publication.

tours overlaid.

Volcano plots enable identification of differentially-expressed genes between two groups of cells (*e.g.*, case versus control). Scatterplot points correspond to genes (not cells) positioned according to the results of a statistical test called a differential expression test. Often, the most-significant points are annotated with gene names, offset in space to prevent overplotting. Plots sometimes include a horizontal rule to indicate a 0.05 p-value threshold and vertical rules to denote absolute fold-change thresholds.

Non-standard plot type. The code "non-standard plot type" indicates a plot type that either does not appear to correspond to one of the plotting functions avail-

able in the most widely used single-cell plotting toolkits (ScanPy[448,426,278] and Seurat[187]) or whose semantics were atypical among single-cell atlas publications. In some cases, these correspond to box or bar plots designed to communicate the level or statistical significance of a marker gene (or signature of multiple positive or negative markers) for a certain subpopulation of cells (Figure 2.11). In other cases, these correspond to visualization designs that did not appear within more than one atlas publication (Figure 2.12). This group of subfigures deserves follow-up study to identify effective visualization designs that would be amenable to reuse in future publications or tools. With a larger sample of subfigures (*i.e.*, from additional atlas publications) and further coding efforts, we believe that these subfigures can be stratified more finely and represent an interesting area for future analysis.

This code was applied to 416 subfigures (22.54%). The top ten co-occurring codes were small multiples (127 cooccurrences; 30.53% of primary occurrences), comparison (106; 25.48%), custom annotations (62; 14.90%), bar plot (49; 11.78%), heatmap (40; 9.62%), statistical significance (35; 8.41%), ligand-receptor (34; 8.17%), scatterplot (32; 7.69%), box plot (30; 7.21%), and cell type composition (25; 6.01%).

Dimensionality reduction scatterplots were used in almost all publications

The code "dimensionality reduction" was applied in 44 of the 45 publications to 432 subfigures (23.40%). Not all scatterplots displayed dimensionality reduction results, but almost all dimensionality reduction results were visualized using scatterplots (in two cases dimensionality reduction results were displayed using hexbinned scatterplots). As expected (based on guidelines in the domain), in the overwhelming majority of the observed dimensionality reduction scatterplots, the points correspond to

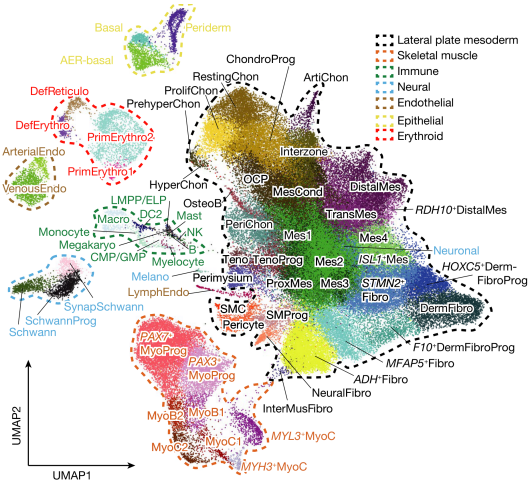

Figure 2.16: Example of code "custom annotations" applied to dimensionality reduction scatterplot subfigure 1b from Zhang *et al.*[466]. Reprinted from an open-access publication. Here, custom annotations take the form of text overlays, lines, and outlined regions which indicate coarse cell types.

cells (as opposed to other types of entities).

Often, we observed dimensionality reduction scatterplot visualizations that involved some form of data subsetting, including filtering and highlighting. We also observed many instances of subclustering (which we consider to be a special case of filtering), in which a new dimensionality reduction was performed on a subset of data points to obtain a new set of 2D coordinates for these points, then visualized in their own scatterplot.

Dimensionality reduction scatterplot points were often colored using a quantitative colormap (to visualize a feature of interest, as in Figure 2.8) or a categorical colormap (to visualize cell type categories, as in Figure 2.16). These scatterplots were often accompanied by custom annotations in the form of text overlays, region outlines, and trajectory arrows.

Heatmaps were used in almost all publications

The code "heatmap" was applied to 244 subfigures (13.22%) in 42 publications. Heatmaps were often used to encode the expression or abundance of a biomarker (often using z-scores) across many cells, with custom annotations to indicate cell categories. Special cases of heatmaps included "clustermaps" in which a dendrogram is aligned to one or both (vertical and horizontal) axes of the heatmap, indicating that the heatmap axis elements have been arranged based on hierarchical clustering results. As clustering only provides a partial leaf ordering, in many cases, such axes appeared to be ordered according to the optimal leaf ordering procedure of Bar-Joseph *et al.*[67]. Alternative heatmap axis orderings were observed, based on pseudotime or a physical axis (coded as "pseudotime" or "pseudoaxis" respectively). Another special case of heatmap included confusion matrix heatmaps used to demonstrate model performance, for instance for cell type prediction models (coded as "model performance"). In yet other cases, heatmaps were used to display patterns of cell-cell interaction or ligand-receptor interaction (both coded as "ligand-receptor") .

Cell type composition was a common goal, often via stacked bar plots

We applied the code "cell type composition" to 108 subfigures (5.85%) from 39 publications. These often (in 76 cases, 70.37%) used a stacked bar plot visualization design, often normalized such that bars sum to one (Figure 2.17), with colors indicating cell types and individual bars stratifying groups of cells, for example grouping by tissue region, donor, or condition. We observed a wide range of cell type granularity, with plots containing from two to tens of cell types (*i.e.*, colors). Alternative designs for cell type composition included small multiples of pie and donut charts. In certain

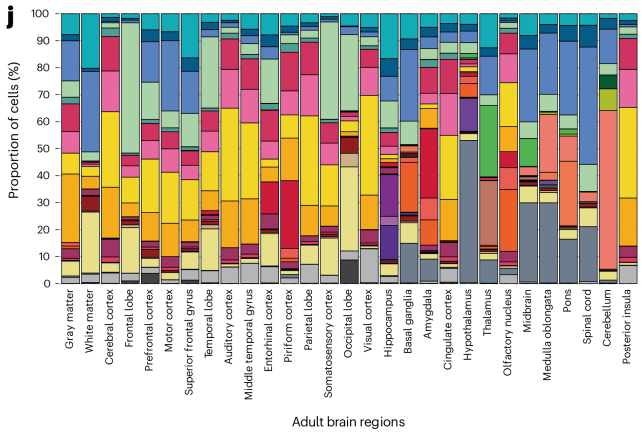

Figure 2.17: Example of code "cell type composition" applied to stacked bar plot subfigure 1j from Chen *et al.*[116]. Reprinted from an open-access publication.

cases, conditions were ordered (*e.g.*, developmental biology studies) such that designs like line and area plots could be used.

Comparison was a common goal, via a diversity of visualization designs

We applied the code "comparison" to 263 subfigures (14.25%) from 38 publications. We noticed three common types of comparisons: case-versus-control (two conditions), multiple spatial regions (two or more conditions), and ordered timepoints (*e.g.*, developmental stages). In many cases (144 instances, 54.75%), comparison was visualized using small multiples. In other cases, comparison was visualized using a within-plot condition axis, such as disease status or age (Figure 2.10).

We hypothesize that we have underestimated the extent to which comparison was an analysis goal in atlas subfigures, as our methodology can only capture explicitly-encoded comparisons. (Implicit comparative visualizations would include those in which the intended audience is expected to have innate understanding of a reference

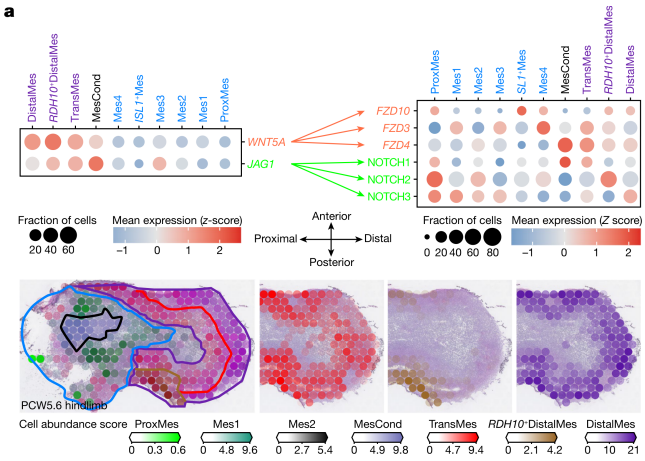

Figure 2.18: Example of code "dot plot" applied to a ligand-receptor interaction subfigure 5a from Zhang *et al.*[466]. Reprinted from an open-access publication.

condition, for instance pathologists with knowledge of healthy tissue appearance in histology images, such that the visualization only encodes a single non-reference condition.)

Dot plots were used in the majority of publications

We coded 164 occurrences of "dot plot" (8.88% of subfigures) in 36 publications. Dot plots use the radius and color channels of circular visual marks arranged in a grid to encode two quantitative values along two categorical dimensions. Typical categorical dimensions are biomarker and cell type. Dot color encodes a measure of biomarker abundance among members of the cell type, while radius encodes a proportion of members in which the biomarker level exceeds some minimum threshold of detection. Dot plots often organized their grid axes according to higher-level categories of cell types or biomarkers or based on hierarchical clustering results, similar to what we observed for heatmap axis organization.

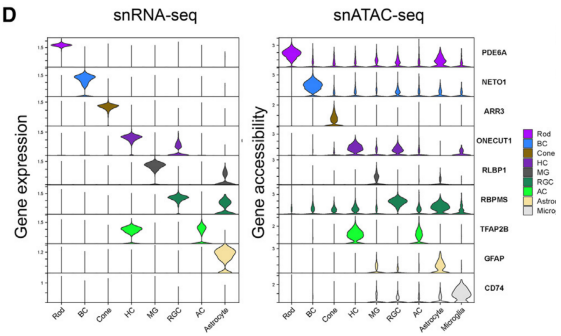

Figure 2.19: Example of code "multi-omics" applied to subfigure 1d from Liang *et al.*[268]. Reprinted from an open-access publication.

Other usages of dot plots included those in which the biomarker axis was exchanged for a biological pathway axis (based on the results of a gene set enrichment analysis, coded as "gsea") and those in which ligand-receptor interactions were visualized (coded as "ligand-receptor").

Violin plots were often used to compare gene expression distributions among cell classes

We applied the code "violin plots" to 75 subfigures (4.06%) from 26 publications. Subcode "violin plot matrix" was applied to 17 subfigures, in which a grid of violin plots share horizontal and vertical axes (similar to the grid of dot plots). Both types of subfigures were used to display distributions of biomarker abundance (*e.g.*, gene expression) for multiple cell groups and biomarkers simultaneously. In some cases, the cell group axis encodes cell types (Figure 2.19), while in others it encodes conditions (Figure 2.10).

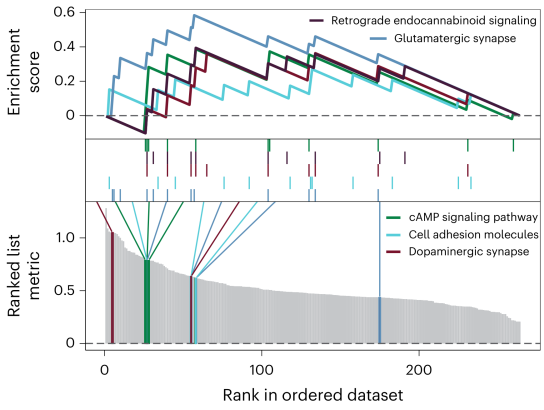

Figure 2.20: Example of code "gsea" applied to subfigure 5f from Chen *et al.*[116]. Reprinted from an open-access publication.

Plots displaying gene set enrichment analysis results were used in approximately half of the publications

We applied the code "gsea" to 68 subfigures (3.68%) from 24 publications. This code indicates a plot which encodes the results of a gene set enrichment analysis (GSEA)[402]. GSEA algorithms identify biological pathways or processes that are significantly associated with specific cell groups or conditions. Rather than analyzing genes individually, GSEA leverages predefined sets of genes from the biological literature, such as those involved in metabolism or immune response, to determine whether they are overrepresented among a ranked list of genes from an experiment (*e.g.*, those differentially-expressed between two conditions).

We identified multiple visualization designs displaying GSEA results. In some cases, such plots display the full ordered list of genes along with the value of the enrichment score at every position within the list (Figure 2.20). In other cases, enrichment scores themselves are directly plotted (Figures 2.21 and 2.22).

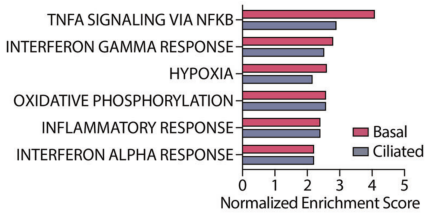

Figure 2.21: Example of code "gsea" applied to subfigure 4e from Lee *et al.*[258]. Reprinted from an open-access publication.

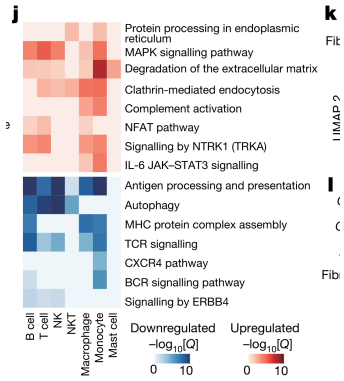

Figure 2.22: Example of code "gsea" applied to subfigure 4j from Lai *et al.*[249]. Reprinted from an open-access publication.

Spatial and multiplexed imaging visualizations were used in the majority of publications

Spatially-resolved visualizations (coded "spatial") appeared in 289 subfigures across 33 publications. This was a broad code, encompassing subcodes including "spatial transcriptomics" and "imaging" (itself including "multiplexed" and "histology" subcodes). Excluding its subcodes, the top co-occurring codes with "spatial" were small multiples (229; 79.24%), custom annotations (216; 74.74%), multi-scale (87; 30.10%), representative sample (41; 14.19%), and comparison (31; 10.73%).

Figure 2.13 is an example in which many of these codes were applied within a single subfigure: histology, multiplexed, custom annotations, small multiples, multi-scale, and representative sample. Other notable "spatial" subfigures include grids of pie charts (Figure 2.23) & cell abundance color scales for spot-based data (Figure 2.18, Bottom) and direct visualization of point-based data. We observed three instances of subfigures displaying 3D spatial data.

Patterns among multi-omics figures

We applied the code "multi-omics" to 31 subfigures (1.68%) from 10 publications. This code indicates a plot which simultaneously displays information from more than one experimental modality, such as both scRNA-seq and scATAC-seq. In most cases, experimental modalities were displayed using small multiples, for example of box plots (Figure 2.24), violin matrix plots (Figure 2.19) or dimensionality reduction scatterplots (Figure 2.25). In only a small number of cases were comparisons between modalities directly plotted (Figure 2.26).

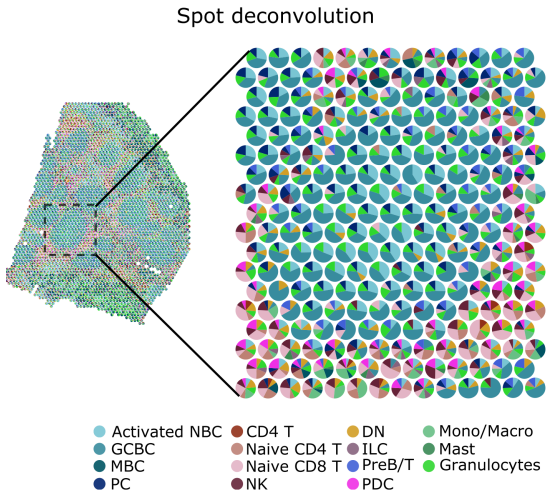

Figure 2.23: Example of code "deconvolution" applied to subfigure 1g from Massoni-Badosa *et al.*[296]. Reprinted from an open-access publication.

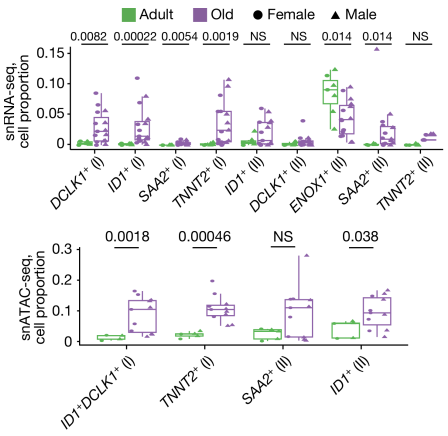

Figure 2.24: Example of code "multi-omics" applied to subfigure 2d from Lai *et al.*[249]. Reprinted from an open-access publication.

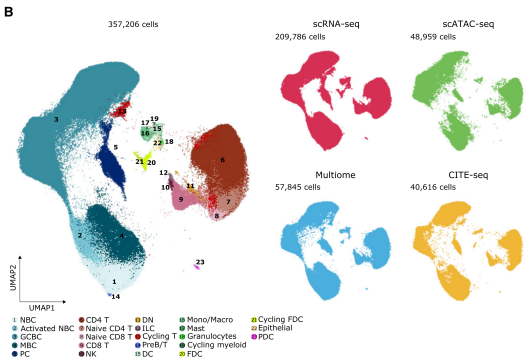

Figure 2.25: Example of code "multi-omics" applied to subfigure 1b from Massoni-Badosa *et al.*[296]. Reprinted from an open-access publication.

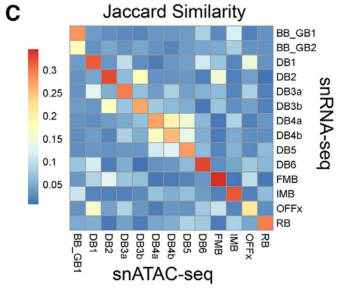

Figure 2.26: Example of code "multi-omics" applied to subfigure 2c from Liang *et al.*[268]. Reprinted from an open-access publication.

Patterns among cell-cell and ligand-receptor interaction figures

Cell-cell interaction refers to communication among cells, enabling coordinated tissue development and immune responses or other types of signaling. These interactions can occur through physical contact or through secreted molecules. Ligand-receptor interaction is a more specific mechanism where a ligand molecule (*e.g.*, hormone) binds to a receptor protein on a cell's surface (or inside). These interactions can be inferred or predicted from co-expression patterns, ligand-receptor knowledge bases, or from spatially-resolved experiments.

We used the code "ligand-receptor" to annotate both types of visualizations (cell-cell and ligand-receptor), and applied this code to 73 subfigures (3.95%) from 19 publications. We noticed a range of visualization designs, including node-link diagrams with circular node arrangements. We applied the subcode "ligand-receptor circle plot" to 21 of these from 9 publications. In these plots, nodes correspond to cell types and edges indicate interaction strengths between cell type pairs. We observed both undirected (Figure 2.27, Right) and directed (Figure 2.28, Left) edges.

Other visualization designs included scatterplots with points corresponding to cell types and the x- and y-axes indicating the strengths of incoming and outgoing interactions (Figure 2.28, Right). In other cases, visualizations directly plotted ligand-receptor interactions (rather than interactions between cell type pairs), for example with dot plots (Figure 2.18, Top) or sankey diagrams.

Patterns among pseudotime, pseudoaxis, and RNA velocity figures

Pseudotime inference algorithms order cells along a continuous trajectory (*e.g.*, based on their gene expression profiles). These trajectories may correspond to dynamic bi-

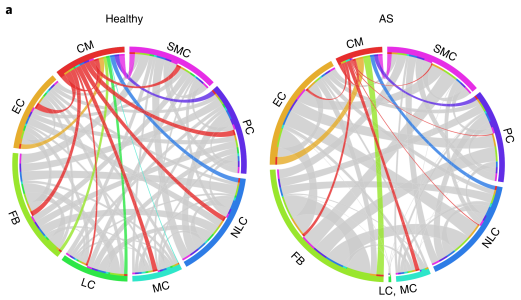

Figure 2.27: Example of code "ligand-receptor" applied to subfigure 2a from Nicin *et al.*[320]. Reprinted from an open-access publication.

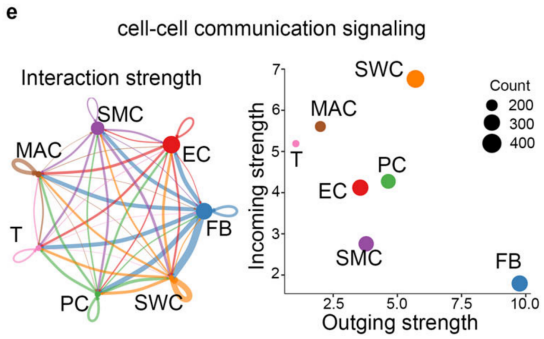

Figure 2.28: Example of code "ligand-receptor" applied to subfigure 1e from Zhao *et al.*[472]. Reprinted from an open-access publication.

ological processes such as differentiation or cell cycle progression. In contrast, RNA velocity methods predict the future transcriptional state of cells by analyzing spliced and unspliced RNA ratios[246]. RNA velocity algorithms return a vector field that indicates the direction and speed of cellular transitions. The two approaches are complementary for studying cellular processes and neither require temporally-resolved experimental techniques.

Many subfigures matched the related codes "pseudotime" (43), "pseudoaxis" (2), and "velocity" (19). Pseudotime trajectories were commonly plotted using arrows layered on top of dimensionality reduction scatterplots. Similarly, RNA velocity information was commonly projected into the dimensionality reduction space and plotted using many small arrows (streamlines) overlaid on the scatterplot to indicate cell state transitions. Other pseudotime visualization designs included heatmaps and line charts with pseudotime along the horizontal axis (Figure 2.29). Similar heatmap designs were used for "pseudoaxis" plots which aim to order cells along a spatial axis.

2.3.3 Survey Browser

We provide an interactive web-based survey browser that can be used to explore the subfigures further (Figure 2.30), available at https://keller-mark.github.io/sc-star-site/.

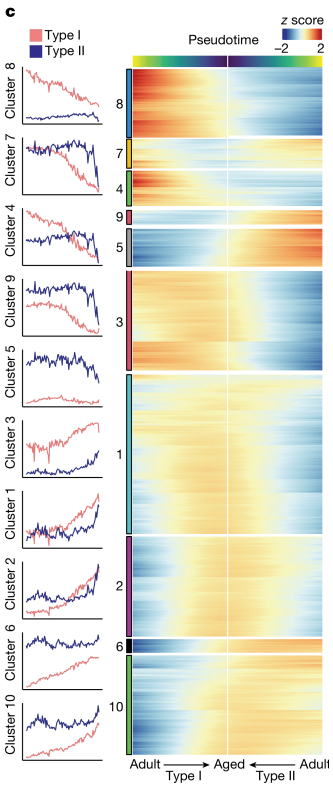

Figure 2.29: Example of code "pseudotime" applied to subfigure 3c from Lai *et al.*[249]. Reprinted from an open-access publication.

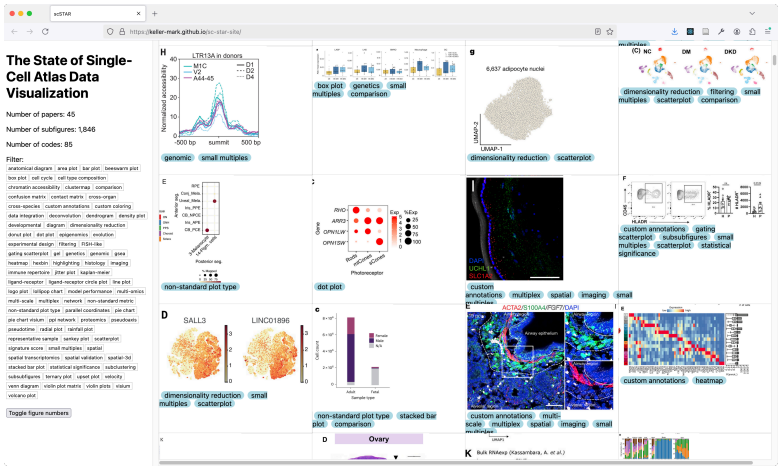

Figure 2.30: Screenshot of survey browser website.

2.4 Analysis of Plotting Toolkits

There are thousands of computational toolkits for single-cell data analysis, including visualization[463]. We characterize the types of visualizations that are supported by the two widely-used toolkits ScanPy[448,426,278] and Seurat[187]. The purpose of this toolkit analysis is to facilitate comparison with our publication subfigure analysis findings. We note that these two toolkits have primarily been designed for analysis of single-cell transcriptomics data.

2.4.1 Methodology

We analyze the real-world usage of ScanPy and Seurat in open-source projects via the GitHub Code Search API. We queried the GitHub API on December 18, 2024 for Python and R files, respectively, in which both the library was imported and a plotting function from the library was used. We manually annotated each plotting function based on a single shared set of domain-agnostic chart types to facilitate com-

parison and interpretation.

2.4.2 Results

The ScanPy and Seurat libraries contain a total of 67 functions for static plotting, with 38 from ScanPy and 29 from Seurat. Our API queries returned a total of 7,689 and 11,280 records for ScanPy and Seurat plotting functions, respectively. We find that for both libraries, the two most-used plotting functions correspond to dimensionality reduction scatterplots in which points represent observations (*i.e.*, cells) (Figure 2.31). Violin plot functions are the third- (for ScanPy) and fourth- (for Seurat) most-used plotting functions. Dot plot functions are eighth- (for ScanPy) and third- (for Seurat) most-used. Heatmap and spatial plotting functions are in the top 15 for both libraries. Scatterplots in which points do not represent cells (*e.g.*, points corresponding to principal components to plot a quantity per component) are ranked seventh (for ScanPy) and ninth (for Seurat). Interestingly, the `spatial` plotting function in ScanPy is the fourth-most used despite the Scverse community now recommending usage of the alternative Squidpy and SpatialData packages for analysis of spatial data[335,291].

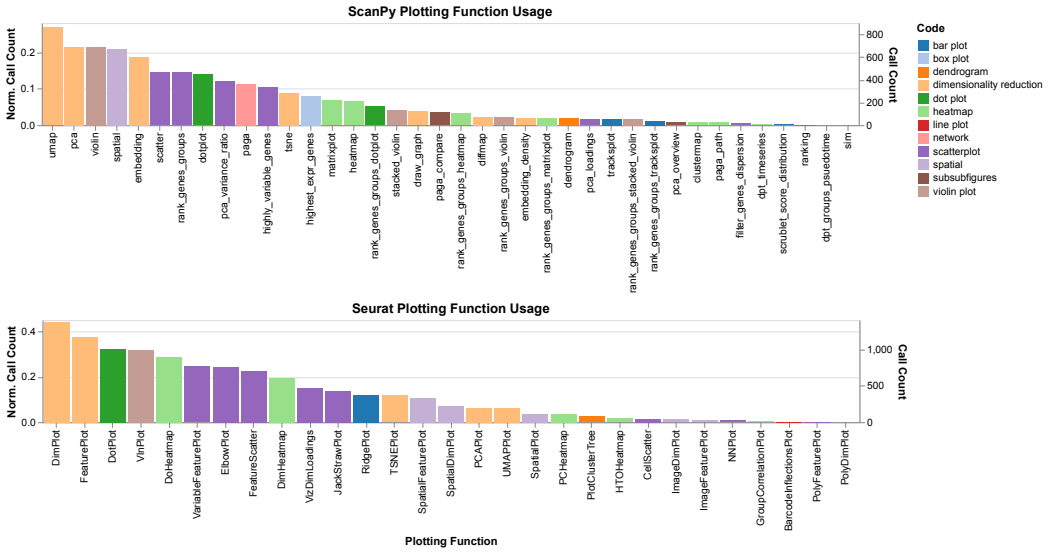

Figure 2.31: Plotting function usage statistics. Usage of plotting functions from ScanPy (Python) and Seurat (R) programmatic toolkits in open-source projects was calculated using the GitHub Code Search API on December 18, 2024. Normalized call counts represent the fraction of function call counts over the total count of library imports for the respective toolkits.

To facilitate comparison with our survey of static figures from atlas publications (Section 2.3), we mapped each plotting function to the closest code (Section F.4). After summing function calls by code, we find that the top most-called plotting functions (in both ScanPy and Seurat) correspond to the codes violin plot, dot plot, dimensionality reduction, spatial, and heatmap (Figure 2.32).

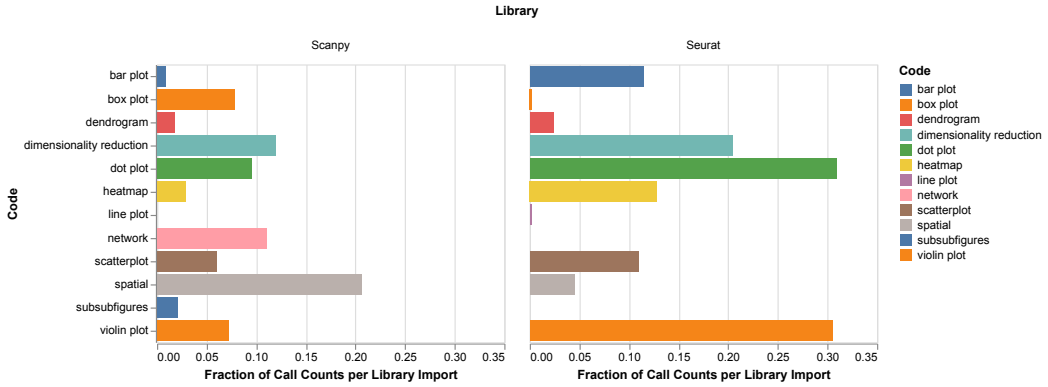

Figure 2.32: Plotting function usage by code.

2.4.3 Caveats

This analysis is limited to open-source code on GitHub and therefore does not consider code in private repositories or on other code hosting platforms. We normalized counts by total library imports to account for factors such as age and adoption of each library, but different normalization approaches could be used depending on the question of interest. In mapping plotting functions to codes, we used the closest code, but in some cases more than one code could conceivably be mapped to a plotting function (*e.g.*, the ScanPy `matrixplot` function could be considered both a heatmap and a dendrogram, but we mapped this to the code "heatmap"). These caveats should be considered when interpreting our analysis of the usage of plotting functions.

2.5 Discussion

In this section, we discuss considerations when interpreting our findings and areas for follow-up studies.

First, we look back to our initial research questions (Section 2.3). The results of our open-coding process (the codes themselves and code application patterns) suggest that the single-cell atlasing domain uses a relatively small set of visualization designs, with common ways to map from the data abstractions of the domain onto visual properties. Although it is challenging to quantify the extent to which the visualization designs within the analyzed subfigures might differ from designs in a broader context such as within biological or natural science publications, our qualitative results suggest this is the case. At the same time, the large number of usages of the "non-standard plot type" code suggest that publications often aim to communicate findings for which an established chart type usage pattern or data-abstraction-to-chart-type mapping does not yet exist within the domain.

Our focus on static figures found in single-cell atlas publications limits the scope of the conclusions to be drawn from this survey. We hypothesize that any analysis of publication figures, especially analyses limited to main-text figures, will be biased toward confirmatory (as opposed to exploratory) analysis goals, as publication figures are intended to communicate findings to peers in journal and reviewer audiences. Included figures will also reflect journal figure/paper size, style, and subfigure granularity requirements. Future analyses could consider supplementary figures (in addition to main-text figures) to potentially capture a broader set of visualization designs and analysis goals.

Analysis of plots within figures, while valuable, ignores the full context available within a paper. Often, captions and text references guide the audience to the analysis goal that motivated the author to include the figure in the paper. Therefore, future work should relate subfigures to their corresponding captions and text references to facilitate analysis of visualization tasks and goals, for instance, using the taxonomy of

Lam *et al.*[251], to correlate plot types with analysis tasks and investigate whether the visual design aligns with the task.

Our reliance on keyword-based paper identification, while systematic, misses important atlas papers with titles such as, "Cells of the human intestinal tract mapped across space and time," (missing "atlas")[149]. However, arbitrary inclusion of these publications would bias our survey and reduce its systematic nature. Future surveys may benefit from language model-based paper identification approaches to overcome limitations of direct title string matching[196].

Our paper inclusion criteria, while justified, also poses limitations. Specifically, our 50,000 cell threshold likely impacts experimental modalities differently. For instance, the relatively more recent spatially-resolved and multimodal experimental techniques, if less scalable, will be underrepresented in our included papers compared to more established techniques such as non-spatial transcriptomics. Future surveys should consider that experimental scalability is a moving target and should be considered separately for each experimental technique.

Because many single-cell papers use the same set of underlying software toolkits for not only data analysis but also plotting and visualization, our results are intertwined with the set of plot types and visual encodings available in these toolkits (and demonstrated within best practices guidelines and tutorials from the domain). On the other hand, the open-source nature of these toolkits might suggest that sufficient demand for particular visualization designs would result in the implementation of new plotting functions in software toolkits and their associated tutorials. Further studies are needed to understand the perspectives of single-cell researchers on the topic of visualization as it is currently applied within atlasing efforts.

2.5.1 Development of Taxonomy

Taxonomic organization of visualization techniques and tasks have proven powerful both in the abstract[95] and within domains[325]. We discuss how the results of our analysis may inform the future development of a taxonomy of visualization design for the single-cell atlas domain.

Our findings of visualization design patterns indicate that a domain-specific taxonomy is warranted. Due to the abundance of figures which involve comparison, such a taxonomy could draw upon Gleicher's Considerations for Visualizing Comparison[171]. We observed many cases of two comparative designs outlined: juxtaposition (small multiples) and superposition (layering), yet fewer cases of explicit-encoding designs. Defining comparative elements in domain terms (*e.g.*, cells, genes, pathways) and outlining domain-specific scalability strategies (*e.g.*, cell types, genomic binning) would be a valuable contribution of such work. Other considerations for a taxonomy include patterns of visualizations for data distributions and patterns suggesting communicative intent.

We identified a rich set of visualizations with the goal of communicating one or more data distributions, which can be considered a special case of comparison. These distribution visualizations can be categorized using comparative scalability strategies, such as summarize (*e.g.*, violin plots) or scan sequentially (*e.g.*, jitter plots or scatterplots), yet also bring their own considerations such as colormaps and density encodings.

Patterns of communicative intent included those aiming to contextualize findings, through techniques such as highlighting and filtering data points, inclusion of companion diagrams (*e.g.*, anatomical), and multi-scale (overview+detail) visualizations.

Explicit types of annotations included text, arrows, pointers, and regions. A taxonomy informed by these considerations would outline a broad space of visualization approaches for single-cell researchers to explore, and would facilitate evaluation of different approaches with respect to analysis goals.

2.6 Research Outlook

Our report highlights several open challenges and opportunities for the visualization community.

Our findings provide the community with a reference point for current practices in the single-cell domain. Using this reference, visualization researchers can assist the domain in grappling with its data complexity through contribution of visualization approaches that closely match the intended analysis goals, both high-level (articulated in reviews and perspective pieces) and low-level (articulated in figure captions and text references). While there may be tradeoffs between established domain practices and visualization best practices, we anticipate embracement of contributed techniques which advance domain goals and are made usable and interoperable via established data structures and plotting toolkits.

A comprehensive survey of the landscape of *interactive* visualization tools for single-cell data, including visualizations integrated into atlas data portals, remains an important area for future work. Not only may such a survey expand our understanding of the visualization design space and capture a broader set of analysis goals, it could facilitate the formalization of interactivity techniques used in the single-cell atlas domain into a taxonomy. In the meantime, our findings based on static figures may be considered alongside existing general taxonomies of interactivity from the vi-

sualization literature[95,113,390,457]. Our publication figure-based findings may facilitate the development of interactive tools which offer seamless transition between exploration, data analysis, and creation of publication-ready figures. Our identification of widespread usage of visualization annotation techniques suggest an opportunity to contribute automated methods for derivation of annotations, including for spatial and imaging data. Finally, there is an opportunity to apply components of visualization taxonomies, both existing and future (Section 2.5.1), to identify additional gaps and chart further opportunities.

This is a quickly-moving domain and will likely benefit from re-analysis over time. While the publications we analyzed did include several multimodal and epigenomics studies, we did not identify meaningful patterns related to these types of subfigures. Similarly, regarding spatially-resolved studies, we primarily identified 2D multiplexed imaging subfigures despite a proliferation of spatial modalities and enthusiasm around 3D spatial biology. We anticipate the data and methodology that we contribute to be applicable to future analyses that better capture these types of studies.

2.7 Conclusion

This chapter provides a unique and timely perspective into how visualization is currently being used within the single-cell domain. In contrast to surveys which restrict their analysis to research published in visualization venues, we survey the state of visualization in the target domain using scientific publications as a lens. On the one hand, this means analysis of visualizations that have not necessarily been published for their visualization-related merit, but on the other hand, analysis of "real-world" practices has the potential to teach us about barriers to adoption of visualization best

practices or to prompt their revision. By providing a survey of visualization usage and an overview of current challenges and opportunities, this survey serves as a useful launchpad from which an audience of visualization researchers might begin to contribute at the intersection of the two domains and offers biological researchers a new perspective on visualization.

3

Interactive visual analytics framework for single-cell data

3.1 Preface

Portions of this chapter have been adapted from a previous publication:

- Vitessce: integrative visualization of multimodal and spatially-resolved single-

cell data[228]

I was the primary author of the Vitessce manuscript, with contributions from Ilan Gold, Chuck McCallum, Trevor Manz, Peter V. Kharchenko, and Nils Gehlenborg. Chuck McCallum implemented the first version of the software which we subsequently built upon. I thank the many individuals who have contributed to Vitessce over the years, not only via code but also via bug reports and feature requests, including lab members Ilan Gold, Chuck McCallum, Trevor Manz, Tabassum Kakar, Eric Mörth, Nikolay Akhmetov, John Conroy, Tos Chan, Jennifer Marx, Sehi L'Yi, Thomas Smits, Alexandra Irger, and Morgan Turner.

3.2 Abstract

Multi-omics technologies with single-cell and spatial resolution make it possible to measure thousands of features across millions of cells. However, visual analysis of high-dimensional transcriptomic, proteomic, genome-mapped, and imaging data types simultaneously remains a challenge. Here, we describe Vitessce, an interactive web-based visualization framework for exploration of multimodal and spatially-resolved single-cell data. We demonstrate integrative visualization of millions of data points including cell type annotations, gene expression quantities, spatially-resolved transcripts, and cell segmentations across multiple coordinated views. The open source software is available at http://vitessce.io.

3.3 Introduction

Techniques for characterizing tissues and organs at single-cell resolution have recently overcome the challenges of spatial localization and measurement of multiple modalities, improving our ability to distinguish cell types, cell states, and cell neighborhoods[462]. The high dimensionality of single-cell omics and imaging data has catalyzed the development of computational models designed to annotate cellular observations with segmentations, cluster memberships, and predicted cell types. Visual analysis of multimodal and spatially-resolved single-cell datasets which integrate genomics, transcriptomics, and imaging modalities can facilitate quality control, communication of results, identification of biomarkers, and generation of hypotheses.

Despite experimental and computational advances in the field, there remains a lack of tools for interactive visualization of multimodal and spatial single-cell datasets. The lack of sufficient tools for visual analysis of heterogeneous biological datasets, termed "the challenge of ultra-scale integrative visualomics," has been highlighted by the visualization research community[374,165]. Visualization of multimodal single-cell datasets currently requires multiple disconnected tools such as image viewers and genome browsers which support individual data modalities. Tools that facilitate discovery of connections across modalities will enable multimodal experimental data to be used to its full potential to improve understandings of health, disease, and regulatory mechanisms[475].

3.4 Related Work

Recent tools for visualization of single-cell data have focused on interactivity and scalability. Tools such as Cellxgene[303], Cirrocumulus[263], and Pagoda2[232] support in-

teractive exploration of datasets containing millions of cells and thousands of genes using scatterplots and heatmaps. However, these tools are designed for visual analysis of transcriptomics data. As a result, they lack support for spatial data types such as cell segmentations or genome-mapped modalities such as chromatin accessibility profiles. On the other hand, tools designed for spatial data such as TissUUmaps[351] and Napari-SpatialData[291] do not support non-spatial or genome-mapped data visualization. The focus to date on single-modality data has resulted in tools that do not support visualization tasks such as comparison and relation across modalities and in spatial context[386].

3.5 Vitessce: integrative visualization of multimodal and spatially-resolved single-cell data

We developed Vitessce to overcome these limitations and enable integrative visualization of multimodal and spatial single-cell data. Vitessce is a scalable web-based data visualization framework which supports simultaneous visual exploration of transcriptomics, proteomics, genome-mapped, and imaging modalities (Figure 3.1). For example, the software provides modules based on Viv[288] and HiGlass[231] toolkits for visualization of multi-scale multiplexed imaging data and genome-mapped data, respectively (Figure 3.2). A strength of Vitessce that distinguishes it from prior work is the ability to deploy visualizations of spatial and single-cell data in a wide range of contexts and compute infrastructures, including in static websites, web applications, data portals, Jupyter Notebooks, RStudio, and R Shiny apps.

Data from technologies such as scRNA-seq[286] and scATAC-seq[98], as well as multimodal and spatial technologies such as CITE-seq[398], CODEX[80], Slide-seq[364], 10x

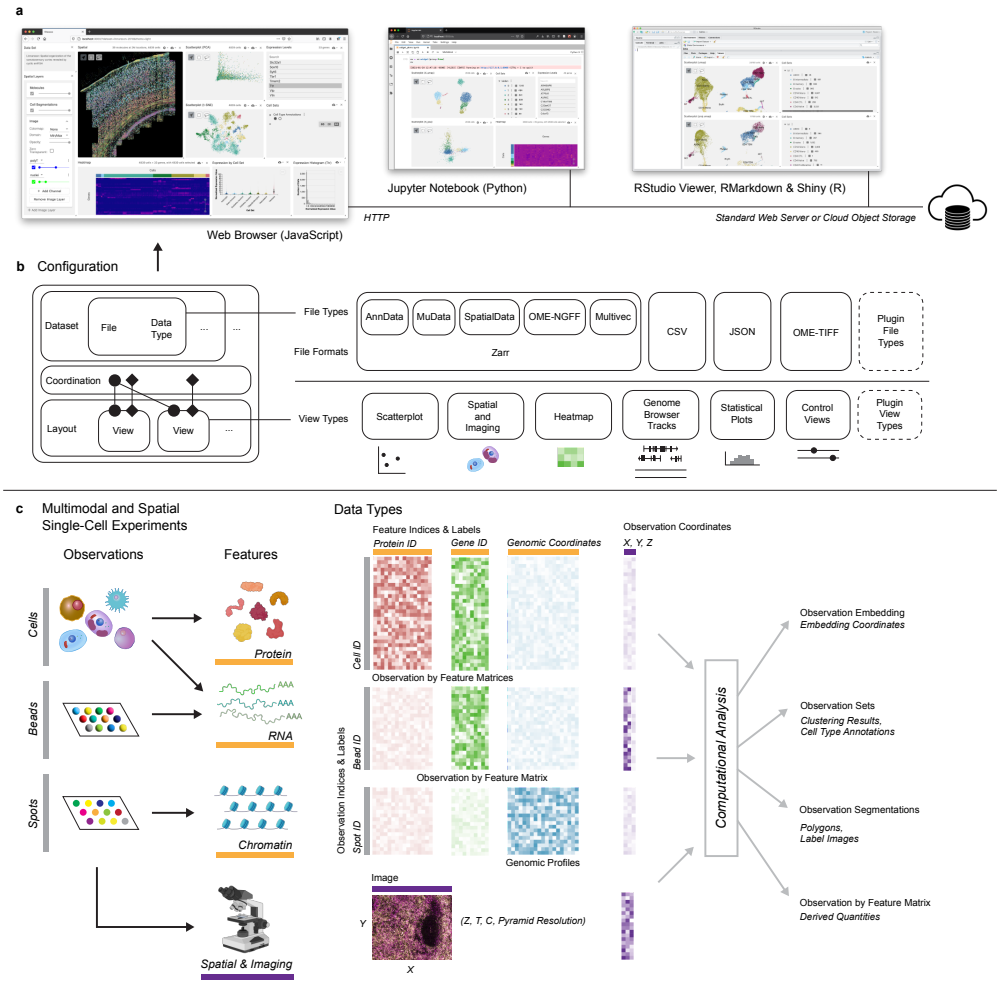

Figure 3.1: Vitessce can be used in multiple settings and can be configured to visualize raw and derived measurements from multimodal and spatial single-cell experiments. (a) Vitessce can be used as a JavaScript component in a web browser or a widget in Python and R analysis environments. (b) Single-cell, single-molecule and microscopy data stored in multiple formats can be visualized in multiple types of views (that is, interactive visualizations). (c) The modular design of Vitessce enables integrative visualization of multimodal and spatial single-cell experiments alongside computational analysis results. The arrows between observations and features represent the ability to visualize data from heterogeneous experiments that measure subsets of features that are shared by subsets of observations. OME, open microscopy environment; NGFF, next generation file format; CSV, comma separated values; JSON, JavaScript object notation; TIFF, tagged image file format; ID, identifier. Icon credits: cell images under 'Observations' from Reactome.org under a Creative Commons license CC BY 4.0; images under 'Features' from BioRender.

Genomics Visium and Multiome can be visualized with Vitessce. Data (pre)processing (*e.g.*, dimensionality reduction and cell segmentation) must be performed outside of Vitessce or through plugins[119], decoupling visualizations from bioinformatic algorithms. Vitessce can access files stored on static web servers and cloud object storage, supporting AnnData[448,427], MuData[94], SpatialData[291], CSV, OME-TIFF[270,76], and OME-Zarr[311,313] formats.

Given the rapidly expanding application of single-cell and spatial assays, the Vitessce framework is designed to reach a large audience of visualization producers and consumers. This audience includes single-cell data analysts intending to communicate results with collaborators, readers of publications who are interested in questions beyond those answered by static figures, and consumers of single-cell databases interested in checking data quality prior to follow-up studies. The framework addresses five key challenges to enable a wide audience to construct, deploy, and access visualizations of multimodal and spatially-resolved single-cell data:

Tailor visualizations to problem-specific data and biological questions. Depending on the experimental approach, a single-cell dataset may contain one or more modalities, with or without spatial coordinates and images (Figure 3.1c). Downstream analysis approaches vary widely and result in derived data types such as dimensionality reductions, clusterings, cell type annotations, and cell segmentations[464]. Heterogeneity in data types and in the types of questions asked by single-cell biologists require different visual representations and interactions, for example moving between expression and physical space. For multimodal datasets, analysts can use Vitessce to visualize modalities independently (*i.e.*, different modalities in different plots) or to visualize the pre-computed results of an integration method that projects

multiple modalities into a shared low-dimensional space.

Vitessce enables visual analysis of a patchwork of data types and supports multiple types of questions by providing a collection of interactive views that can be linked and coordinated using a modular approach. These views enable visualization of datasets containing highly-multiplexed images, cell and organelle segmentations, cell-by-gene expression matrices, cell-by-peak count matrices, cluster-level chromatin accessibility profiles, flat and hierarchical cell type annotations, and dimensionality reductions. Multiple views can be arranged in a grid layout that allows simultaneous visualization of data from multiple experiments. The views in Vitessce are designed to scale to large datasets. For example, the scatterplots can display millions of cells and the heatmaps can display tens of thousands of features (*e.g.*, genes, proteins, or chromatin accessibility peaks) by using WebGL and DeckGL[433]. Viv[288], HiGlass[231], and Vega-Lite[378] are used to support visualization of multi-gigabyte images, genome-wide measurements, and basic statistical analyses, respectively. We provide documentation to facilitate development of additional view types as plugins (Appendix G).

Integrate and explore multimodal data with coordinated multiple views. Visual analysis of multimodal and spatial single-cell data involves identification and relation of patterns that span modalities and data types. The current ecosystem of disparate tools such as genome, scatterplot, and image browsers makes such tasks difficult. Vitessce addresses the challenge of relation across modalities by supporting coordinated multiple views[88] (Appendix A), enabling interactions such as selection of genes and cell types to be reflected in multiple visualizations (Figure 3.1b). Visualization producers can configure any subset of views to be linked on any subset of over fifty parameters, including visualization properties and user preferences. Changes in

coordinated values are reflected in configuration updates, enabling snapshots of state to be captured. Configurations can be saved and shared via URL, making the interactive visualization reproducible and trackable[128].

Explore visualizations in different computational environments. Visual inspection of data is crucial to single-cell data analysis[278]. Vitessce can be integrated directly into not only web applications, but also exploratory computing environments where data analysis is performed. In Jupyter Notebooks and RMarkdown, Vitessce can be used as an interactive widget to visualize analysis results alongside the corresponding code (Figure 3.1b). Vitessce R and Python widgets are accompanied by object-oriented application programming interfaces (APIs) to simplify configuration. Widgets can be configured with local or remote data, enabling visualization of datasets that may be located in the cloud or on a compute cluster. These APIs are compatible with the data structures used by popular single-cell data analysis packages, including Scanpy[448,427], SnapATAC[157], and Seurat[377]. Export functions are included in the APIs to simplify remote data hosting.

Deploy and share interactive visualizations. Interactive visualizations can be more dynamic, scalable, shareable, and easier to deploy when they do not rely on a specialized server, as Manz *et al.*[288] demonstrate for multiplexed bioimaging visualizations. Implemented as a client-side web application without server-side requirements, Vitessce loads data from static files stored on standard web servers or cloud object storage systems including Google Cloud Storage and Amazon Web Services S3. Data loading, transformation, and rendering processes in Vitessce are performed directly in the web browser. While the JavaScript framework itself does not rely on a specialized

server, integrations with server-side computing infrastructure can be achieved through the development of plugin views which rely on such infrastructure internally. This is demonstrated by Cheng *et al.*[119] (Section 4.4) in plugin views that communicate with a specialized server for data integration in a human-in-the-loop setting.

Access data from multiple file formats. The community lacks consensus file formats for single-cell measurements and microscopy images. This can be attributed in part to the existence of proprietary file formats, an ecosystem fragmented along programming language boundaries, and the nature of emerging experimental methods whose outputs do not conform to existing formats. While minimum information standards for metadata have been proposed[381,161], several file formats are commonly used to store single-cell and imaging data in the analysis ecosystem[427,377]. To address this challenge, Vitessce abstracts data loading from visualization rendering, enabling support for multiple file formats, including CSV, AnnData[448,427], MuData[94], SpatialData[291], Multivec[231], JSON, OME-TIFF[76], and OME-Zarr[311,313]. Data stored in novel file formats can also be visualized in Vitessce through the development of plugin data loading modules (Appendix G).

Vitessce enables visual exploration of single-cell experiments which span the modalities of transcriptomics, epigenomics, proteomics, and imaging within a single integrative tool. We demonstrate the construction, deployment, and access of visualizations through use cases in Figure 3.2 and Extended Data Figures G.1-G.7.

In a CITE-seq use case, Vitessce can be used to validate the presence of cell types characterized by markers in both RNA and protein modalities. Stoeckius *et al.*[398] measure gene expression and surface protein abundance in a sample of human cord blood mononuclear cells (CBMCs), investigating well-characterized immune cell type

markers in these modalities. The findings of Stoeckius *et al.* can be reproduced by visualizing the dataset in Vitessce using linked scatterplots and heatmaps to explore protein abundance and gene expression levels simultaneously (Extended Data Figure G.5). For instance, we observe that natural killer (NK) cells can be identified based on CD56 protein levels and the expression of genes *GZMB*, *GZMK*, and *PRF1*, as reported by the authors[398].

A second use case demonstrates visualization of spatially-resolved gene expression data from an smFISH experiment. In a mouse brain sample, Codeluppi *et al.*[122] discover a transition region between the Pyramidal L2/3 and L4 neuronal cell types which contains Pyramidal L3/4 cells. This finding can be reproduced in Vitessce using the spatial plot and heatmap views (Figure 3.2a). In Vitessce, the colors of cell segmentations and scatterplot points can be mapped to gene expression values. Through interactive selection of the genes *Lamp5* and *Rorb*, we observe the reported co-expression pattern that is specific to the Pyramidal L3/4 cells[122].

The development of Vitessce has been propelled by collaboration with cell atlas consortia. Usage of the framework in the data portals for HuBMAP[391] and KPMP[132], as well as its open development on GitHub, has led to valuable feedback that has informed the requirements and future research directions. Decoupled from any particular database, Vitessce can be integrated as a visualization frontend into projects with varying backend data organization strategies. The modularity and extensibility of the framework allows for supporting additional visualization types, file formats, and single-cell modalities as they emerge. We anticipate that the community will embrace these characteristics by contributing plugin modules and incorporating Vitessce into applications that provide affordances for specific audiences, tasks, and technologies.

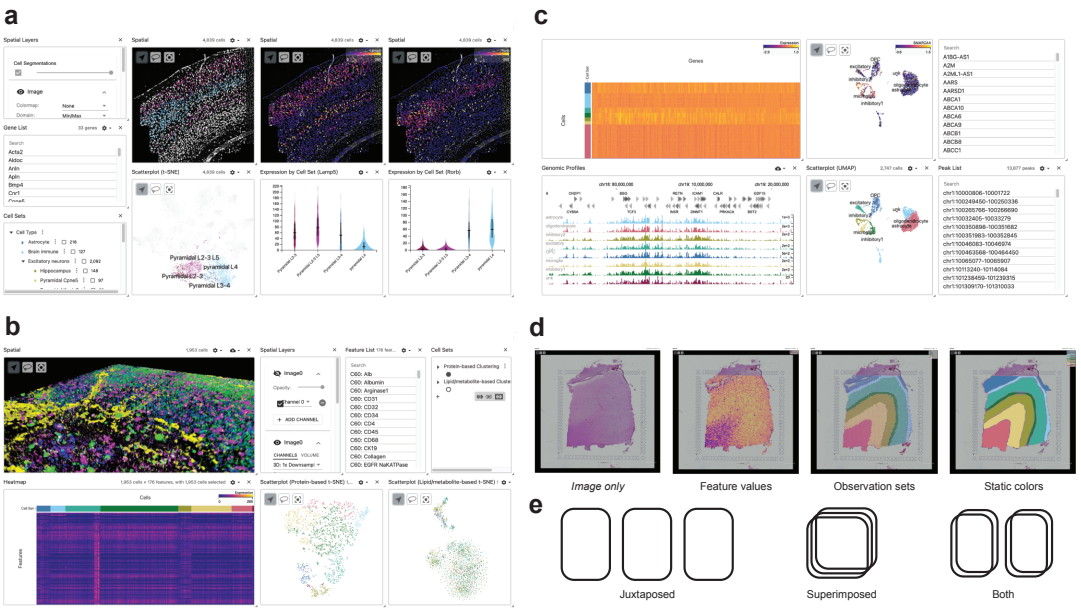

Figure 3.2: Sample use cases for Vitessce. (a) Visualization of a single-molecule FISH experiment containing multiplexed images, spatially resolved RNA molecules and cell segmentations[122]. (b) Volumetric rendering of a three-dimensional multimodal imaging mass spectrometry (IMS) dataset alongside a heatmap and side-by-side protein-based and lipid/metabolite-based dimensionality reduction scatterplots[411]. (c) Simultaneous visualization of gene expression and chromatin accessibility from a 10x Genomics Multiome dataset. (d) Schematic of supported visualization layers in the spatial view of Vitessce, namely, points, spots, segmentations and images[297]. Spot and segmentation layers can be colored by feature values, set membership or static colors. (e) Vitessce supports both juxtaposed and superimposed arrangements of multiple spatially resolved visualizations. Chr, chromosome; t-SNE, t-distributed stochastic neighbor embedding; UMAP, uniform manifold approximation and projection.

3.5.1 Methods

Further method details can be found in Appendix G. This appendix includes additional figures to demonstrate use cases for Vitessce, as well as software architecture and implementation details. While Appendix G contains details about the implementation of coordinated multiple views (CMV) in Vitessce, Appendix A extends upon this to describe a general framework called Use-Coordination which we designed for implementation of CMV in other visual analytics systems, informed by our work on Vitessce.

3.6 Discussion

As a flexible framework, Vitessce serves as an important foundation upon which many tools for exploration and communication of single-cell multimodal and spatial data can be built. Subsequent chapters in this thesis focus on particular applications which build upon Vitessce for comparative tasks (Chapter 4), for spatially-resolved data (Chapter 5), and for epigenomics data (Chapter 6). Yet beyond my work and beyond our lab, Vitessce is already being used in other open-source and commercial projects[45]. As of April 12, 2025, we are aware of integration of Vitessce in at least eight data portals, seven data processing pipelines, three publication-associated websites, and two commercial software products. Consideration of all of these applications together suggests trends in the usage of Vitessce which could inform future directions.

First, the quick uptake of Vitessce in multiple contexts suggests that our work to address key challenges in construction and deployment of visualizations has been successful. We largely attribute this to the lack of specialized server requirements, de-

coupling from any particular backend data source, and support for domain-standard file formats and data structures. The aforementioned types of usage (data portals, pipelines, publication websites, and commercial software) primarily involve one or more software developers who have integrated Vitessce into another piece of software. We have less insight into how end-users who are bioinformatics practitioners or computational biologists construct Vitessce visualizations. While we can identify some such usages via the GitHub Code Search feature (*e.g.*, searching for imports of the Vitessce Python package in Jupyter notebook files), this methodology will not capture usage outside of GitHub (*e.g.*, saved only locally, saved in the HuBMAP Workspaces platform, or configured directly in the Vitessce online editor). It remains important future work to understand how different user audiences configure Vitessce visualizations and the challenges they face. LY'i *et al.* find that in the related area of genomics, the visualization authoring experience is enhanced by blending multiple types of interfaces: code-, natural language-, shelf-, and template-based[283]. It remains an open question whether such a blended interface would enhance the authoring process for Vitessce, in particular because the Vitessce configuration revolves around plot types as opposed to a more expressive grammar of graphics-based specification[281] (although we have taken initial steps to define a grammar that would support visualizations of SpatialData[291] datasets[292]). Stepping stones to such a comprehensive graphical authoring interface include the development of a sidebar element with additional controls (Figure G.12) or integration of a coordination editor (Figure A.2a). The development of a grammar-based approach to Vitessce configuration also represents an area for future study, yet the taxonomic organization of single-cell visualizations (Section 2.5.1) should serve as a prerequisite to such work to ensure an expressive and useful grammar.

3.6.1 Limitations and Future Work

Further integration with Scverse data structures. At the outset of the development of Vitessce in 2019, the AnnData data structure had not yet been adopted as the primary data structure used in the single-cell data analysis domain and within many cell atlas projects. Further, the Scverse consortium had not yet been established[426], nor had formats such as MuData[94], SpatialData[291], or OME-NGFF[311]. As a result, the data model in Vitessce was designed to generalize to alternative data sources, including JSON and CSV files (in addition to Scverse data structures, as they emerged). However, this flexibility complicates certain aspects of the implementation. For example, certain assumptions can be made about tables within AnnData objects, such as their shapes and the ordering of their rows, yet we can not necessarily rely on these assumptions in Vitessce as data can be provided via more flexible file formats. The performance of Vitessce could be improved (and maintenance complexity reduced) by further embracing Scverse data structures.

Improved support for ontology terms as identifiers. Vitessce is designed to support visualization of multimodal data from one or more single-cell experiments. However, in multi-experiment settings, Vitessce relies upon the data provider to ensure that identifiers across datasets match. For example, if two datasets use different gene identifiers (*e.g.*, one using Ensembl Gene IDs and the other using Gene Symbols), then in its current implementation Vitessce will be unable to identify corresponding genes between the datasets, as a user might expect. The same problem can arise for cell type and protein identifiers (including within image channel names). In the future, we could employ ontology terms as the identifier-of-truth to resolve these

challenges. However, this could present challenges of mapping existing string identifiers (*e.g.*, for cell types) to the corresponding (or at least closest) ontology term, as well as introduces questions related to ontology selection and versioning. This presents opportunities to develop improved tools for ontology term mapping, and to improve how this metadata is represented in single-cell and imaging data structures.

Database-backed architecture. Recent developments of in-process SQL databases such as DuckDB[240] are enabling new architectures for interactive visualization software, such as Mosaic[198]. In these architectures, an SQL database stores application data, and the visualization implementations can leverage SQL data transformations, with automatic optimizations for certain classes of transformations. Importantly, the database communication is highly efficient as the database runs in the same application process, without requiring network communication between the database and the application. Using WebAssembly, the database can run directly in a web browser. Finally, as the SQL programming language is widely used and its syntax is straightforward to learn, applications can enable users to map visual properties to custom SQL queries, including queries defined using graphical interfaces. In Vitessce, this type of architecture could be used to store all tabular data, including for the tables within AnnData objects. Then, data transformations could be written using SQL expressions. Using DuckDB extensions for spatial data analysis, this could extend to operations such as calculating the vertices of a polygon enclosing a set of scatterplot points. Moving data transformations from JavaScript to SQL would reduce the code complexity and simplify its maintenance. While we would further expect automatic query optimizations available in DuckDB to improve performance, benchmarking would be required to make this conclusion.

Scalability. As the scale of single-cell datasets continues to grow, software to visualize them must continue to improve in scalability. While Vitessce currently contains optimizations for large-scale image and genomic data rendering, such as retrieving data subsets (*e.g.*, image tiles) at a given resolution (*e.g.*, from pyramidal/multi-resolution image formats), there are opportunities to adopt data tiling and summarization strategies for additional data types, including non-spatial and non-genome-mapped data. One complication is that certain data types, such as gene expression matrices, lack a well-defined ordering that would facilitate tiling and summarization[67]. Addressing these challenges are essential for Vitessce to support efficient visualization of emerging atlas datasets containing millions of cells.

Export publication-ready figures. We have focused on the development of interactive features for exploration. Vitessce currently lacks support for export of publication-ready static figures, for example in a vector graphics format such as SVG. We could use the strategy taken by tools such as HiGlass which support both WebGL-based interactive visualizations and support SVG exports[231]. As WebGL and SVG visualizations use different APIs, this requires duplicate implementations which must be kept in sync.

Storytelling and narratives. In narrative visualizations, stories consisting of text descriptions at waypoints enable guided exploration of a dataset. Within the biomedical domain, the Minerva tool supports narrative visualization of multiplexed imaging data[357]. Narrative visualizations require not only the ability to display data and their authored descriptions, but also to navigate among waypoints. Provenance tracking and action recovery (*i.e.*, undo/redo) approaches such as Trrack[128] facilitate

this type of navigation.

Sidebar. Configuration of Vitessce is performed via a JSON-based specification of datasets, views, and coordinated properties. While individual views in Vitessce are highly interactive, larger scale changes such as dynamic additions of views or data files currently must be made through changes to the JSON configuration. However, users may aim for a graphical user interface to make such modifications. We have designed such a graphical interface, which takes the form of a sidebar element with buttons for not only addition of views, but also modification to the configured datasets and data files, action recovery (undo/redo) (Figure G.12). Future work is needed to complete the implementation of this sidebar to enable more types of configuration modifications via the Vitessce user interface.

Highlight, Select, Filter. Vitessce contains support for highlighting and selecting entities within visualizations. With coordinated views, these highlights and selections are linked across views. However, selection operations in one view may result in a different view effectively becoming filtered. For example, usage of the lasso tool to select cells in the scatterplot currently results in visualization of only these cells in the gene expression violin plot view. This inconsistency may confuse users, especially those who have not previously used Vitessce. Tan *et al.*[405] provide empirical guidelines for coordinated multiple view visualizations based on consistency.

3.7 Conclusion

In this chapter, we discuss the gap in tools for integrative visualization of multimodal and spatial single-cell datasets. We introduce Vitessce to fill this gap, addressing the

need for a scalable, interactive, and extensible framework that supports visualization of heterogeneous data types in multiple computational environments. Its modular architecture, compatibility with multiple file formats, and support for coordinated multiple views enable researchers to integrate previously disconnected data modalities and view them with a single tool. The adoption of Vitessce by multiple data portals, publication-associated websites, and commercial products underscores its utility. Nevertheless, more research is needed to improve its usability, improve integrations with ontologies and knowledge bases, and ensure that the software remains scalable. This foundational work also supports the projects described in the chapters that follow, where we explore applications involving comparative, spatial, and epigenomic data analyses.

4

Cross-experiment comparisons

4.1 Preface

Portions of this chapter have been adapted from a previous publication which I have co-authored:

- Polyphony: an Interactive Transfer Learning Framework for Single-Cell Data Analysis[119]

This chapter presents both the Polyphony project, in which comparisons are between query and reference datasets, as well as the Compasce project, in which comparisons are between case and control groups (*e.g.*, between healthy and disease samples or between samples from disease subgroups). The Polyphony project was led by Furui Cheng with contributions from myself, Huamin Qu, Nils Gehlenborg, and Qianwen Wang, with results published in IEEE Transactions on Visualization and Computer Graphics[119]. My contributions to Polyphony include the development of the user interface and front-end web application, co-interviewing expert users, and paper writing and editing. I led the Compasce project, with additional contributions from Michael Rose, Nikki Bonevich, Matthias Kretzler, Nils Gehlenborg, and the Kidney Precision Medicine Project (KPMP) consortium. I thank all user interview participants who contributed to the evaluations of these projects.

4.2 Introduction

In large-scale human cell atlasing efforts, single-cell experimental techniques are being applied to profile cells from tissue samples originating from tens or hundreds of different human tissue donors. Thus, the complex and high-dimensional data at the single-cell level can be linked to corresponding demographic and clinical metadata at the donor level. A key goal of the single-cell field is to leverage such data linkages (*i.e.*, between cells and their donors) to make comparisons between healthy and disease-afflicted donors that can be used to develop diagnostics, prognostics, and therapeutics that operate at the cellular and molecular levels (Table 1.1).

Atlas data portals, including those of the Human BioMolecular Atlas Program (HuBMAP), Kidney Precision Medicine Project (KPMP), and Cellular Senescence

Network (SenNet) currently embed visualizations of individual single-cell experiments using Vitessce (Chapter 3). However, investigators are interested in using atlas datasets to conduct comparative analyses across multiple experimental datasets, with comparisons driven by clinical and experimental metadata to obtain insights related to health and disease.

Another important type of comparative analysis is that of atlas-guided reference mapping: projection of new single-cell datasets (query datasets) onto an existing reference (*i.e.*, atlas dataset). This query-to-reference projection facilitates tasks such as cell type annotation. For example, the reference dataset may be known to contain rare cell types and cell type annotations that have been validated through expert review. Reference-based cell type annotation not only allows using label transfer to quickly annotate the query dataset, it also enables using annotation methods that leverage the full, high-dimensional gene expression matrix rather than relying on limited lists of marker genes. The field lacks interactive tools that support either type of comparative analysis (case-versus-control or query-versus-reference).

We have addressed this problem by developing comparative visualizations using our Vitessce framework for single-cell data visualization. These comparative visualizations have been designed based on interviews we conducted with biologists and clinicians to understand current comparative analysis workflows and challenges.

4.3 Related Work

In this section, we discuss related literature on methods for comparative analysis of single-cell data, visualization for comparison, and tools designed for visualization of single-cell comparative analysis results.

4.3.1 Computational Methods for Comparative Analysis

Many computational methods for single-cell data analysis can be considered comparative. As there are over one thousand computational methods[463] and "comparative analysis" is a broad term, here we focus on the types of analyses highlighted in the Single Cell Best Practices guide within the "Dealing with Conditions" section: Differential Gene Expression Analysis, Gene Set Enrichment and Pathway Analysis, Compositional Analysis, and Perturbation Modeling[278,201].

The goal of differential expression analysis is to identify genes which exhibit expression patterns that are significantly different in one group of cells versus another. This is achieved by first stratifying the cells into groups, for instance, according to a condition variable such as disease status. Cells may also be filtered, for instance, if the analyst is interested in differences within a particular cell type class. The analysis proceeds by fitting a statistical model to the gene expression data, estimating parameters such as library size (*i.e.*, total number of reads) and dispersion (*i.e.*, coefficient of variation). Although many different models have been proposed, there are two main classes: cell-level models and "pseudobulk"-level models. Pseudobulk-based methods enable reuse of models initially designed for bulk (in contrast to single-cell) RNA-sequencing data by aggregating single cells. In benchmarks, the pseudobulk approach has been shown to effectively account for the sparsity of single-cell data, despite many cell-level models explicitly modeling sparsity (*e.g.*, "dropout" events)[432]. After estimating the model parameters, such methods will test each gene based on the null hypothesis that its expression is the same in the *case* condition as it is in the *control* condition, for example using an exact test[363]. This enables calculating the probability of encountering relative gene expression abundance between the groups that is at

least as extreme as that which has been observed.

While differential expression test results can be informative on their own, it can be challenging for analysts to interpret the individual genes returned. The data analyst may be unfamiliar with these genes or it might be unclear whether the gene is causal for the condition phenotype. Thus, gene set enrichment analysis (GSEA) was developed as a way to interpret ranked gene lists by using prior biological knowledge in the form of curated gene sets which share biological function (*i.e.*, pathways), regulation, or genomic location[402]. Given a ranked gene list L and a predefined set of genes S, GSEA determines whether the genes in S are enriched at the top or bottom of L versus distributed randomly throughout L. The GSEA method computes an enrichment score for set S and a P-value (based on a null distribution derived through permutation of condition, sample, or gene labels). This process can be repeated for an entire database of gene sets (adjusting for multiple hypothesis testing). Alternative scoring and significance approaches have been proposed, for example, BlitzGSEA which improves upon the performance of the original algorithm[247]. Gene set enrichment analysis approaches not only assist in interpretation long lists of differentially-expressed genes but also facilitate identification of signals that would otherwise be overwhelmed by noise, such as sets of genes which individually are subtly affected by a condition but can be uncovered through their set membership.

Whereas differential expression and gene set analyses are biomarker-centric, identifying genes or gene sets that are condition-specific, investigators also hypothesize that relative abundances of different cell types are relevant to disease processes and characteristic of particular disease states. Thus, statistical models of cell type composition have been proposed. Such models must account for the proportional nature of the data (*e.g.*, an increase in abundance of cell type A will necessarily result in decreases

in abundance for other cell types). Two such compositional analysis methods are scCODA[101] and Cacoa[350]. ScCODA estimates the effect of a condition (*e.g.*, disease versus healthy) on cell type proportions, with respect to a selected reference cell type whose abundance is assumed to be independent of the condition. Cacoa similarly relies on selection of a reference cell type to identify condition-associated changes in cell type composition. While the authors of scCODA and Cacoa demonstrate good performance on simulated and real datasets (*e.g.*, reproducing cell type abundance findings from prior studies), the cluster-based methods of scCODA and Cacoa are sensitive to the accuracy and granularity of cell type annotation. To overcome this limitation, tree-based and cluster-free methods have been proposed[329,350,130,51].

4.3.2 Computational Methods for Reference-Based Analysis

It is of great interest to use large atlas datasets as references to address current challenges in single-cell data analysis. One such challenge is the time-consuming nature of manual cell type annotation. In reference-based analysis, an atlas dataset serves as a reference onto which new, smaller "query" datasets can be projected. Through query-to-reference mapping, the process of cell type annotation can be streamlined: data can be harmonized through transfer learning strategies that assume data exists in a shared (typically low-dimensional) space where comparisons can be made reliably. The single-cell architectural surgery (scArches) method learns a low-dimensional embedding space for a reference dataset using a conditional variational autoencoder (CVAE)[275]. To project a query dataset into this space, the weights of the CVAE neural network are fine-tuned using an "adaptor" strategy, whereby only the weights connected to the query dataset are allowed to vary while the weights connected to the reference remain fixed. The Symphony method[223], on the other hand, uses a linear

mixed model and efficiently compresses the reference dataset into a set of cluster descriptors to facilitate reference mapping. Reference-based analysis can be viewed as a special case of comparative analysis in which the two "conditions" being compared are *query* and *reference*.

4.3.3 Comparative Data Visualization

There is a rich literature on visualization for comparison in general contexts which can be used to inform designs for visualizing comparison for single-cell data. Gleicher outlines four key considerations for visualizing comparison: Comparative Elements, Comparative Challenges, Scalability Strategies, and Comparative Designs[171]. Gleicher categorizes potential Scalability Strategies and Comparative Designs. In this categorization, there are three identified scalability strategies: *Scan Sequentially*, *Select Subset*, and *Summarize Somehow*, and there are three categories of comparative designs: *Juxtapose*, *Superpose*, and *Explicit Encoding*. We can apply Gleicher's considerations to navigate the design space of comparative visualizations in the single-cell domain.

4.3.4 Comparative Visualization of Single-Cell Data

There are several existing interactive tools for visualization of single-cell data. These include Vitessce (Chapter 3), CELLxGENE Explorer[303], and UCSC Cell Browser[396]. These (and most other current tools) are designed to visualize data from a single experiment or biosample, as opposed to showing data from multiple experiments or samples simultaneously (*e.g.*, multiple tissue samples from an atlas). At a minimum, current tools can visualize a pre-integrated dataset of cells from multiple biosamples (assuming the tool is scalable to the total number of cells) and can color or stratify

cells according to a sample- or donor-level metadata variable by storing the variable as a column of the cell-level metadata table (and visualizing as if it were a cell-level metadata variable such as "cell type"). However, then such a metadata variable must occupy the visual channel that would otherwise be used for displaying cell type information (*i.e.*, preventing simultaneous visualization of cell type *and* a donor-level variable). Beyond this, current tools are generally limited in supporting sample- or donor-level information (in the absence of redundantly storing it at the per-cell level) and further limited in visualizing such information to facilitate comparison. In other words, in order for a tool to display certain comparative visualizations in which comparison is based on a sample-level metadata variable, the tool must first be designed to store and reason about sample-level data (in addition to cell-level data).

One visualization tool designed to support sample-level comparisons is SpaCeCo by Somarakis *et al.*[393], which enables comparison of two cohorts of spatial single-cell 'omics data samples. SpaCeCo presents innovative comparative visualization designs intended to facilitate four analysis tasks identified through collaboration with pathologists: *Comparison based on abundance of cell types*, *Comparison based on multicellular microenvironments* (defined as spatially co-occurring pairs of cell types or user-defined combinations of cell types), *Detection of outlier samples*, and *Relation of findings to their spatial position*[393]. The authors report primarily positive feedback from study participants with their approach. However, SpaCeCo is limited to spatially-resolved data (as opposed to non-spatial single-cell data) and focuses on comparisons of cell type abundance (relative and absolute) and spatial cell type co-occurrence (with no support for comparisons of molecular data such as gene expression or protein abundance). A focus on cell type abundance is limiting due to the constraints imposed by (flat/non-hierarchical) cell type annotations generally: a fixed

level of cell type granularity, potential annotation uncertainty (with potential segmentation errors in the context of spatial 'omics), and throwing away the molecular-level data which underlies cell type definitions and could be more clinically actionable. The SpaCeCo paper does not discuss the potential for switching among multiple cell type annotation results or among multiple pairs of cohorts. Further, the tool only supports display and comparison of cell type abundances in the forms of proportions and absolute numbers, which are not interpretable measures due to the compositional nature of cell type proportions. Such data requires compositionally-informed statistics and comparisons with respect to a selected reference cell type[101,350] (which aligns with the authors' note that, "'statistical testing of differences found between cohorts' was requested [by users of SpaCeCo]")[393].

Another related tool offering comparative visualizations of single-cell data is the Cytosplore Simian Viewer, which focuses on cross-species comparisons[71]. The authors note that the development of this system was motivated by the specific aim of their collaborators to identify human-specific traits through pairwise comparison between human and one other species. The Cytosplore Simian Viewer system consists of multiple coordinated interactive visualizations, including dimensionality reduction scatterplots, gene expression bar plots, an evolutionary distance heatmap, and a differential expression table. The authors focus on enabling users to accomplish the following tasks: *Identify cross-species cell clusters of interest by exploring in-species cluster attributes*, *Identify genes of interest for the identified cross-species clusters*, and *Compare cluster-level gene expression values between species*. The authors acknowledge that this system was designed and implemented around one particular dataset from their collaborators and it was evaluated using a case study methodology, noting that both of these aspects could be improved upon through follow-up research.

Other related work includes the CELLxGENE Discover platform[129] which contains three modules enabling query and visualization across the millions of cells and hundreds of curated datasets available within the platform. In contrast to the desktop-based visual analytics systems of SpaCeCo and Cytosplore Simian Viewer, CELLxGENE Discover is a web-based data portal-style platform. The platform contains modules called Differential Expression, Gene Expression, and Cell Guide. The Differential Expression module can be used to compare two groups of cells on-demand, with groups defined in terms of tissue type, cell type, and donor-level metadata variables including disease, ethnicity, and sex. Results from differential expression tests are displayed using a table with three columns: Gene, Log Fold Change, and Effect Size. The Gene Expression module displays (non-spatial) gene expression data using dot plots stratified by cell type and, optionally, a sample-level metadata variable such as Sex or Disease. Biomarkers in the dot plot can be either user-selected, iteratively added based on cell type marker genes, or initialized using the results of a query made using the Differential Expression module. The Cell Guide module displays information about a cell type of interest, based on data from the Cell Ontology, and lists marker genes for the cell type which have been identified both computationally and canonically (*i.e.*, through expert curation). Individual datasets of interest can be visualized using the CELLxGENE Explorer tool[303]. While these visualization modules integrated into CELLxGENE Discover are powerful, the modules are located on different webpages which constrain users to viewing a single visualization at a time. Further, the Discover platform does not contain visualizations of differential expression results (only a table view), nor does it offer gene set enrichment or cell type composition analysis features. While spatial data can be displayed using the Explorer scatterplot viewer, the support for spatial data types is limited to the Ex-

plorer module. Finally, because Discover presents data from many datasets that have been automatically integrated, the authors advise users that, "the displayed values may still be affected by batch effects, especially for rare cell types,"[129], potentially undermining trust in derived findings. This exemplifies the need for comparative visualization solutions that are interoperable with user-processed atlas datasets, enabling users to leverage the results of an expert-validated data integration process (or tools designed to facilitate the data integration process itself).

Within the KPMP consortium, the KPMP Atlas Explorer website contains features for exploration of differential gene expression results[42]. This includes the ability to select a cell type (or FTU) of interest using an anatomical diagram, as well as to select an experimental modality such as scRNA-seq, and view the precomputed differential expression results for the selected cell type versus all other cell types. The results are displayed using a table, similar to the CELLxGENE Differential Expression module. The table allows for selection of a gene of interest to view its expression values in a dimensionality reduction scatterplot. The tool also links to CELLxGENE Explorer where the same dataset can be visualized. While the KPMP Atlas Explorer enables users interested in KPMP data to obtain an overview of cell type marker genes, the tool does not offer features to compare groups of samples (*e.g.*, healthy reference tissue samples versus those from individuals with Chronic Kidney Disease). Further, the tool contains a limited set of interactive visualization features (*e.g.*, scatterplot zooming, panning, and tooltips) and is tightly coupled to the KPMP Atlas data, preventing reuse with other atlas datasets.

4.4 Polyphony: an Interactive Transfer Learning Framework for Single-Cell Data Analysis

Reference-based cell-type annotation can significantly reduce time and effort in single-cell analysis by transferring labels from a previously-annotated dataset to a new dataset. However, usage of end-to-end label transfer methods in this context is challenging due to the entanglement of technical (*e.g.*, from different sequencing batches or techniques) and biological (*e.g.*, from different cellular microenvironments) variations, only the first of which must be removed. To address this issue, we propose *Polyphony*, an interactive transfer learning (ITL) framework designed to complement biologists' knowledge with computational methods. *Polyphony* is motivated and guided by domain experts' needs for a controllable, interactive, and algorithm-assisted annotation process. We identified these needs through interviews with seven biologists. We introduce anchors, analogous cell populations across datasets, as a paradigm to explain the computational process and collect user feedback for model improvement. We further design a set of visualizations and interactions to empower users to add, delete, or modify anchors, resulting in refined cell type annotations. The effectiveness of this approach is demonstrated through use cases and interviews with two biologists. The key contributions of this project include:

- An outline of the domain problem and analysis tasks in reference-based single-cell data analysis.
- *Polyphony*, a visual analytics tool that tightly integrates interactive visualizations with an anchor-based interactive transfer learning framework to facilitate human-AI collaboration during single-cell data integration and annotation.

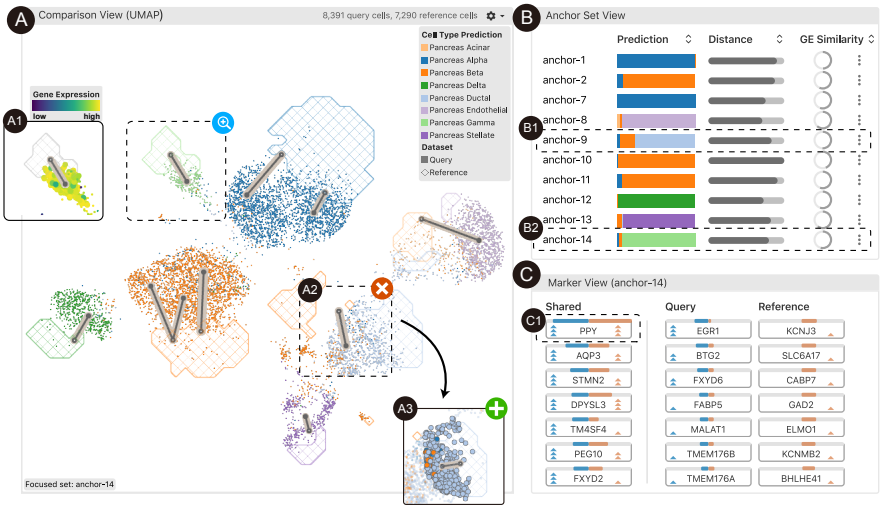

Figure 4.1: The interface of Polyphony contains three views: the comparison view (A), the anchor set view (B), and the marker view (C). The comparison view provides an overview of the joint embedding space, and offers users interactions to inspect (A1), delete (A2), and add (A3) anchors. The anchor set view orders the anchors in a table, supporting inspection and comparison of different anchors (B1-2). The marker view shows the significant genes (C1) for the query and reference cells from a focal anchor.

- An evaluation consisting of feedback from experts demonstrating the usefulness of the system.

4.4.1 Informing the Design

We conducted multiple rounds of interviews with seven experts in single-cell data analysis (six postdoctoral researchers and one assistant professor) to understand the domain problem and data, clarify analysis tasks, and evaluate the system design. This design process follows the Nested Model for visualization design and validation proposed by Munzner[316].

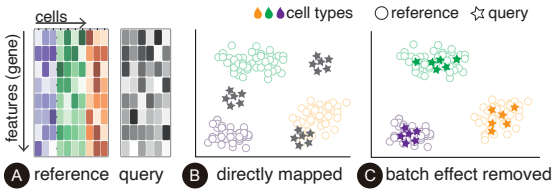

Figure 4.2: Background on batch effects. (A) Single-cell transcriptomics data is represented by a matrix that records the amount of RNA corresponding to each gene (row) detected in each cell (column). Batch effects appear in data collected from different studies and must be removed to allow an integrative analysis (B, C).

Domain Problem

In single-cell transcriptomics, cell type annotation is considered an elementary analysis task, for which many computational methods have been developed[248]. Reference-based annotation enables efficient cell type identification by transferring labels from reference datasets to new datasets. A critical step in many reference-based annotation methods is to integrate different datasets into the same embedding space, which allows labels to be transferred from one dataset to another. Batch effects occur when non-biological factors (*e.g.*, sequencing instrument differences) result in systematic biases in the data produced by an experiment. Batch effects can be difficult to resolve since they are usually coupled with biological variability among different experimental samples. Under-correction (*i.e.*, failure to fully remove batch effects) or over-correction (*i.e.*, removal of both technical and biological variability) of batch effects can lead to inaccurate label transfer (Figure 4.2). We aim to improve data integration and annotation results by involving human users and utilizing their feedback to better distinguish batch effects from biologically meaningful variation.

Analysis Tasks

Through initial interviews, we identified six tasks that users aim to conduct during reference-based single-cell analysis, and further find that two such tasks (T1.5 and T1.6) are not well supported by existing tools.

T1.1: Gain an overview of the joint embedding space. Experts usually start their analysis with an overview of the cell embedding space, which is typically mapped into a 2D coordinate system using dimensionality reduction methods (reported by experts E1-E7; Section 1.9.2). This overview enables a preliminary assessment of the *clustering quality* (*i.e.*, whether cell type clusters are easily distinguishable) and the *cell integration quality* (*i.e.*, how the cells from different datasets are fused). Experts trust cell type annotation results only when they perceive that cells are well clustered and integrated in the latent space.

T1.2: Understand cell correspondence across datasets. After gaining an overview of the embedding space, experts expect to further explore the cell correspondences across the datasets (*i.e.*, which query clusters are nearby which reference cell types, raised by E5). The correspondence helps experts understand how their data is mapped onto the reference and suggests how the reference labels might be used to annotate their data.

T1.3: Identify potential unique cell populations. Experts are generally interested in identifying unique cell populations that appear in the query dataset but not the reference. In specific, when such cell populations are present, experts expect to select query cell clusters that do not overlap with well-characterized cell types for

further inspection (suggested by E2). In some cases, experts have unique prior knowledge of certain genes, such as cell marker genes that distinguish cell types that have been poorly defined in the literature. They expect to evaluate marker gene expression magnitude to verify and refine the cell population selection (suggested by E3).

T1.4: Understand cell identities. The analysis ends with a full list of annotated cell types. For each cell cluster, experts not only record the cell type label but also the biomarkers (*i.e.*, genes that are differentially expressed among clusters and define cell types). These biomarkers are often related to cell morphology and physiology and play an important role in downstream analysis. Typical machine learning models for cell type annotation only produce cell type labels. Experts (E2, E3, E6) expressed their need to identify biomarkers through separate statistical analyses.

T1.5: Understand the integration process. While ML-based annotation methods enable efficient analyses, the computational process is often opaque to users. Experts expressed the need to understand how each step of the data integration process changes their data. For example, E6 mentioned that they usually investigate how the clusters in the embedding space change before and after data integration. Importantly, the analysis goal is to gain a deep understanding about the data. Experts wish for a tool that is, "more than just giving me an answer" (E4).

T1.6: Refine the model to align human knowledge. The annotation results are not always satisfactory, where the model may under-correct or over-correct the dataset. Instead of checking and manipulating the annotation results one by one, an alternative method is to correct the model systemically, as suggested by E5. Enabling

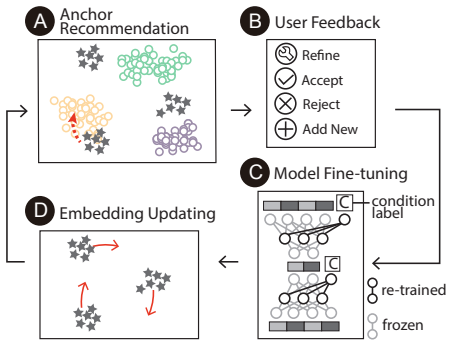

Figure 4.3: The proposed interactive transfer learning framework includes four key steps: anchor recommendation, user feedback, model fine-tuning, and embedding updating.

users to examine, validate, and refine the model can increase user trust, improve their understanding about the data, and generate more accurate annotations[389].

4.4.2 Polyphony

To address the articulated user needs during reference-based cell type annotation, we developed the Polyphony system which tightly integrates an interactive transfer learning framework with a specialized visual analytics system.

Interactive Transfer Learning Model

The ITL process includes four steps (Figure 4.3). First, the model compares the two datasets and recommends *anchors* to users (Figure 4.3A). Afterwards, users inspect and update these anchors according to their domain knowledge (Figure 4.3B). The model is then fine-tuned using the user-confirmed anchors (Figure 4.3C), updating the annotations and embedding for the query dataset (Figure 4.3D). The model then recommends new anchors based on the updated embedding space. These four steps repeat until the user is satisfied with the integration and decides to apply the predic-

tions to annotate the remaining cells.

Prior to user interactions, the joint embedding space is initialized using a conditional variational autoencoder (CVAE) model based on scArches[275] and implemented using the scVI framework[274]. The CVAE model generates latent embeddings for the input data where the effects of condition labels are regressed out. In the framework inherited from scArches, such condition labels are used to index batch effects (*e.g.*, sequencing technologies, experiment laboratories, patients, or a combination of these categorical variables). The CVAE model is first trained on the reference dataset and then adapted to the query dataset by adding a conditional node to index batches in the query dataset (Figure 4.3C). The parameters in the original model are fixed while the weights related to the new conditional node are learned on the query dataset. As a result, the CVAE model can generate embeddings for the query dataset while removing the batch effects between reference and query. Upon learning a latent representation, cell type labels for the query dataset are predicted using a k-nearest-neighbors (kNN) classifier trained on the reference dataset, in a manner similar to Kang *et al.*[223].

In the absence of user feedback, the CVAE model can potentially over-correct and confuse biological variability with batch effects. To resolve this issue, we use the concept of an integration anchor, which we define as a pair of cell sets (one from each dataset: query and reference) that share biological similarities and should be proximal in the joint embedding space. While anchor-based techniques have previously been proposed in the field of single-cell data integration[70,182,223,241,434], they are typically identified based on statistical assumptions between datasets, which may be invalid in certain scenarios (*e.g.*, the query dataset contains a novel cell type). In contrast, we generate anchors in a progressive manner and involve humans to validate and refine

them.

Initial anchors are recommended using the efficient algorithms from Symphony[223] and Harmony[241]. First, the reference dataset is compressed into its cluster descriptors, which contain information about the cluster centers and cluster sizes. Then, query cells are mapped to reference clusters by calculating the probability of each query cell being mapped to each cluster and taking the highest-confidence assignments. The resulting pairs of query and reference cell sets form the initial anchor recommendations, which users can subsequently inspect and interact with using specialized interactive visualizations (Section 4.4.2).

As users inspect the anchors and interact with the interactive visualizations, they can provide feedback to the model which can be used in the fine-tuning and embedding updating steps (Figure 4.3C-D). Users can delete low-quality anchors, map a reference cell set to an alternative set of query cells, or confirm anchors that look promising according to the statistical information (*e.g.*, distance and expression similarity metrics) and shared-versus-distinct marker genes between the query and reference cell sets. Based on this feedback, we "correct" the embedding of the user-confirmed anchors by mapping them to the corresponding reference cluster. Inspired by the correction algorithm of Symphony[223], we use a linear mixed model to update the query cell embeddings, applying corrections only to those query cells contained in user-verified anchors. Then, we fine-tune the model using a semi-supervised approach. Specifically, we expect that the generated latent embeddings of the anchor query cells match the corrected embeddings. Therefore, we add a penalization term in the loss function to measure the distance between the generated and the corrected embeddings for each cell. Upon embedding updates, we update the cell type annotation predictions using the kNN classifier, producing refined annotations that have been informed

by user feedback at the anchor level.

User Interface

The Polyphony user interface contains three coordinated views: the *comparison view*, the *anchor set view*, and the *marker view*. The comparison view (Figure 4.1A) offers an overview of the embedding spaces (T1.1), enabling users to understand the distributions of cells and identify cell populations of interest (T1.3). The anchor set view (Figure 4.1B) shows statistical information about each anchor (T1.2). The marker view (Figure 4.1C) displays differential gene expression analysis results for a focal anchor (T1.4). Together, these views support interactions that enable users to communicate feedback to the model via anchors (T1.5, T1.6).

The user workflow typically begins with the comparison view, where the user gets an understanding of the data integration quality. The user may next find a recommended anchor of interest for further inspection. In the anchor set view, the user checks this anchor as well as other anchors, including those that have been recommended by the system and those based on manual user selections. The anchor set view contains controls for selection, deletion, confirmation, and editing of anchors. To inspect a selected anchor, the user looks to the marker view. The marker view displays genes which are significantly differentially expressed in the query cell set, the reference cell set, or both cell sets. The user can select a gene from the marker view, which will color the cells in the comparison view according to the expression values of the gene. Finally, by submitting their feedback, the user prompts a round of fine-tuning and embedding updates. Multiple rounds of visualization, feedback, and model updates can be performed to refine the data integration.

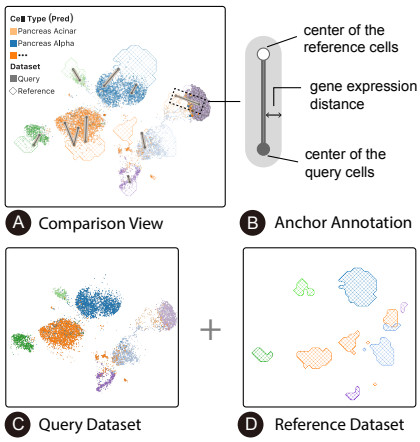

Figure 4.4: Comparison view. (A) The complete visualization design. We use different encodings for the query (C) and the reference (D) datasets. The anchor annotations (B) encode the gene expression distances between the query cells and the reference cells, helping users to better understand the integration quality.

Comparison View. The comparison view (Figure 4.4A) presents both the query and the reference datasets along with different properties of the cells (*e.g.*, cell type labels or predictions, dataset membership, and expression magnitude of selected genes) and their geometry structures in the latent space. This view helps users obtain an overview of the embedding space, evaluate the integration quality (T1.1), and identify cell populations of interest for further examination (T1.3). Further, the anchor visualization (Figure 4.4B) in the embedding space provides an intuitive approach for users to understand the cell correspondence across datasets (T1.2).

We use different visual representations to encode query (Figure 4.4C) and reference cells (Figure 4.4D) in the same view. For the query cells, we use a scatterplot to provide detailed information about these cells and offer flexible selections. The color of each point encodes the predicted cell type. For the reference cells, we first calculate the density of each type of cell across a grid (with an adjustable grid size) in the projected space. Then, we draw contours to outline dense grid regions according to a

unified density threshold adjustable by users. The contour areas are textured to distinguish them from the query cell scatterplot points.

Layered above the hybrid visualization of cells from both datasets, we display anchor annotations to help users understand anchor quality and membership in the embedding space context (T1.2). The annotation design for an anchor (Figure 4.4B) is composed of two parts. The inner part contains two linked dots which indicate the centers of the query (solid dot) and reference (hollow dot) cell sets. The outer part encodes the quality of the anchor using the line width (*e.g.*, a wider line denotes lower quality). As suggested by the experts, we use gene expression (GE) similarity as the anchor quality metric. We define GE similarity as the intersection over the union of the gene sets that express significantly differentially between the anchor cells and the rest of the cells, where the significant genes are selected as the top-100 genes when ranked by the z-score under a Wilcoxon test. This design enables users to assess anchor quality and identify anchors to inspect and modify depending on the analysis task at hand.

We note that this visualization is an evolution of an initial design (Figure 4.5) in which we used two coordinated scatterplots to support the comparison task: one using color to encode the dataset membership of each cell, another using color to encode cell types (or cell type predictions). We found that the users were not able to easily join this information together when many cell types were present (*e.g.*, making it difficult to distinguish well-integrated cell types from poorly-integrated cell types). Distinguishing cell type-specific integration quality is essential in selecting proper anchors (T1.2). This observation motivated us to design a single-view visualization to encode the dataset and cell type information simultaneously. In doing so, we tested both heatmaps and contour lines to visualize the reference cell density layered below

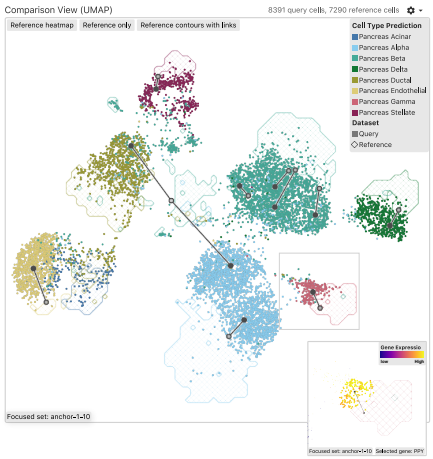

Figure 4.5: Former design of comparison view. In this earlier design, the comparison view used two coordinated scatterplots to simultaneously display gene expression quantities and cell type annotations.

the query cell scatterplot. We found that experts prefer contours because heatmaps not only take longer to render but also are visually overwhelming.

Anchor Set View. The anchor set view arranges the anchors in a table (Figure 4.1B), where each entry represents an anchor that has either been recommended by the system or created by the user. The recommended anchors denote those cells that the model considers most similar between the query and reference datasets (T1.2). By interacting with these anchors, users understand the model results (T1.5) and provide feedback for further model refinement (T1.6).

For each anchor, the table presents the composition of the cell type predictions, the distance between the pair of sets in the latent space, and the GE similarity. These properties were verified by experts to ensure that they represent the most important descriptive information for an anchor. We use horizontal stacked bar plots to show the proportions of predicted cell types, where the color encodes the cell type, consis-

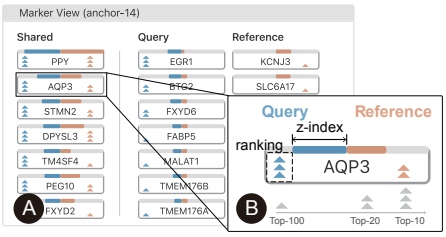

Figure 4.6: Marker view. (A) This view groups significantly differentially-expressed genes into three columns. (B) The glyph design enables the comparison of gene significance and gene ranking simultaneously.

tent with the comparison view. In the distance column, we use bar length to indicate the median distance between cells in the query set and the center of the reference set. In the GE similarity column, we use the sweep angle of an arc within a circle to encode the similarity score (in the range 0 to 1).

The GE similarity scores help users make sense of the differences between the query and reference cells in the gene expression space. A lower similarity score indicates that the two cell groups share very different marker genes, suggesting the anchor should be removed in most cases. In contrast, when the similarity score is high, users may still need to investigate the top differentially expressed genes. Sometimes, two similar cell types may share expression patterns for all but a few critical marker genes whose expression patterns distinguish the cell types. In such a case, expert knowledge is required to make a judgement.

Marker View. The marker view (Figure 4.6A) enables users to conduct gene-level analysis on the significantly differentially-expressed genes (T1.4). As suggested by domain experts, we horizontally divide the interface that lists the significant genes into three columns: significant in both query and reference ("shared"), significant only in query, and significant only in reference. This design helps them distinguish the shared

significant genes and those specific to each dataset.

For each gene, we display a glyph (Figure 4.6B) to enable the comparison of gene significance (z-scores in our case) and gene significance rankings simultaneously in both the query and reference sets. We use two bars with contrasting colors to visualize the z-scores of a gene in the query and the reference datasets. A longer bar indicates a higher level of significance. Stacked triangles represent the ranking of a gene in the query dataset (triangles on the left side) and the reference dataset (the ones on the right side). More triangles indicate a higher ranking and imply a more important gene. Three triangles indicate a top-10 ranking. Two or one triangle(s) indicates a top-20 or top-100 ranking, respectively. These thresholds were suggested by the domain experts in our interviews.

To help users better understand the distribution of gene expression in the whole cell population, we coordinate the marker view with the comparison view. When users click a gene, the color encoding in the comparison view will be changed to represent the gene expression magnitude for each cell (Figure 4.1A1).

System Implementation

Polyphony contains a front-end user interface for data visualization and a back-end server for data storage, model training, and anchor operations. The front-end is built upon Vitessce (Chapter 3). We extend Vitessce using its plugin API to define new views, coordinated properties, and data loading functionalities designed for reference-based analysis. The back-end is a Python application that relies on Scanpy[448], scVI-tools[274], scArches[275], PyTorch[341], and scikit-learn[346]. All cell-related data produced in this system (*e.g.*, latent vectors, model predictions, and anchor sets) are compressed and stored in AnnData format[427] allowing users to easily share the results.

Finally, we use a Flask-based web server for communication between the front-end and back-end components.

4.4.3 Evaluation

We conducted semi-structured interviews with two of the initially-interviewed experts in single-cell data analysis (E3 and E4) to obtain qualitative feedback and evaluate the Polyphony system. Since the experts were already familiar with the project, we briefly reminded them about the background and demonstrated Polyphony with a showcase using a pancreas dataset. Then, we let the experts explore the system for 30 minutes freely. The participants were encouraged to ask questions during the exploration. We took notes on the system usage and asked the experts the rationale behind their operations upon anchors. Afterward, we collected their feedback on their usage experience, expected usage in their work, and desired improvements.

System design. Both experts agreed that visualization designs in Polyphony were intuitive and that the system was easy to use. At first, E3 expressed confusion that an anchor did not contain all cells in what appeared to be a cluster in the embedding. After explaining that anchors are analogous cell sets across datasets that the model is confident about, they understood. They highly appreciated the flexibility in selecting cell populations of interest and obtaining recommended reference cells. They commented, "this tool does not just give me an answer but allows me to interact with it." E4 suggested that in their studies, they noticed that existing computational methods for integration sometimes produce undesired results. They highly appreciated that Polyphony enables improving the integration through their feedback.

Anchor validation. E4 confirmed that whether an expert confirms or rejects anchor recommendations highly depends on the expert's knowledge about the corresponding cell type and marker genes. They commented, "...if I knew this type of cell and saw related marker genes expressed, I would be pretty sure about the anchor." Otherwise, they need to check the literature to make the decision. E3 expressed similar opinions and suggested that certain types of cells are more likely to be chosen as the anchors by users than others, stating, "...some cell types are well-understood by biologists, while some are hard to define."

Application scenarios. Both experts said that they would use the system to understand how the integration was performed and make further improvements. E4 suggested that this process can help them to understand unfamiliar cell populations in their data. They described the process as first using the familiar cell populations as anchors to integrate the datasets. Then, they would check how the unfamiliar populations are mapped to the reference for further understanding. On a different note, E3 commented that they would like to use the system to conduct comparative studies of cells from different sources. For instance, they might compare cells from different organs or from different patients by dividing them into a treatment group and a control group.

Desired improvements. Both experts are eager to use the system with their own data (with over 100,000 cells). They also wanted the system to contain multiple preloaded reference datasets (*e.g.*, cells collected from different organs) to select from. Further, E3 hoped that the system could enable users to simultaneously select multiple cell populations and perform differential expression analysis within the selected

sub-populations of cells. E4 suggested that the system could also integrate gene set enrichment analysis[402] to provide annotations for genes, which would help experts quickly gain a high-level understanding of the potential functions of the cell populations.

4.4.4 Discussion

The Polyphony system enables users to steer the integration process via anchors based on their observation of the integration results and their domain knowledge about cell types. Observing these cells and interactively steering the model allows users to conduct the joint analysis with much-improved efficiency. We find that the anchor paradigm is user-friendly while also facilitating model updates through semi-supervised learning.

Limitations and Future Work

Scale to large datasets. Polyphony currently supports the analysis of moderately-sized real-world datasets containing tens of thousands of cells. We tested Polyphony using three datasets. Two are moderately-sized datasets and one is a large-scale dataset containing 274,346 immune cells from patients with COVID-19 (154,723 reference cells and 62,469 query cells)[269]. The model updates (with 50 epochs) take about four minutes for the COVID dataset. These experiments were conducted on a server with an NVIDIA Tesla K80 GPU.

While the visualization front-end in Polyphony can display hundreds of thousands of cells with smooth interactions, preprocessing steps (*e.g.*, computing anchor centroids) are a bottleneck (three minutes for 200,000 cells). This can be reduced by

moving steps to the back-end, facilitating use with the largest available references (*e.g.*, Azimuth references, some of which contain over 500,000 cells[187]).

Select a meaningful reference dataset. Polyphony assumes a high-quality reference dataset and focuses on transferring labels from the reference to annotate query datasets. However, users may inadvertently use a low-quality reference (*e.g.*, wrong labels, mismatched cells), which can severely undermine the annotation results. More research is needed to understand how reference dataset quality influences system usage.

The configuration and deployment of Polyphony could also be improved. While currently, the user must provide AnnData objects containing the query and reference datasets, in the future Polyphony could provide pre-loaded high-quality reference datasets. Further, while we deployed the back-end system to a cloud server during its development and evaluation, more work is required to deploy & maintain the system long-term and make it accessible to biologists worldwide.

Support multi-modal and spatial omics data. In this work, we focus on the use of Polyphony with single-cell transcriptomics datasets. Future work could extend the anchor-based framework to support multimodal and spatial omics data. A promising direction is to leverage multimodal integration models, such as totalVI[166], to learn joint embeddings. Then, new visualizations can be designed to support query-versus-reference comparisons across different modalities.

4.5 Compasce: Comparative Visualization for Single-Cell Resolution Tissue Atlas Data

Biologists, clinicians, and pathologists aim to use large atlas datasets containing millions of cellular measurements and hundreds of images to improve disease diagnosis and develop therapeutics. To gain a comprehensive understanding of biological function, multiple individual experimental datasets obtained from samples associated with different disease states must be analyzed together. However, interactive visualization tools for single-cell data currently lack comparative visualization features that leverage associated clinical metadata, as well as multimodal data such as histology images. We introduce `Compasce` ("Compare Single-Cell Experiments"), a system for comparative and interactive visualization of single-cell data. The design of `Compasce` is informed by exploratory expert interviews in which we identified gaps between user goals and current tools. Through case studies and user interviews, we validate that `Compasce` fills these gaps and enables users without programming expertise to perform important comparative visualization tasks which encompass hypothesis generation and confirmation.

The key contributions of this project include:

- A formative interview study with experts to understand analysis workflows for comparison of data from multiple single-cell experiments, including spatially-resolved data.
- Abstraction of the domain data and analysis goals into visual analysis tasks.
- *`Compasce`*, a visual analytics tool that is designed to support comparison of single-cell atlas data based on clinical metadata.

- Evaluations consisting of two case studies and two user interviews.

4.5.1 Informing the Design

This section describes our design process, including interview and survey methodology. We summarize our findings in the form of analysis tasks, goals, and workflows which have informed the system design.

Formative Interview Methodology

We conducted semi-structured interviews with members of disease- and organ-focused cell atlasing consortia to understand current approaches and challenges in comparative analysis of multiple single-cell datasets (Section H.1). Participants were also asked to express their vision for an ideal workflow (*i.e.*, assuming that current challenges have been overcome).

Study Protocol

We conducted ten interviews between November 2022 and April 2023, with each lasting one hour. One researcher led the interview questions while a second researcher took notes and raised additional follow-up questions, in a pair interview style with driver and navigator roles[52]. Prior to each interview, interviewers reviewed relevant publications authored by the interviewee to gain a baseline understanding of their prior research. When participants referred to interactive visualizations, graphical interfaces, or publication figures, they were invited to share further details with interviewers to contextualize the discussion. Where a participant approved, the interview was recorded to facilitate transcription. No compensation was provided to interview

participants.

Participants and Recruitment

We recruited known experts within the KPMP consortium to participate. We aimed to include clinician-scientists, pathologist-scientists, biologists, and biological data analysts. Through their consortium membership, these experts share the goal of improving the understanding and treatment of kidney disease. At the start of each interview, we asked each participant whether they identified with each of seven pre-defined personas (Section H.1). We conducted ten formative interviews with members of the KPMP, with interviewee self-reported personas reported in Table 4.1. Of note, these included four clinicians and one pathologist.

Qualitative Data Analysis

We analyzed the interview transcripts using an open coding approach and the AtlasTI qualitative data analysis software[168]. We organized and related the codes to identify key analysis goals and domain-specific questions.

Methodological Limitations

While these semi-structured interviews have been valuable in improving our understanding of analysis goals and current challenges, our interview methodology has several known limitations. First, all interview participants are members of the KPMP consortium, which is focused on a single organ (kidney) and disease (kidney disease), so it is possible that our results do not generalize to other contexts (*e.g.*, non-human, other organs, or other diseases such as cancer). Further, participant responses may

Table 4.1: Formative interview participants.

Interviewee	Self-reported Persona(s)
P1	Researcher: data analyst
P2	Researcher: summary data consumer Researcher: data analyst Researcher: data miner
P3	Clinician Researcher: summary data consumer Researcher: data analyst
P4	Researcher Clinician
P5	Researcher: data miner
P6	Clinician Researcher: summary data consumer Researcher: data analyst
P7	Researcher: data analyst Researcher: data miner Researcher: summary data consumer
P8	Clinician Researcher: summary data consumer
P9	Researcher: data analyst
P10	Pathologist Researcher

be limited due to shared assumptions and processes among consortium members. On the other hand, the consortium membership context helps to ensure a common understanding of data types, experimental capabilities, and shared terminology. Next, while interviews allow for in-depth conversations and follow-up questions, interviews are challenging to perform at scale (compared to a survey or literature review, for example). While in some cases interview participants shared examples of figures, papers, or interactive tools to explain their responses, our methodology was also not designed to observe experts conducting their data analysis work. Finally, we anticipate that participant responses will be informed by current solutions (experimental, data, statistical, visualization) that exist, despite potentially being suboptimal in addressing expert goals, but as such a study is not performed in a vacuum this limitation is not easy to overcome.

4.5.2 Formative Interview Results

Through qualitative analysis of interview transcripts, we identified recurring analysis tasks and goals articulated by participants in the context of conducting comparative analysis of multiple single-cell resolution datasets (including spatial and imaging datasets). Interview participants responded to interview questions in domain-specific terms (Section H.2). As proposed by Munzner[316], here we translate tasks and goals to more domain-agnostic terminology.

Analysis Goals

Through the interviews, analysis goals were articulated that were both confirmatory and exploratory in nature. In confirmatory cases, the user enters the analysis with a

hypothesis that one or more biomarkers of interest are associated with an outcome of interest; the goal of visualizing the data is to determine whether or not to reject (*i.e.*, verify) their hypothesis. In exploratory cases, the user may have a hypothesis that some biomarkers exist which are associated with an outcome of interest, but the user does not have specific biomarkers in mind *a priori*; the goal of visualization is to generate hypotheses. As an orthogonal consideration (*i.e.*, in both confirmatory and exploratory cases), users may (in some cases) aim to be critical of disease state annotations due to uncertainties about whether clinical annotations are reflective of the underlying molecular etiology (*e.g.*, what is currently binned into a single disease (or disease subtype) may in reality be a continuum or have multiple molecular causes that result in one clinical presentation). Thus, user goals may be to understand biomarker expression distributions regardless of sample-level annotation (*e.g.*, in which cell types or spatial regions is the biomarker expressed) or, in the absence of a biomarker of interest, identify biomarkers for a cell type or spatial region of interest. Alternatively, users may aim to identify samples or donors which exhibit certain expression patterns.

Analysis Workflow

Based on the above analysis goals, we propose a workflow comprised of four tasks (Table 4.2).

Table 4.2: Analysis workflow.

	Biomarker(s) of interest (targeted / confirmatory)	Lack of biomarker of interest (untargeted / exploratory)
Stratify samples (case vs. control)	**T2.1**: targeted w/ stratification	**T2.2**: untargeted w/ stratification
No stratification	**T2.3**: targeted w/o stratification	**T2.4**: untargeted w/o stratification

Users aim to navigate among these tasks in a single tool that offers functionality for both confirmatory and exploratory analyses. For example, a user may begin with an untargeted analysis, identify a gene or pathway (*i.e.*, biomarker) of interest, and transition to a targeted analysis upon selection of the biomarker.

Analysis Tasks

We describe in detail four main tasks articulated by interviewees and thus those which we have designed our visualization system to support. The first and second tasks entail stratification of the samples based on user-selected case-control group pairs of interest (based on sample-level metadata). The third and fourth tasks do not stratify the data into sample groups, to account for the aforementioned uncertainties around whether clinical variables at the sample level align with disease processes at the molecular level.

Here, we also discuss subtasks which require spatially-resolved data, despite focusing on non-spatial data for the initial system implementation. Users primarily articulated comparisons involving the following types of items: biomarkers, pathways, cell

types, spatial regions/patterns/images, and samples/donors. We use the typology of Brehmer and Munzner to describe the tasks and subtasks in abstract terms, referenced using `monospace` font (as also done by Somarakis *et al.*[393])[95].

All four tasks comprise `discovery`, as their goals are to generate or verify hypotheses (rather than to `present` or `enjoy` information). Note that we refer to hypothesis generation as exploratory and hypothesis verification as confirmatory. We consider tasks in which the user selects biomarker(s) of interest ahead of time to be confirmatory in nature, as the user may enter the analysis with a hypothesis in mind (*e.g.*, that a gene reported to have a particular function in the literature is upregulated in a certain sample group or cell type).

T2.1: Cohort comparison based on the expression/abundance of one or more biomarkers of interest. This task involves stratification of the data into sample groups (*i.e.*, cohorts) and is targeted (to the user-selected biomarkers of interest). For instance, users aim to understand whether and how expression of a gene or transcript differs between groups, and, if so, in which cell type(s) the expression patterns differ. The user first `locates` cell types in which the biomarker is expressed and then `compares` the cell type-specific expression distributions between cohorts. The output of this task may be the rejection of the hypothesis that the biomarker expression distribution is the same between case and control groups. Alternatively, the output of this task could be the list of cell types in which the expression distribution differs between the case and control groups. If the user has selected multiple biomarkers of interest, they may `compare` the biomarker distributions to each other, either rejecting the hypothesis that the biomarkers vary together or obtaining a list of one or more biomarkers which are not correlated with the remaining selected biomarkers.

When spatially-resolved data is present, the user may additionally be interested to understand whether spatial patterns of the biomarker expression differ between the sample groups. If a characteristic spatial pattern is known, this would involve `locating` the pattern in samples from each group and `comparing` the pattern instances; otherwise, the user must first `explore` the spatial data to `identify` patterns of interest prior to `comparing` instances between groups. Thus, the output of this process may be the rejection of the hypothesis that a spatial pattern is exhibited consistently among sample groups, or alternatively may be the instances or `locations` of a newly `identified` spatial pattern that is sample group-specific along with a `summarization` of the properties of the pattern.

T2.2: Cohort comparison based on significant differences that have been identified using statistical tests. This task involves stratification of the data into sample groups (*i.e.*, cohorts) and is untargeted (*i.e.*, the user has not selected biomarker(s) of interest). Upon comparison of these groups, the user aims to `identify` biomarkers, pathways, cell types, or spatial patterns exhibiting significant differences between the cohorts. This identification, leveraging differential expression results, could produce outputs such as a list of biomarkers or a list of the cell types expressing them. Alternatively, through visualization of gene set enrichment test results, users may aim to `identify` biological pathways or gene sets associated with differential expression findings, providing outputs that include lists of relevant pathways, their member genes, or the cell types in which certain pathways are enriched. Outside of biomarker- or pathway-specific findings, users may aim to `identify` cell types whose abundance significantly differs between groups, either generating a list of these cell types or determining whether to reject the hypothesis that the condition

of interest (*i.e.*, that stratifies the cohorts) affects cell type composition (overall or for specific cell types). Finally, with spatially-resolved data (and spatially-informed algorithms), users may `identify` or `locate` spatial patterns that are enriched in one cohort over the other.

T2.3: Identify cell types based on the distribution of expression/abundance of one or more biomarkers of interest. This task does not involve stratification of the data into sample groups but is targeted (to the user-selected biomarkers of interest). Through visualization, users aim to `compare` cell types according to the distribution of the levels of the user-selected biomarkers. This comparison of cell types may result in outputs such as a list of `identified` cell types of interest or a conclusion regarding the homogeneity (*e.g.*, in the case of a housekeeping gene) or heterogeneity of the biomarker distribution among these cell types (rejecting or not rejecting the users' hypothesis). Alternatively, this task could involve `comparisons` among donors or samples to assess the variability in biomarker expression or abundance, or to `identify` samples of interest (*e.g.*, outliers). Finally, with spatial data, the user may `identify` or `locate` spatial patterns of the selected biomarkers (regardless of stratification of samples into cohorts), highlighting how biomarker expression varies spatially across any or all samples.

T2.4: Comparison of cell types (or niches or FTUs) based on significant differences that have been identified using statistical tests. This task does not involve stratification of the data into sample groups, nor is it targeted to biomarkers of interest. Users aim to `identify` biomarkers or pathways that exhibit significant differences when `comparing` one cell type (or spatial neighborhood or func-

tional tissue unit (FTU)) against others. Thus, the output may be a list of biomarkers or pathways that are expressed or enriched, respectively, in a cell type specific manner. Alternatively, the output of this task may be the list of cell type(s) in which certain genes or pathways were expressed or enriched. With spatial data, users aim to `identify` or `locate` spatial patterns that are enriched in particular cell types, regions, or FTUs relative to others.

4.5.3 Survey on interest in comparisons in reference context

The aforementioned formative interviews were conducted with members of the KPMP consortium, representing a disease-focused context (kidney disease). As expected, the members of this consortium were interested in comparing samples based on disease-related metadata. As we were further interested in understanding the types of comparisons that users aim to conduct in the absence of disease-related metadata, we surveyed members of the Human BioMolecular Atlas Program (HuBMAP) data analysis working group. The survey questions asked participants to rank fifteen metadata variables from one to five (first- to fifth-most interesting), as well as to select a least-interesting variable for comparative analysis (Section H.3). We only asked participants about metadata variables available within HuBMAP, as we aimed to use the responses to determine which comparisons of HuBMAP data to perform. As some of the variables contained large fractions of missing values (which was indicated in the survey), we note that this missingness could impact the responses. In other words, our survey results may not generalize to situations in which metadata values are more complete or in which additional variables are available.

Nine participants responded to the survey, five of whose roles were Professor/Principal Investigator, one Postdoctoral Fellow, one Graduate Student, one Staff, and one

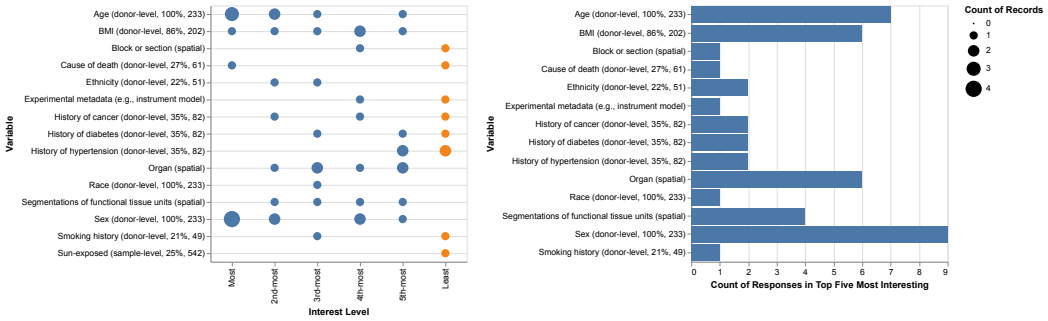

Figure 4.7: Results of survey of HuBMAP consortium members about comparisons of interest. Left) Count of responses for each interest level and variable. Right) Total number of responses in the top five most interesting levels per variable. BMI: body mass index. Parenthetical information represents the type of variable (*i.e.*, the type of entity the variable describes), the completeness as a percentage of entities, and the completeness as a number of entities.

Other ("NIH Program"). Four respondents indicated that they were experts in a particular organ or tissue type, four indicated expertise in biology, three indicated expertise in bioinformatics, one indicated expertise in computer science, and one had expertise in anatomy. The survey results indicate that participants were most interested in comparisons of sex (9 of 9 participants included this in their top five), age (7 of 9 in top five), organ (6 of 9 in top five), and body-mass index (6 of 9 in top five) (Figure 4.7). All but one of the least-interesting variable selections were accompanied by at least the same amount of top-five-most interesting selections for the corresponding variable, with the exception being the sun-exposed variable (which one participant indicated as least-interesting and zero participants indicated as top five most-interesting). Thus, we can use the top-ranked variables to prioritize comparative data analysis and visualization efforts among HuBMAP datasets.

4.5.4 Visualization System

We provide an overview of the user interface of the visualization system before elaborating on visualization designs.

Biomarker Selection Interface. To facilitate the analysis workflow (Table 4.2), we implemented an interface that enables users to specify biomarkers of interest and sample group pairs of interest (or lack thereof). The design of the biomarker selection interface is informed by the observation that interview participants articulated an interest in asking questions not only at the gene level, but also at the protein, pathway and cell type levels. Given that single-cell data is generally not mapped to an alternative data modality (*i.e.*, it exists in feature space of the original experimental modality), we implemented an interface that enables a user to specify a biomarker in a modality-agnostic manner (Figure 4.8). Then, the user can map their modality-agnostic query to a modality-specific one. For example, a user who first queries for a pathway of interest will subsequently be prompted to select an experimental modality of interest, say transcriptomics, and will then be offered a list of member genes included in the pathway to include or exclude, completing their modality-specific selection.

Application Layout. We designed an application layout in which the user can scroll vertically among multiple interactive visualizations in the center of the page (Figure 4.9). Along the top of the application, we display a header containing the current selections (of sample group pair and biomarkers) for context. Along the right side of the application, we display a cell set manager (allowing users to select cell type(s) of interest or an alternative cell type annotation column) and a sample group

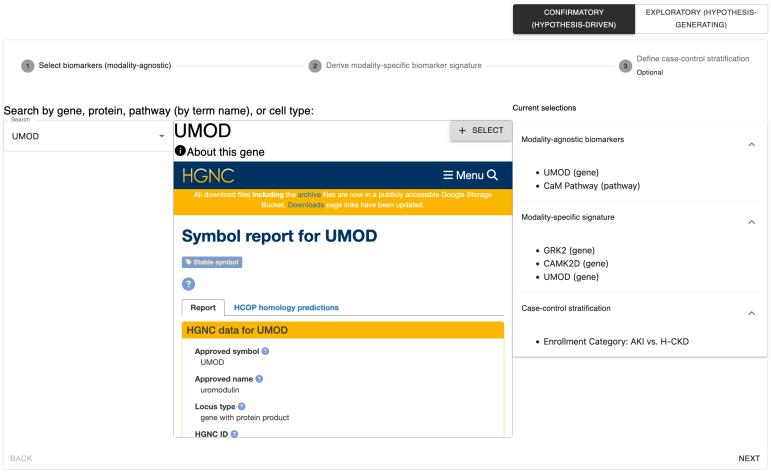

Figure 4.8: Biomarker selection interface, in the first step of the confirmatory workflow. Upon searching for a gene or pathway, an embedded webpage (*e.g.*, from HUGO Gene Nomenclature Committee or Reactome) displays information about the biomarker. In the confirmatory workflow, this wizard-style interface prompts users along three steps: Select biomarkers (modality-agnostic), Derive modality-specific biomarker signature, and Define case-control stratification.

manager (allowing users to select an alternative sample group pair for comparison). The heading and right sidebar are sticky, remaining in fixed positions as the user scrolls. In the left sidebar, usage instructions and additional context (*e.g.*, citations of analysis methods used) are displayed for each visualization. Finally, this layout is flexible and is not coupled to Vitessce. Instead, we provide a mechanism to obtain a "handle" to each Vitessce view using a unique view ID. Developers can use this mechanism to arrange the comparative visualizations flexibly within other applications (*e.g.*, data portals) requiring alternative layouts or styling.

4.5.5 Visualization Designs

In this section, we describe the visualization designs included in this system and discuss how each design facilitates one or more of the identified tasks.

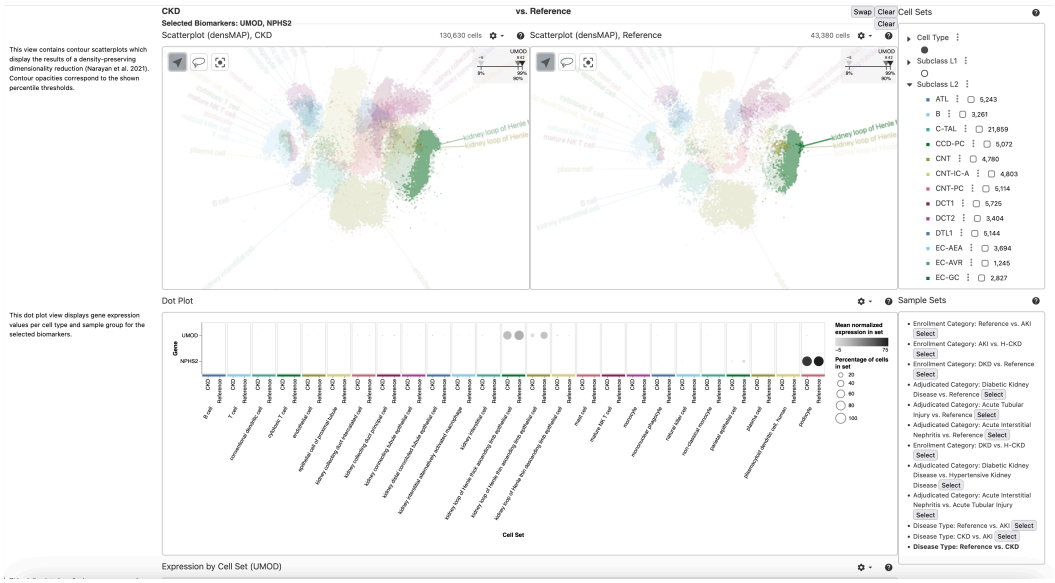

Figure 4.9: Application layout. The application consists of three columns. Left) The left sidebar contains information about each interactive visualization, including instructions related to interactive features and context about data processing details. Middle) The middle column of the application contains the interactive visualizations. In some cases, case-versus-control comparisons are displayed using side-by-side views (as in the contour plots shown here), while in other cases the comparisons are displayed using inner plot axes (as in the dot plot, which first groups the data by cell type along the horizontal axis). Right) The right column of the layout consists of "sticky"-positioned sidebar elements, including the cell set manager and the sample set manager views. Top) Along the top of the application, the sticky-positioned header element displays the currently-selected sample groups and biomarkers.

Dimensionality Reduction Contour Plots

While aspects of the usage of dimensionality reduction for single-cell data continue to be debated (Section 1.9.2), the fact is that data analysts are accustomed to using dimensionality reduction visualizations to obtain overviews of single-cell datasets. We know this not only from feedback during the formative interviews conducted in this study, but also during user interviews for the Polyphony project (Section 4.4), which are consistent with our finding that dimensionality reductions comprised over 23% of subfigures in our survey of visualizations contained in atlas publications (Section 2.3.2). During the interviews, participants reported interest in simultaneously visualizing gene expression values for multiple cell types (**T2.3**) across cohorts (**T2.1**), for example, noting, "I want to see transcriptomic data in specific cell types by patient characteristics and disease stage." However, participants also reported challenges such as, "We are stuck with this problem...We cannot view the cell type annotation and the expression values...at the same time, since we need to use colors for both. Unless you have two side-by-side..., one for the expression and one for the cluster," (P2). In the case of cohort comparisons, as we aim to display the data for two cohorts side-by-side, we did not want to further use the visual channel of juxtaposition to address the problem of simultaneously encoding the cell type and gene expression variables. Drawing from the feedback on the design of the comparison view in the Polyphony project (Section 4.4.2), we knew that it could be difficult for users to relate the two variables to one another using small multiples, even across coordinated scatterplots.

Instead, we designed a visualization for the dimensionality reduction results that uses per-cell type contours rendered at different opacity levels based on the density of the data points and their gene expression levels (Figure 4.10). This approach con-

trasts with the dimensionality reduction contour approach used in Polyphony (Section 4.4.2) in several ways. First, we compute the dimensionality reduction using DensMAP[319], a density-preserving variant of UMAP, to ensure that weighted point densities are meaningful to visualize. In other words, since 2D point densities resulting from UMAP do not necessarily correspond to densities in the original high-dimensional space, when points are weighted by expression value and the mean value is computed along a 2D grid, these mean values will be influenced by point density in the 2D space despite these densities not being interpretable according to the UMAP algorithm (*i.e.*, UMAP preserves dataset structure by constructing a graph based on nearest neighbors of points in high dimensional space, as opposed to directly using the distances between points in its objective function), while the objective function of DensMAP contains an additional term which encourages the preservation of point densities between the low- and high-dimensional spaces. Next, we render three levels of contours with different opacity values, corresponding to three user-defined thresholds. When a biomarker such as a gene is selected, we define these thresholds in terms of percentiles, which we compute based on the entire distribution of gene expression values (Figure 4.10C2). The contour polygons are determined by weighting each data point (*i.e.*, cell) according to its gene expression value and taking the average value in each grid cell. In the absence of a selected biomarker, contour thresholds are defined based on cell counts along the grid. Finally, this approach allows rendering plots for each cohort side-by-side to enable comparison (Figure 4.9, Upper Middle).

In such a visualization, we further aim to enable identification of the cell type corresponding to each contour. While a categorical color legend fulfills this criteria, we also wanted to display text labels directly in the plot to avoid sole reliance on color and to render the labels close to their contours (not only improving acces-

sibility but also resolving the issue of having potentially more categories than colors in the colormap). However, our existing scatterplot text label positioning approach (Section G.9.2) was suboptimal, rendering labels over the contour data points and obscuring the visualization, especially in cases of long cell type names (Figure 4.11, Left). We therefore implemented an alternative method in which we position the labels around the outside of the data points (Section H.4.1 and Figure 4.11, Right).

Dot and Violin Plots

Interview participants noted that it can be beneficial for interactive visualizations to use familiar visual representations. One participant stated, "I think because it's so widespread, that method of visualization, UMAPs and violin plots make sense... It's easier for all of us to be presented with data in a way we are familiar with," (P8). When asked about which types of plots they use for different data types or to communicate particular types of findings, another participant noted usage of dimensionality reduction plots, violin plots, and dot plots, stating "A lot of people, with single-cell data, when we are looking at the expression pattern across cell types, it'll be either dot plots or violin plots. I think those are what we are most commonly doing, a lot of times for a first-pass analysis. I will also pull up – I don't know what you call them – I think they're called feature plots where points are color-coded on the UMAP," (P4).

Dot plots and violin plots in single-cell contexts typically display similar types of information. Whereas violin plots visualize the full distribution of expression values for particular stratified groups of cells, dot plots display summarized information about each distribution (mean expression value and percentage of expression values above a minimum threshold) and therefore easily visually scale to more than one gene. Yet as neither dot plots nor violin plots (or dimensionality reduction plots)

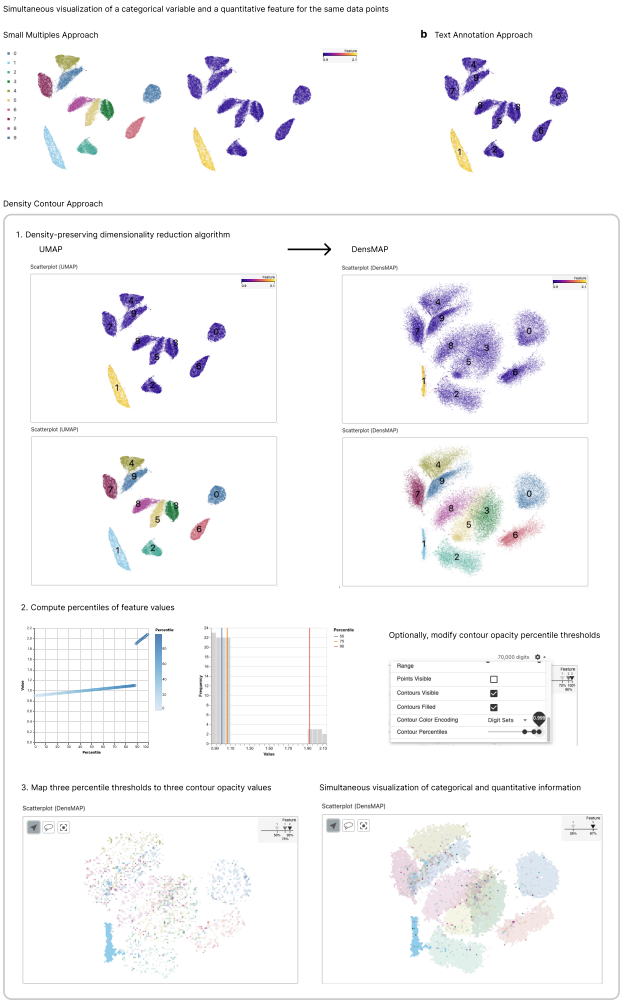

Figure 4.10: Designs for simultaneous visualization of quantitative and categorical information with dimensionality reduction results. These examples use the MNIST dataset of handwritten digits, with the digit one mapped to the value 2, other digits mapped to the value 1, and a small random jitter applied to all values. a) Small multiples approach: two scatterplot instances are used, the first in which the categorical variable is mapped to the color encoding, the second in which the quantitative variable is mapped to the color encoding. b) Text annotation approach: the quantitative variable is mapped to the color encoding, while the categories are overlaid using text positioned at the cluster centroid. c) Density contour approach: dimensionality reduction is performed using a density-preserving algorithm. Percentiles are computed for the quantitative variable value distribution. Three percentile thresholds are selected, which are mapped to three different contour opacity values. Contours are computed using a grid-based method in which each scatterplot point is weighted by the quantitative variable. The mean of weighted values for each category within the grid cell determines whether the values surpass the threshold for each of the contours such that the grid cell should be contained within a contour with the corresponding category color and opacity value.

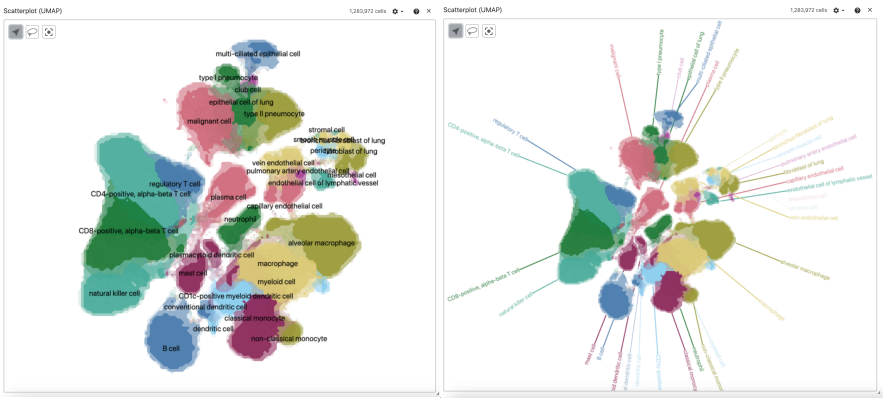

Figure 4.11: Text label positioning for dimensionality reduction contour plot. (Left) Force simulation-based positioning of text labels. (Right) Nearest circle point-based positioning of text labels.

visually scale to the full transcriptome (*i.e.*, over 20,000 genes for human datasets), these plots require that the user has selected one or more biomarkers of interest (**T2.1** and **T2.3**).

Typical dot and violin plots group cells by a single variable, cell type. As **T2.1** entails understanding how gene expression values differ not only by cell type, but also by cohort, we designed interactive dot and violin plots which display the data for each cohort side-by-side (Figure 4.12). In other words, the horizontal axes are hierarchical, first grouping by cell type before next grouping cells within each cell type by cohort. In the absence of a selected pair of cohorts, these plots display the data stratified only by cell type (across all samples in the dataset), supporting **T2.3** (Figure 4.13).

Volcano Plots and Gene Set Enrichment Bar Plots

As dimensionality reduction, dot, and violin plots can help users to understand distributions of several selected biomarkers (**T2.1** and **T2.3**), they are unable to help in the absence of selected biomarkers (**T2.2** and **T2.4**). Instead, in these settings

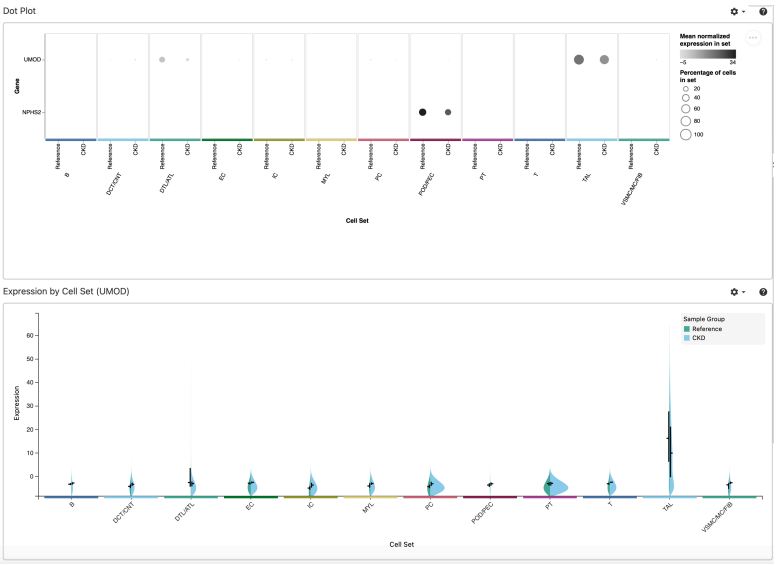

Figure 4.12: Comparative dot plots and violin plots. Data visualized in these plots is first stratified by cell type, then by sample group.

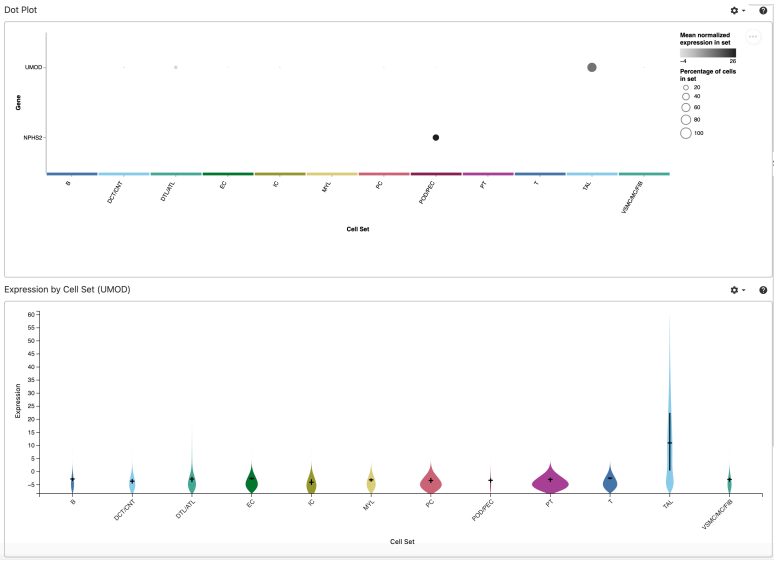

Figure 4.13: Dot plots and violin plots, in the absence of a selected pair of sample groups.

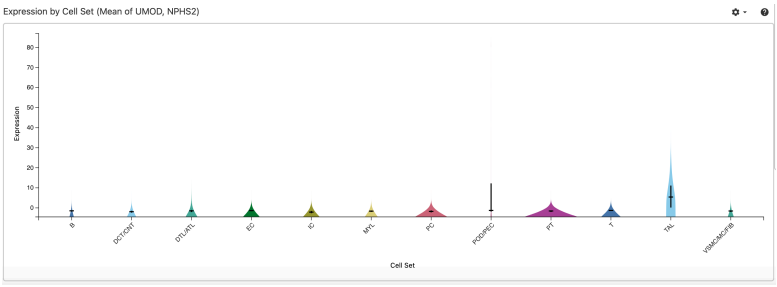

Figure 4.14: Violin plots displaying aggregate expression values per cell type for two selected genes. In this case, we aggregate the expression values for two genes by taking the mean for each cell. Users can select an alternative aggregation or summarization strategy, such as sum or difference (in the case of two selected features).

participants reported expecting other plot types, with one participant stating, "Pathway standard visualization is bar plots, where you have visualized the minus log10 P-value. If you go into the papers, most people show it this way. For differentially expressed genes, you can use volcano plots," (P5). Therefore, we designed interactive volcano plot and bar plot visualizations to display the results of differential expression tests and gene set enrichment tests, respectively.

We implemented these plots such that the comparisons displayed differ based on whether a pair of cohorts has been selected (**T2.2**, Figure 4.15) or not (**T2.4**, Figure 4.16). Under scenario **T2.2**, these plots display the results of a within-cell type comparison (performed for each cell type): considering only cells from cell type X, which genes are significantly differentially expressed between the case and control cohorts? In contrast, under scenario **T2.4**, these plots display the results of comparing cells within each cell type to cells outside the cell type (*i.e.*, cells in cell type X versus the rest of the cells), intended to help users identify cell type-specific expression patterns and enriched pathways.

We designed these plots to be interactive, so that users in scenarios **T2.2** and **T2.4** can transition into **T2.1** and **T2.3**, respectively. Upon identification of a biomarker

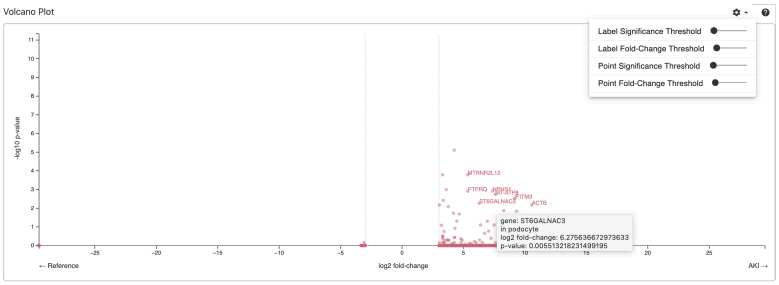

Figure 4.15: Volcano plot, case versus control, within cell type. The tooltip displays statistical information for a hovered point. The plot settings allow the user to control the thresholds used to filter data during rendering of points and text labels. The horizontal and vertical rules indicate the current point significance and fold-change thresholds.

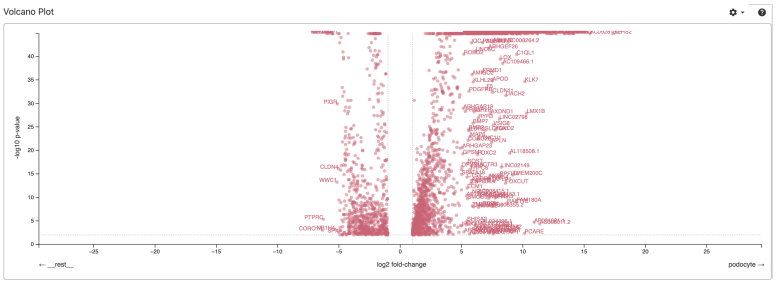

Figure 4.16: Volcano plot, one cell type versus rest. Arrows along the bottom left and bottom right of the horizontal axis indicate the direction of differential expression results.

or pathway of interest from the volcano or bar plot, the user can click the element to make a selection. Selection of a pathway results in selection of its member genes. Other interactive features in these plots include tooltips (Figures 4.15 and 4.17), enabling users to view the complete statistical information for each data point (*e.g.*, P-value, fold-change).

In developing these interactive plots, we noticed issues of over- and under-plotting that we had not foreseen at the design stage. Specifically, as we designed the volcano plot to display the results for each selected cell type simultaneously, overplotting (of both points and labels) quickly becomes an issue upon selection of multiple cell types (Figure 4.18, Top). Filtering plotted and labeled points based on P- and fold-change

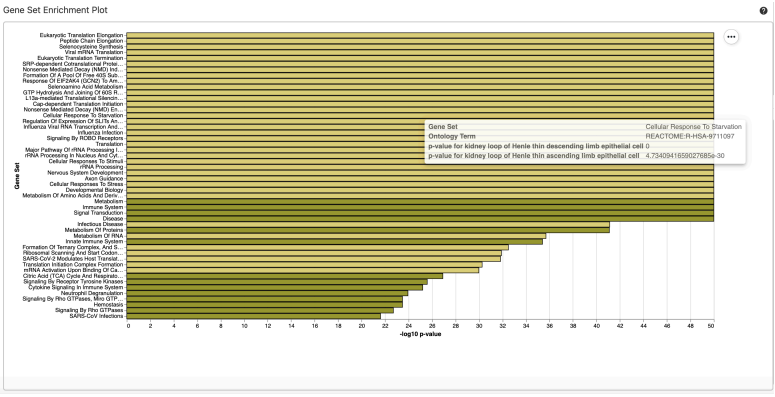

Figure 4.17: Gene set enrichment bar plot. Each bar represents a gene set that has been found to be significantly enriched in at least one cell type, following differential expression analysis. The bar color corresponds to the cell type in which the gene set is most significantly enriched, while the tooltip displays P-values for other selected cell types in which the gene set was also found to be significantly enriched. P-values of zero are clamped to minus log10-transformed values of 50.

values did not effectively resolve these issues. In the gene set enrichment bar plot, we encountered many P-values of zero (which we clamped to minus log10-transformed values of 50), resulting in under-plotting as we only display the K most-significant enrichment results. To account for these issues, we implemented an interactive tabular data view, in which the user can sort results by P-value or fold-change and navigate the paginated results (Figure 4.18, Bottom).

Cell Type Composition Bar Plot

We know from Section 1.5.5 that single-cell data is useful in part because changes in gene expression programs can be distinguished from shifts in cell type composition. Thus, we designed a visualization to display the results of the ScCODA method for compositional analysis of cell types[101]. We developed an interactive bar plot in which the bar encodes the log-fold change in cell type abundance between the selected pair of cohorts, facilitating **T2.2** (Figure 4.19). Tooltips display additional statistical infor-

Figure 4.18: Volcano plot and tabular view. Top) The volcano plot displays differential expression results for all selected cell types simultaneously, resulting in an overplotting issue. Bottom) The tabular view displays the same data visualized in the volcano plot, with controls for column sorting and row selection.

mation produced by the ScCODA method. As our data processing pipeline runs this analysis for each specified cell type annotation column, the bars in the plot reflect the currently-selected cell types and their granularity.

Treemap Plot

We implemented an interactive treemap plot to provide context to users about the amount of cells which correspond to their current selections. While this plot does not directly address one of the four identified tasks, it aims to help users understand roughly whether there is enough data to ask their biological question and the extent to which the amounts of cells in each cell type and sample group are balanced. In this treemap, each rectangle area corresponds to the number of cells in the specified category, such as endothelial cells within CKD samples. The height of the treemap

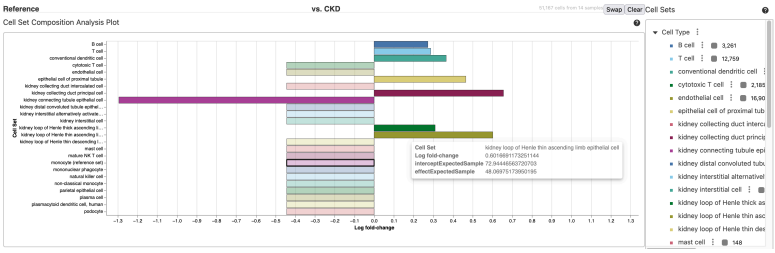

Figure 4.19: Cell type composition bar plot. This view displays the results of a cell type composition analysis performed using the ScCODA algorithm[101]. Cell types with significantly different composition between the selected pair of sample groups are displayed with opaque bars, while not-signficant results are displayed with transparent bars. The single outlined bar denotes the automatically-selected reference cell type that was used for analysis.

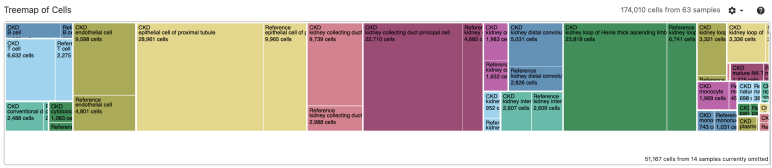

Figure 4.20: Treemap plot, colored by cell type. Rectangle area encodes the number of cells in the (cell type, sample group) category. In this case, rectangles are first organized by cell type, then by sample group.

(within the outer rectangle of the visualization) corresponds to the ratio of the number of currently-selected cells over the total number of cells contained in all samples in the integrated dataset. Interactive features of this plot include tooltips (to display full text labels when cut off), click interactions (to select a cell type of interest), and options to change the primary hierarchy level (nested by cell type first or sample group first) and to change the color encoding from cell type (Figure 4.20) to sample group (Figure 4.21).

4.5.6 System Implementation

The comparative visualization system consists of interactive visualizations analysis results that have been precomputed by a data processing pipeline and stored in an An-

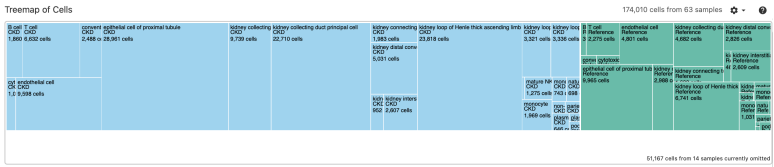

Figure 4.21: Treemap plot, colored by sample group. In this case, rectangles are first organized by sample group, then by cell type.

nData object with comprehensive metadata outlining the comparative analyses that have been performed. The interactive visualizations have been implemented as views in the Vitessce framework (Chapter 3), using the DeckGL[433], Vega-Lite[378], and D3[87] JavaScript libraries. Further, these developments involved extending the Vitessce data model to support mapping cells to samples, storing sample-level metadata and comparison metadata, and representing the outputs of differential expression, gene set enrichment, and cell type composition analyses. Specifically, we extend the data model with the following *data types*:

- `sampleEdges`: Tuples of (observationId, sampleId) to map observations (*e.g.*, cells) to samples.
- `sampleSets`: Lists or hierarchies of sets of samples.
- `featureStats`: Per-feature statistics. For example, differential expression test results.
- `featureSetStats`: Per-feature-set statistics. For example, gene set enrichment test results.
- `obsSetStats`: Per-observation-set statistics. For example, cell type composition analysis results.
- `comparisonMetadata`: Comparison metadata.

To enable connection to ontologies, for instance to retrieve the list of member genes in a given pathway, we extended the Vitessce plugin API to support plugin asynchronous function calls for a predefined set of *function types* with well-defined function signatures. For example, we define a function type `transformFeature` which takes as input a feature such as a pathway ID as well as the target type, such as `gene`, and returns a corresponding list of target entities (in this case the member genes of the pathway). We use knowledge graph files from EnrichrKG to implement this functionality[153]. While we define default functions of each type, the plugin architecture allows users of Vitessce to define their own `transformFeature` function, for example using a different ontology source, overriding the built-in implementation. Other function types include `autocompleteFeature` (used by the search interface) and `featureToUrl` (used to embed a descriptive webpage about a feature, such as a GeneCards page describing a gene; Figure 4.8). We anticipate that these types of plugin asynchronous functions could be useful in other scenarios, especially those in which the field lacks an established data model, such as mapping sample IDs to image files to support multi-sample spatial data analysis, which are often stored in an atlas- or consortium-specific way.

Data Processing Pipeline. We implemented a comparative data processing pipeline to showcase the comparative visualizations and facilitate their development. The data processing pipeline takes a pre-integrated AnnData object as input, with the expectation that the results of the data integration process have been validated to ensure that batch effects have been minimized such that downstream analyses can be appropriately trusted. In addition to this AnnData object, the pipeline requires specification of the column containing sample identifiers, specification of one

or more columns containing cell type annotations, as well as the definition of pairs of sample groups to compare (Figure 4.22). As the pipeline runs, it executes analysis functions from Scanpy[448] and Pertpy[203], and produces analysis results in the form of data frames and arrays. These data frames and arrays are stored within the resulting AnnData object, along with the analysis metadata (*e.g.*, analysis type, groups being compared, cell type column used, and other analysis parameters). Figure 4.23 demonstrates a real-world example in which pipeline execution specifying 12 case-control group pairs for a dataset containing 25 cell types generates approximately one thousand data frames containing analysis results.

The pipeline steps can be run either sequentially or in parallel, to support limited-resource Jupyter notebook environments and cluster computing environments, respectively. The parallel functionality is implemented using Snakemake[308]. To reduce the memory requirements of the pipeline and enable parallel reading and writing of data, we implement a lazy-loading AnnData wrapper class which supports partial reading of data as needed, as well as partial writes of data to Zarr format (Section D). We modified the Scanpy[448] library to support computations on Zarr and Dask arrays, as well as to use the DensMAP dimensionality reduction algorithm (which we have contributed back to the main Scanpy code repository)[319]. Finally, we emphasize that this pipeline implementation is intended as a prototype and does not necessarily use the best-performing analysis methods (*e.g.*, informed through benchmarking studies). Rather, this pipeline is meant to generate sufficient analysis results to showcase the visualization features, enable user evaluations, and serve as a testbed for developing a specification for comparative metadata representation.

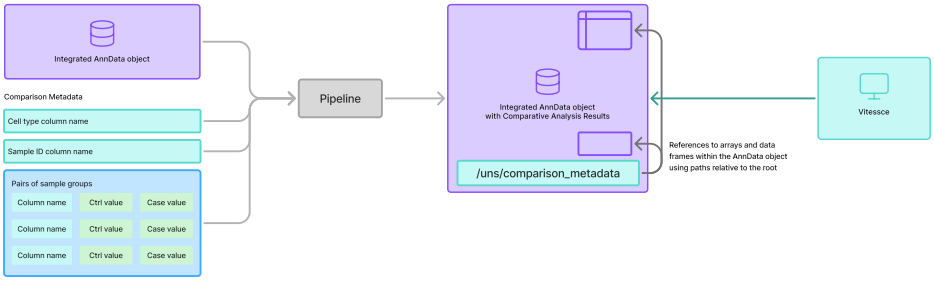

Figure 4.22: Comparative analysis data processing pipeline. Required inputs are an AnnData object containing multiple single-cell experiments that have already been integrated (batch-corrected) and verified, plus comparison metadata including the names of one or more cell type columns, the column containing sample identifiers, and tuples defining pairs of sample groups to be compared, defined in terms of sample-level columns and pairs of their values. Once computed, arrays and dataframes are stored within the resulting AnnData object. A new `comparison_metadata` field within the "unstructured" section of the object references these arrays and dataframes using relative paths, enabling the downstream visual analytics system (in this case, Vitessce) to locate results of interest.

Comparative Metadata Representation. Visualizations of the results of comparative analyses require information about the provenance of those analyses, including details related to the input data, data transformations applied, and the method (along with any parameters) used. This information should not only be shown to users for context, but also enables identification of particular subsets of data to load, given the biological questions of a user. However, there are not yet standard or widely-adopted approaches to store the results of comparative analyses. As noted by Hrovatin *et al.*, "There is a need to define the type of analysis results that should ideally be provided by reference atlases, and to develop the necessary infrastructure to share these outputs,"[207]. To address this problem, we have defined one such approach in which we store metadata about the comparative analyses that have been performed within the unstructured ("uns") section of an AnnData object (Section H.5). This metadata

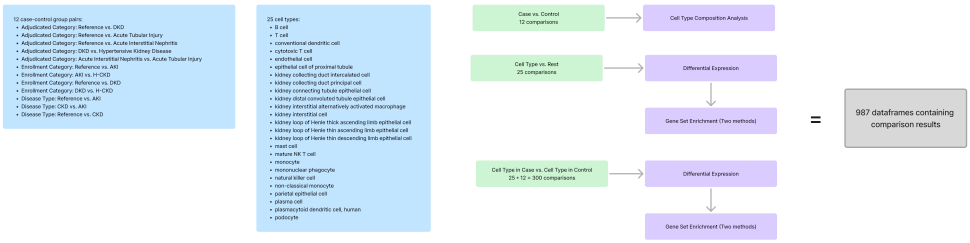

Figure 4.23: Comparative analysis data processing results. The pipeline executes multiple types of analyses, including cell type composition, differential expression, and gene set enrichment. The total number of resulting dataframes depends on the number of sample group pairs and the number of cell type categories.

refers back to result-containing data frames or arrays stored within the AnnData object.

4.5.7 Evaluation

We have evaluated this system through two case studies and two expert interviews.

Case Studies

We demonstrate the system through two case studies.

Kidney disease atlas. In this case study, we use a kidney atlas dataset from the KPMP consortium. This dataset has been previously integrated and analyzed in depth[250]. This dataset contains 255,177 cells from 77 samples. The dataset contains three cell type annotation columns, with differing levels of granularity, as well as three disease subtype columns (Disease Type, Enrollment Category, and Adjudicated Category). Disease type is a coarse categorization, while enrollment category represents a finer-grained tissue donor annotation that was defined at the time of study en-

rollment. Meanwhile, adjudicated category represents the disease subtype annotation following an adjudication process which considers not only electronic health records, but also review of the kidney biopsy and consensus among a panel of nephrologists and pathologists.

Through comparison of Reference and Acute Kidney Injury (AKI) samples (based on Enrollment category), we can reproduce findings from Lake *et al.*[250], for example, the downregulation of genes such as FOS and JUND in AKI – genes which are known from mouse studies to be protective for AKI[127]. These genes appear in the top results within the differential expression table, and selection of each gene of interest enables expression levels to be compared in the contour, dot, and violin plots. Using the cell type composition plot upon selection of level-two cell type annotations, we see that the adaptive proximal tubule (aPT) cell type is significantly more abundant than expected in the AKI condition compared to the Reference condition. This finding is also consistent with that of Lake *et al.*[250] who note that the aPT cells share characteristics with cells in failed repair states in rodents. In the exploratory mode, other top results in these visualizations were not discussed by Lake *et al.*[250], such as WFDC2, which exhibits upregulation in AKI samples, consistent with literature which identified the gene as a biomarker of renal fibrosis[55].

Heart dataset. In this case study, we use a human heart dataset from the HuBMAP consortium. This dataset is published as a consortium "data product" as it has been integrated by the data coordination center within the consortium and contains data from all HuBMAP samples for a particular organ (in this case, heart) that have been profiled by single-cell transcriptomics. The dataset contains 39,181 cells (post-quality control and filtering) from 24 samples. As HuBMAP profiles non-diseased tissue sam-

ples, we focus on those types of comparisons articulated by consortium members (Section 4.5.3) and sample groups containing at least two tissue samples. The dataset contains cell type annotations from two different methods and four donor-level metadata variables: sex, age, race, and body-mass index (BMI).

As this dataset has not been analyzed in a previous study, we perform a visual analysis with a more exploratory workflow. First, we use the treemap visualization to explore how the amounts of cells in different pairs of sample groups are balanced, selecting different pairs using the sample set manager in the right sidebar. Next, we identify that age group (Below 50 vs. 50 and Above) and body-mass index (Overweight vs. Obesity) pairs have relatively balanced distributions of cell types between cohorts. Focusing on the case in which cohorts are stratified by body-mass index groups, we first look to the cell type composition plot and find that fibroblasts are slightly more abundant than expected in the overweight group than the obesity group. Selecting this cell type enables us to filter the differential expression results to the fibroblast cell type results. Using the volcano plot, we identify the gene COL6A2 to be significantly differentially expressed in the fibroblasts in the obesity group. Selecting this gene, we confirm the difference in expression patterns between cohorts using the contour, dot, and violin plots. Looking to the biological literature (via a web search outside of the visualization system), we see that COL6A2 is known to be expressed in fibroblasts in heart tissue samples and associated with both body-mass index and coronary artery disease risk[459].

Expert Interviews

Thus far, we have conducted user interviews with two participants (with these two participants previously having been part of the ten formative interviews): one fol-

lowing the design stage, and one following the preliminary implementation, roughly following Munzner's nested model[316] (Section 1.10).

Post-design evaluation. In the formative interviews, we asked participants about their domain-specific questions and the data types used to answer these questions. Following the nested model, the answers to these two types of questions facilitated the domain problem characterization and data abstraction design stages, respectively. Following analysis of the formative interview transcripts (Section H.2), we designed the system workflow and designed mockup interfaces and visualizations using Figma[14]. Where possible, we incorporated static visualizations produced using real KPMP data, generated using Matplotlib, Seaborn, and Altair[421], into the Figma mockups to roughly indicate planned visual encoding choices. We used the commenting features of Figma to annotate visualizations and other interface elements with relevant quotes from the formative interviews, to clearly indicate the domain-specific questions motivating their inclusion (as well as to identify elements lacking evidence for inclusion).

Upon completion of the mockups, we conducted an interview with one participant (P7) to obtain further feedback prior to software implementation. This participant confirmed that the presented system design would satisfy major use cases, stating, "I think the exploratory [design] all works. If you sort of stick to the statistical and analysis approaches that are used in the landmark publications that we have [in KPMP], I think in that case, it's probably fine. I think the, you know, selecting the biomarker, that's probably going to be the biggest use case. And this all makes perfect sense... It's fine that this would be limited to things that you know we – or the users – think are going to be the high-value questions that people want to ask." However, this par-

ticipant also noted that they would be inclined to perform certain types of analyses outside of the context of such a visualization tool, stating, “I think it’s when you start getting into some of the more detailed stuff. Or, you know, for reasons statisticians may be interested. They may want to take a different statistical approach to look at things, and in that case you want to have either the data in-house, or sort of in some kind of enclave that you sort of access and run your own algorithms on.” When asked follow-up questions about how detailed of an analysis the participant might expect from this type of visual analytics tool, they noted that they would expect two levels of cell type granularity as well as the ability to select particular cell types and to select case and control cohorts of samples, stating, “I mean, I think it’s reasonable, maybe it’s level 2 [cell types]... I don’t think it makes sense to go deep into the discrete subtypes. I think that, sort of, whether it’s marker genes that show up for the podocyte [cell type], in kind of a case-versus-control situation, I think that probably would make sense from an analysis standpoint. So yeah, it’ll be constrained to some level of data. But I think for people that use the web app, that’ll probably answer most of what they’re looking for, and then, for ones that want to get deep under the hood, they can navigate to an enclave or download the data. That will be where people set their sights. The participant then asked clarifying questions about the spatially-resolved visualization designs, with them concluding that these would benefit from a series of video tutorials. We also discussed whether and how to aggregate data from multiple biomarkers, with the participant noting that conclusions drawn based on multi-gene signature scores would require validation. The participant also noted that there might be sufficient data to perform only a subset of comparisons, stating, “It’s really gonna depend on how many cells get sequenced and annotated for a given cell type in patients.” Towards the conclusion of the interview, the partic-

ipant reported, "I think it's hitting the major stuff," and "I think that covers what you would want to look at as a biologist or a scientist who isn't necessarily adept at doing their own coding...This was really cool to look at." Considering this preliminary feedback, we proceeded to the software implementation stage.

Post-implementation evaluation. The first stage of software implementation has been focused on those parts of the system designed for non-spatially-resolved transcriptomics data. We conducted a post-implementation interview to evaluate the system at this stage with one participant (P8). This participant initially reported quite positive feedback, stating, "I think the tools you have here are fantastic. I'm super excited, especially the ability to throw some genes together, create a dot plot, compare it between a couple things, and boom, make a figure. It is very helpful. The discussions and the sharing get going faster this way." As more features of the system were demonstrated to the participant, they asked questions which prompted requests for certain changes and improvements. For instance, the participant requested that more context about the analyses performed to generate the data displayed in the visualizations be shown, stating, "Now, there are many differential expression approaches. Which one was used here, or was that done before you got the data?... It's just good to know which [method was used]." At another point, the participant noted the need to clearly indicate when data points have been filtered and thus hidden from a plot, stating, "I think you do need to be careful. I don't know what the help question-mark icon says... but the idea that not everything that's significant is being shown... how would you know? Would you just assume that it's not significant in this data? The user needs to know." Another topic which came up during this interview (which also came up during the post-design interview) was how to handle multiple selected genes

in visualization designs that are less scalable to multiple features (*e.g.*, per-cell type violin plots). When asked whether they expect violin plots to display expression distributions for a single gene across cell types versus for a single cell type across genes, the participant noted, "Both situations are helpful for different approaches to a question." We discussed one idea of addressing this problem through a more interactive dot plot, with the participant stating, "If you have several genes on the dot plot, you might be able to manipulate the violin plot without redoing everything, so you can just go through and interrogate them and look at them," or a standalone gene selection interface, which the participant also considered to be a "great idea." This interview concluded on a lighthearted note with the participant stating, "I think this is awesome... Send it to production already!"

4.5.8 Methods

Further method details can be found in Appendix H. This appendix includes the formative interview methodology and summarizes the results in the form of recurring questions asked of users. Further, the appendix includes visualization implementation details, planned visualization designs, and information about the comparative analysis metadata format.

4.6 Discussion

4.6.1 Limitations and Future Work

Evaluation with additional users and user personas. Thus far, we have conducted two user interviews to evaluate this visualization system. While these were conducted with different interview participants comprising different user personas,

more work remains to evaluate the system with users, including those across all target user personas including pathologists. We anticipate that pathologists, for example, will be most interested in spatially-resolved and imaging data; therefore, it could be worthwhile to pursue these evaluations following the implementation of the comparative spatial and imaging visualizations.

On-demand analyses and pipeline scalability. The comparative analyses visualized in this project were performed on a high-performance computing cluster ahead of time, and in some cases the data processing took over 24 hours to complete for datasets containing hundreds of thousands of cells. While the pipeline steps can be run in parallel, the scalability can be further improved, for example running sub-analyses in parallel (*e.g.*, running differential expression tests for each cell type in parallel, as opposed to only running differential expression and cell type composition steps in parallel). While the optimization of the data processing pipeline was not a primary goal of this project, it represents important follow-up work.

Alternative data processing strategies can be explored, such as on-demand analyses that would enable comparative visualizations of datasets for which results have not been pre-computed. Such a strategy could also involve cacheing of results as they are generated to prevent duplicate computations for future users.

Adoption of comparative metadata standards. We address the problem of a lack of established comparative metadata representation by developing our own representation and metadata schema. While we have documented this representation (in Section H.5 and in the Vitessce documentation website) and provide examples, the adoption of this metadata format more broadly will require community buy-in, for

example, by cell atlas consortia or in alternative data processing pipelines.

Analyses agnostic to cell type annotation. Currently, we address the problem of cell type annotation granularity by running analyses for multiple cell type annotation columns. In the future, we aim to incorporate visualizations of the results of analysis methods which do not rely on cell type annotations, such as LEMUR[51] or Milo[130]. However, as not all users may be familiar with these methods and how to interpret their results, alternative approaches such as hierarchical analyses (*e.g.*, guided by the cell type ontology or hierarchical clustering) may be superior[329,212].

Suggestions of related analyses. Our visualization system focuses on exploratory and confirmatory workflows (Table 4.2). However, we envision that in both workflows, suggested analyses or next steps could assist users. For instance, during confirmatory analysis in which a user has selected a gene of interest, additional biomarkers could be suggested to the user for subsequent analysis. These suggestions could be informed by ligand-receptor interactions, co-expressed genes, or membership in a pathway or protein complex. Identification of certain trends in the data could also prompt suggestions, for example, of a different group of samples exhibiting a similar trend.

Implementation of additional visualizations. In the future, we intend to implement additional visualizations (and corresponding data processing steps) to support comparisons that involve additional experimental modalities (*i.e.*, beyond transcriptomics). These include spatial and imaging-based visualizations, as well as visualizations for chromatin accessibility and epigenomics data, such as plots of differential accessibility test results and genome-mapped visualizations. We also aim to incorpo-

rate visualizations of further data modalities available from the KPMP, such as urine biomarkers, which are of particular interest to clinicians. We intend to implement additional visualizations which support sample- and donor-level stratification, to enable users to better understand heterogeneity across donors and samples and to identify outlier samples or donors. Finally, we aim to provide more flexibility in how multiple biomarkers are visualized, to account for feedback that arose in both the post-design and post-implementation interviews, by implementing an interactive violin plot matrix (*i.e.*, as opposed to a single row of violin plots) with support for interactively transposing the matrix grid axes.

Comparative visualizations for spatially-resolved data. During the formative interviews, participants reported interest in asking several comparative questions of spatially-resolved and imaging data. For instance, "What does the spatial transcriptomics data look like for the selected biomarker? How does it differ between case and control groups?" We designed visualizations to answer these questions using representative sample and small multiple approaches (Section H.6.1), yet we focused on non-spatial visualizations for the preliminary system implementation. We further identified comparative questions that require delineation of functional tissue unit structures using segmentation (Section 5.5), for example, to answer questions such as, "What is the distribution of expression of the selected biomarker within glomeruli versus outside?" Finally, we emphasize that further work remains to integrate multiple classes of spatial data analysis methods (Section 5.3.1) into the comparative data processing pipeline and to subsequently design scalable and effective visualizations to display their results (*i.e.*, beyond representative samples and small multiples).

Cross-organ and cross-species comparisons. This project was conceptualized within the context of the KPMP, which is focused on a single organ and species (kidney, human). However, in other cell atlas contexts, such as the Human BioMolecular Atlas Program (HuBMAP), cross-organ and cross-species comparisons are key project goals (Table 1.1). We anticipate that our approach and visualization system could be extended to support these goals through modifications to the comparison metadata representation (*e.g.*, to capture organ- and species-level metadata) and data processing pipeline. Extensions to the internal Vitessce data model may also be required to represent homologous features across species (*i.e.*, mappings of genes between human and mouse).

Beyond pairwise comparisons. So far, we have focused on pairwise comparisons (*i.e.*, case-versus-control style). Additional research is needed to support comparisons of more than two groups, or comparisons based on continuous variables (*e.g.*, age). These scenarios introduce challenges in explicit visual encoding of comparison (*i.e.*, directly visualizing the difference between pairs of values). Supporting such scenarios will not only require new visualization techniques, potentially drawing upon our survey findings (Section 2.3), but will also require statistical methods that consider these types of comparisons.

4.7 Conclusion

This chapter has introduced and evaluated two interactive visualization systems designed to support comparative analyses of single-cell data, Polyphony and Compasce. Polyphony leverages an interactive transfer learning model and a client-server architecture, while Compasce relies on a comparative data processing pipeline to precom-

pute results. Both systems extend the Vitessce framework to facilitate interactive visualization. Through case studies and expert feedback, we have shown that these systems address the identified biological questions. Although thus far our evaluations have indicated strong initial support from domain experts, several avenues for future work remain, including improving server and pipeline scalability, integrating additional data modalities, and adding visualization capabilities for spatially-resolved data, cross-organ comparisons, and more complex comparative scenarios.

5

Exploration of interpretable spatial domains

5.1 Preface

This chapter presents contributions that I have made to multiple projects related to visual exploration of spatial -omics data, including three that have been described in

publications that I have co-authored:

- Viv: Multiscale Visualization of High-Resolution Multiplexed Bioimaging Data on the Web[288]
- OME-Zarr: a cloud-optimized bioimaging file format with international community support[313]

The Viv project was led by Trevor Manz with contributions from Ilan Gold, Nathan Heath Patterson, Chuck McCallum, me, Bruce W. Herr II, Katy Börner, Jeffrey M. Spraggins, and Nils Gehlenborg. The OME-Zarr paper was led by Josh Moore with contributions from Trevor Manz, Ilan Gold, me, Nils Gehlenborg, and many others.

Portions of this chapter have been adapted from the following in-progress manuscript for which I am first author:

- Interactive visualization of kidney micro-compartmental segmentations and associated pathomics on whole slide images

I led this segmentation visualization project with additional contributions from Nicholas Lucarelli, Yijiang Chen, Samuel Border, Andrew Janowczyk, Jonathan Himmelfarb, Matthias Kretzler, Jeffery Hodgin, Laura Barisoni, Pinaki Sarder, and Nils Gehlenborg. I thank Stephen Fisher for discussions around the standardization efforts for multi-channel segmentations in the HuBMAP consortium and for inspiring Figure 5.2b.

Section 5.6 describes the SpatialQuery algorithm which was developed by Shaokun An and Martin Hemberg. Contributions to integrate SpatialQuery into HuBMAP Workspaces were made by Shaokun An, Martin Hemberg, Thomas Smits, Nikolay Akhmetov, Austen Money, John Conroy, Lisa Choy, Tiffany Liaw, Yan Ma, and Morgan Turner.

5.2 Introduction

A challenge for cell biologists is understanding which structural patterns are associated with function, for instance, characteristic of health and disease. Historically, single-cell profiling techniques have required dissociation of tissues, but recent technologies have overcome this limitation and measure molecular states within cells while preserving spatial context (Section 1.5.6). With spatially-resolved 'omics data, a key analysis goal is to identify spatial molecular patterns associated with biological function using computational methods. Such methods can infer spatially-variable genes, ligand-receptor interaction patterns, or enrichment of neighboring cell types. In this chapter, we contribute interactive techniques for visualization of spatial single-cell 'omics data and downstream analysis results.

5.3 Related Work

We provide an overview of prior research into the development of analysis and visualization techniques for spatially-resolved single-cell data.

5.3.1 Spatial Data Analysis Methods

There are many analysis approaches for spatially-resolved single-cell datasets, and along with them, many different data representations for the results[334]. We provide an overview of the landscape of approaches in Figure 5.1. Approaches can be roughly divided into those which use image-based data as input versus single-cell data as input. In the context of visualization, we additionally consider the approaches of small multiples and selection of a representative sample, which do not necessarily require

computational methods or statistics. Single cell-based analysis methods all accept the spatial coordinates of cells as input, and either cell type annotations or the full gene expression matrix (or both). These methods may derive results in the form of spatially-variable genes[267], putative interacting genes (*e.g.*, ligand-receptor interactions)[145], meta-genes[120], isodepth[121], spatial cell type co-occurrence[335], or so-called "spatial domains"[454,81].

Spatial domain identification methods can be viewed as spatially-informed unsupervised clustering approaches. The resulting domain memberships of cells are expected to be spatially coherent, if not contiguous. In contrast, spatially variable gene identification methods return rankings and statistical significance values for genes which exhibit spatial patterns of expression. Meta-genes are vectors of gene weights which correspond to spatial patterns, and can be considered as a form of spatial factor analysis. Methods may or may not internally construct a spatial graph in which cells are nodes and edges indicate nearby cells, inverse-weighted by proximity[93,78]. In such graphs, nodes may be labeled according to cell type annotations or may be attributed according to gene expression vectors. In some cases, methods draw upon geospatial analysis functions, such as autocorrelation and Ripley's statistics[335,150].

5.3.2 Tools for Bioimaging Data Visualization

Visualization of bioimages, such as microscopy images, is not a new problem. There are many widely-used desktop-based tools for bioimaging visualization, such as ImageJ[383], CellProfiler[252], and Napari[392]. Tools such as OMERO[54,312] and the Digital Slide Archive[180] help users to organize large collections of images with metadata. There are also tools designed specifically for visualization of pathology images, such as QuPath[66] and HistomicsUI[280]. The comparative pathology workbench focuses on

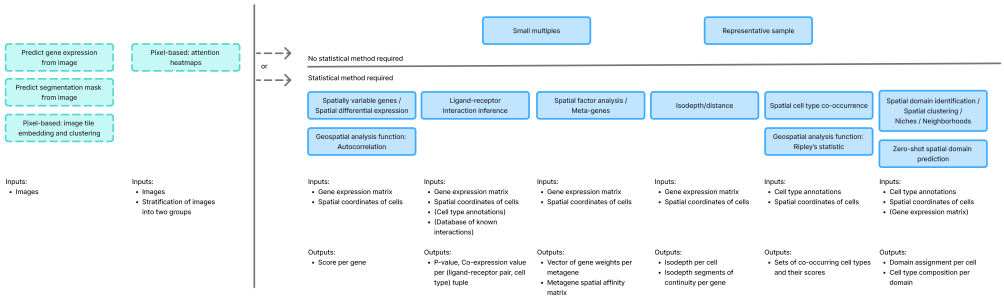

Figure 5.1: Landscape of spatially-resolved data analysis methods. (Left) Image- and pixel-based methods. These may be followed by one or more single-cell methods (or usage of small multiples or a representative sample). For example, spatial differential expression analysis may be performed following the prediction of gene expression values from a histology image. (Top right) Image-based or single cell-based data can be visualized using a small multiples or representative sample approach, neither of which necessarily involves a statistical method. (Bottom right) Single-cell analysis methods which involve statistics.

visualization of small multiples to show many images simultaneously[444].

Web-based tools include OMERO[54,312] and HistomicsUI[280], which offer many of the same features as desktop-based tools, including support for both red-green-blue (RGB) and multiplexed images, and integrated bioimage analysis algorithms. Web-based tools can be more accessible, avoiding challenges of installation and enabling sharing based on URLs. However, web-based tools for multiplexed image visualization have historically relied on server-side rendering approaches, either requiring a specialized server to dynamically perform color mapping and data transformation upon requests ("online") or requiring pre-computed channel groupings and data transformations ("offline"). The former option introduces latency and server-side complexity, while the latter option requires disk space and limits flexibility to the pre-computed options.

In addition to considering the visualization features of such tools, it is also impor-

tant to consider which file formats are supported. In bioinformatics generally, and bioimage analysis specifically, conversion between file formats is a major problem. Conversion can result in loss of information (*e.g.*, 64-bit to 32-bit numeric types), confusion of metadata (*e.g.*, different formats may support different fields), and consumption of time and storage space. Further, many bioimage file formats are proprietary and specific to a given instrument or microscope vendor, complicating conversion. The Open Microscopy Environment (OME) developed the OME-TIFF image format and the Bioformats suite of conversion tools to address these challenges[76]. More recently, the OME-Zarr format has been introduced[311], based on the Zarr array storage format (Section D). OME-Zarr serves as a more cloud-friendly image format (compared to OME-TIFF), which is beneficial as images are becoming larger and analysis workloads are increasingly being run in cloud environments.

5.3.3 Tools for Spatial Single-Cell Data Visualization

Many existing tools support interactive visualization for pathology, for microscopy, and for 'omics data - but few tools support visualization of all three data types together. Several visualization tools are designed for both pathology and multiplexed imaging data. For desktop-based tools, these include CellProfiler[252], Napari[392], ImageJ[383], and QuPath[66], which help users to perform image analysis using a combination of programmatic APIs and graphical user interfaces[252,383,66]. On the web, Minerva[357] provides guided exploration of multiplexed images alongside narrative content and author annotations. A number of image and cell-based analysis pipelines have been designed for close integration with particular visualization tools, including MCMICRO[380] for Minerva and SquidPy[335] for Napari.

On the other hand, there are many related tools designed for visualization of spatially-

resolved ‘omics data. This includes TissUUmaps[351], a scalable web-based tool for interactive visualization of single-cell and single-molecule ‘omics data. Notably, in many such tools, cells and transcripts are the primary supported entities, with all current tools lacking support for visualization of other types of segmented observations (*i.e.*, larger functional tissue units), and by extension they lack support for associating such segmentations with multiple observation-by-feature tables containing associated quantitative or qualitative measurements or derived features. There is a need for tools which support visualization of both computational pathology and spatial ‘omics data types.

5.4 Viv: Multiscale Visualization of High-Resolution Multiplexed Bioimaging Data on the Web

As discussed in Section 5.3.2, the two prevailing paradigms for visualization of multiplexed bioimaging data on the web have major drawbacks. Web browsers provide built-in capabilities for visualization of RGB images (*e.g.*, standard camera images), which bioimaging applications have traditionally used for visualization through server-side conversion. In the “offline” case, the application relies on precomputed conversion of the channels within the multiplexed image to individual RGB image tiles which can be loaded and visualized as needed. In the “online” case, the application requests image tiles from a specialized server which performs pseudocolor mapping on-demand.

We devised an alternative approach to web-based bioimaging visualization which requires neither a specialized server nor precomputed RGB image tiles. We developed Viv, a JavaScript library which contains modules for loading and rendering tiles from

multiplexed bioimaging file formats directly in the web browser (*i.e.*, client-side)[288].

Loading bioimaging data from its primary OME-TIFF and OME-Zarr-based formats requires loading its metadata, including not only experimental metadata but also organizational metadata such as dimensions, resolutions, byte offsets, and compression information. For large images, our approach relies on the multi-resolution (pyramidal) support built into the OME-TIFF and OME-Zarr formats, in which lower-resolution images are stored, enabling direct access to image tiles at the resolution that is most appropriate for a given viewport. While this is a form of precomputation, it does not require precomputed color mappings, and the downsampled images remain useful for non-visualization use cases. We take advantage of existing tools for loading TIFF- and Zarr-based data in JavaScript including GeoTIFF.js[22], Zarr.js[40], and Zarrita.js[47], and have contributed bug fixes and features back upstream into these open-source tools.

To render the data once loaded and decompressed, we use WebGL. WebGL is a web standard which provides APIs for 2D and 3D graphics rendering using the graphics processing unit (GPU). Specifically, we wrote WebGL shaders which integrate with the DeckGL library[433] to perform GPU-accelerated pseudocolor mapping, as well as providing immediate updates to changes in opacity, channel visibility, and affine coordinate transformations. DeckGL enables easily rendering Viv image layers alongside non-image layers, for example, to overlay polygons, text, lines, or points.

In addition to our integration of Viv into Vitessce (Chapter 3) for the rendering of bioimages and segmentation mask images, our distribution of Viv as a standalone open-source JavaScript library[38] has enabled its integration into over 120 open-source tools (based on the GitHub dependency graph as of March 29, 2025) and several commercial tools.

5.4.1 OME-Zarr: a cloud-optimized bioimaging file format with international community support

The introduction of the OME-Zarr format in 2021 presented a challenge of adoption. Given the hurdles of data conversion and tool compatibility across bioimaging formats, it was unclear whether yet-another-format would be adopted by developers and data analysts in the bioimaging ecosystem. As Viv, and by extension Vitessce, represent visualization tools in the ecosystem with support for OME-Zarr, we were invited to contribute to a 2023 survey of early-adopting tools, titled "OME-Zarr: a cloud-optimized bioimaging file format with international community support,"[313]. This survey covers not only nine visualization tools supporting OME-Zarr, but also nine data analysis libraries and four data conversion tools, across several programming languages including Python, C++, and Java. The survey suggests that the rapid uptake in adoption of the format can be attributed in part to the early engagement of the bioimaging community and to the amount & diversity of example datasets made available. Finally, the survey is intended to showcase that, given the number and types of supporting tools, the format is ready for production usage in the hope that this might accelerate further adoption of the format in other tools, forming a feedback loop.

5.5 Interactive visualization of kidney micro-compartmental segmentations and associated pathomics on whole slide images

Following our work to visualize both single-cell (Chapter 3) and bioimaging datasets (Section 5.4) on the web, we aimed to incorporate pathological annotations into such visualizations to assist in data interpretation, in collaboration with pathologists and

computer vision experts in the Kidney Precision Medicine Project (KPMP). These collaborators apply machine learning techniques to perform segmentation of functional tissue units (FTUs) in whole-slide histology images. Together, we built a pipeline to apply previously developed segmentation models of kidney structures and extract quantitative features from these structures. Such quantitative analysis also requires qualitative inspection of results for quality control, exploration, and communication. Therefore, we extend Vitessce (Chapter 3) to enable visualization of segmentations of multiple types of functional tissue units such as glomeruli and tubules in the kidney. Moreover, we propose a standard representation for files containing multiple segmentation bitmasks which we define polymorphically such that existing formats including OME-TIFF, OME-NGFF, AnnData, MuData, and SpatialData can be used. We demonstrate that these methods enable researchers and the broader public to interactively explore renal morphometry of biopsies from the Kidney Precision Medicine Project (KPMP) and the Human Biomolecular Atlas Program (HuBMAP).

5.5.1 Background

Application of machine learning and artificial intelligence to bioimaging is transforming pathology and making computational pathology a reality. Advancements in computational image analysis now make it possible to perform panoptic segmentation in which instance-type and semantic-type segmentations can be computed simultaneously. The current best-performing methods can achieve accuracy rivaling a human pathologist. Confidence in segmentation results has opened the door to derivation of instance-level features, measurements termed pathomics. The datasets that result from application of these segmentation and pathomic feature extraction methods are complex and consist of histology images, multiple types of segmentation bitmasks

(*i.e.*, label images), pathomic observation-by-feature matrices, and categorical or hierarchical labels for observations. Further complexity of datasets may come in the form of network/graph-based representations or machine learning-based predictions of other 'omics measurements such as transcriptomics or proteomics.

Consortia such as the Kidney Precision Medicine Project (KPMP)[132] are applying computational pathology methods to patient biopsies on a large scale to study how diseases such as chronic kidney disease (CKD) and acute kidney injury (AKI) affect renal pathology. Researchers aim to combine computational pathology with statistical methods and complementary multimodal omics experiments to improve diagnosis, prognosis, and treatment for kidney patients. Similar efforts are underway for other diseases such as in the Human Tumor Atlas Network (HTAN)[372] and for normal reference tissue in the Human Biomolecular Atlas Program (HuBMAP)[391].

KPMP is making hundreds of computational pathology datasets available to researchers, patients, and the broader public. This wide audience has the common goal of developing a deeper understanding of kidney disease and reference kidney function using the rich and high-quality data generated by the consortium. However, the individuals in this audience have differing levels of computational expertise. Many may lack familiarity with the specialized software tools and file formats commonly used in bioinformatics and computational pathology. This presents obstacles for non-computational researchers, patients, and others who aim to make sense of the data.

Even for researchers with computational skills, the complexity, size, and number of computational pathology datasets can pose obstacles to their analysis. In many cases it may not be feasible or practical for researchers to download data locally due to storage constraints and/or privacy concerns. Researchers may need to select a subset of biosamples of interest prior to downloading data for further analysis. To min-

imize time spent on data cleaning and data conversion, computational pathology results should be shared using open standard file formats that are efficient (in terms of storage and access) and compatible with a wide range of software tools. Infrastructure costs and maintenance time should also be minimized so that researchers can focus on extracting biologically- and clinically- relevant insights.

The rise in complexity and size of datasets in computational pathology has not been matched by establishment of standard data representations or the development of interactive visualization tools. In contrast, standards and tools which support either bioimaging data or single-cell omics data alone are actively being developed by researchers in biology and computer science[311,291]. Visualization tools for bioimaging data often lack support for categorical and quantitative measurements associated with segmented entities. On the other hand, visualization tools for spatially-resolved 'omics data often support cellular entities and lack support for both **multiple entity types** (*e.g.*, cells, tubules, glomeruli) and **multiple data modalities** (*e.g.*, transcriptomics and pathomics). Many existing tools in both domains (*i.e.*, bioimaging and single-cell 'omics) require specialized servers, non-standard data formats, or are implemented in the form of desktop applications that must be installed.

To overcome these obstacles, we developed a web-based interactive visualization tool for computational pathology data that loads data from the cloud. In conjunction, we propose a standard representation for segmentations of multiple types of observations, which can be linked to pathomics or other 'omics measurements at the observation level. We applied this approach to KPMP kidney biopsies processed through a previously described deep learning-based segmentation and pathomic feature extraction pipeline[215,169]. We show that this interactive visualization approach makes computational pathology pipeline outputs for hundreds of samples accessible to a broad

audience of data consumers.

5.5.2 Results

Interactive visualization of segmentations and associated pathomics features. To support visualization of computational pathology data, we have extended Vitessce (Chapter 3).

The Vitessce framework was originally designed for spatially-resolved single-cell 'omics data visualization. Computational pathology datasets share many characteristics with single-cell datasets but differ in the scale and number of types of entities. For example, spatial single-cell experimental methods such as multiplexed immunofluorescence result in datasets containing high-resolution images, segmentation bitmasks, and cell-by-channel expression matrices. These data types are analogous to the whole-slide images, segmentation bitmasks, and observation-by-feature matrices in pathomics. However, many tools designed for single-cell data consider and support only one type of observation - cells - and one type of feature - genes. In contrast, pathologists must consider many types of entities beyond cells, including smaller structures such as organelles and larger structures such as functional tissue units.

Vitessce supports simultaneous visualization of datasets containing multiple types of observations and features. For example, a segmentation bitmask containing glomeruli can be mapped to the "glomerulus" observation type, and can be visualized alongside its corresponding glomeruli-by-pathomic feature matrix. Meanwhile, another set of segmentations representing tubules can be mapped to the "tubule" observation type with its corresponding pathomic feature matrix. Visualization properties can be configured and linked independently for each observation type. For example, tubules can have a different opacity value than glomeruli, or colors of tubules can be mapped to

feature A while those for glomeruli are mapped to feature B.

Implementation of support for multiple types of observations and features in a single-cell visualization tool is non-trivial. All components within the application must account for the specific type of observation and feature under which they are scoped. The number of data files that the application needs to consider increases, and the internal representations of tabular data must account for multiple tables, all with potentially unique shapes due to the presence of different numbers of rows (*i.e.*, observations) and columns (*i.e.*, features).

Other characteristics of the Vitessce framework inform our choice to use it for the computational pathology visualization use case. As web-based software, it does not require specialized server infrastructure or installation of a desktop application. Large images and other data files can be loaded from cloud object storage systems or standard web file servers. Vitessce can be used not only in web applications but as an interactive widget in computational notebooks such as Jupyter and RMarkdown documents. The Vitessce framework is robust and production-ready as it is already being used in data portals for large atlas-building consortia such as HuBMAP and KPMP to visualize data from spatial single-cell experiments.

Establishment of standard representation for segmentations of multiple observation types. Along with the visualization software, we propose a standard way to represent and organize bitmask segmentation data which comprises multiple types of observations (Figure 5.2). Rather than developing a specialized file format, we build upon established formats for tabular and imaging data.

We propose the following approach, which simply takes the form of a convention that leverages existing open standard formats:

- **Images**: Store RGB or multiplexed imaging data using open standards: OME-TIFF[270,76] or OME-NGFF[311,313].

- **Segmentations**: Store segmentation bitmasks in OME-TIFF[270,76] or OME-NGFF[311,313] format in which each channel corresponds to one observation type. Pixel values start from 1 and correspond to instances of observations, with 0-value pixels representing background. Channel names should specify which channels correspond to which observation types (*e.g.*, cell, tubule, glomerulus).

- **Pathomics**: Store pathomic features in open standard tabular formats: CSV, AnnData[448,427], or MuData[94]. Each table should correspond to a single observation type (*i.e.*, bitmask channel). There must be an additional data structure to map between tables and observation types - this is included in the MuData format (which is a container for multiple tables) but may require an additional file in the case of single-table formats such as CSV and AnnData.

The OME-TIFF and OME-NGFF formats are familiar to researchers performing and analyzing microscopy data. CSV is a widely-used tabular format. AnnData & MuData are efficient tabular formats that are commonly used by single-cell biologists but are designed more generally to represent observations that are accompanied by rich metadata such as feature vectors and categorical class labels. As the SpatialData[291] format becomes adopted within the bioimaging and 'omics ecosystems, we anticipate that SpatialData will also become useful for storage of computational pathology data.

The abstract nature of our proposed data representation approach makes it possible to be used with both current and future file formats. While we provide a reference implementation of a visualization tool that supports this representation, we antici-

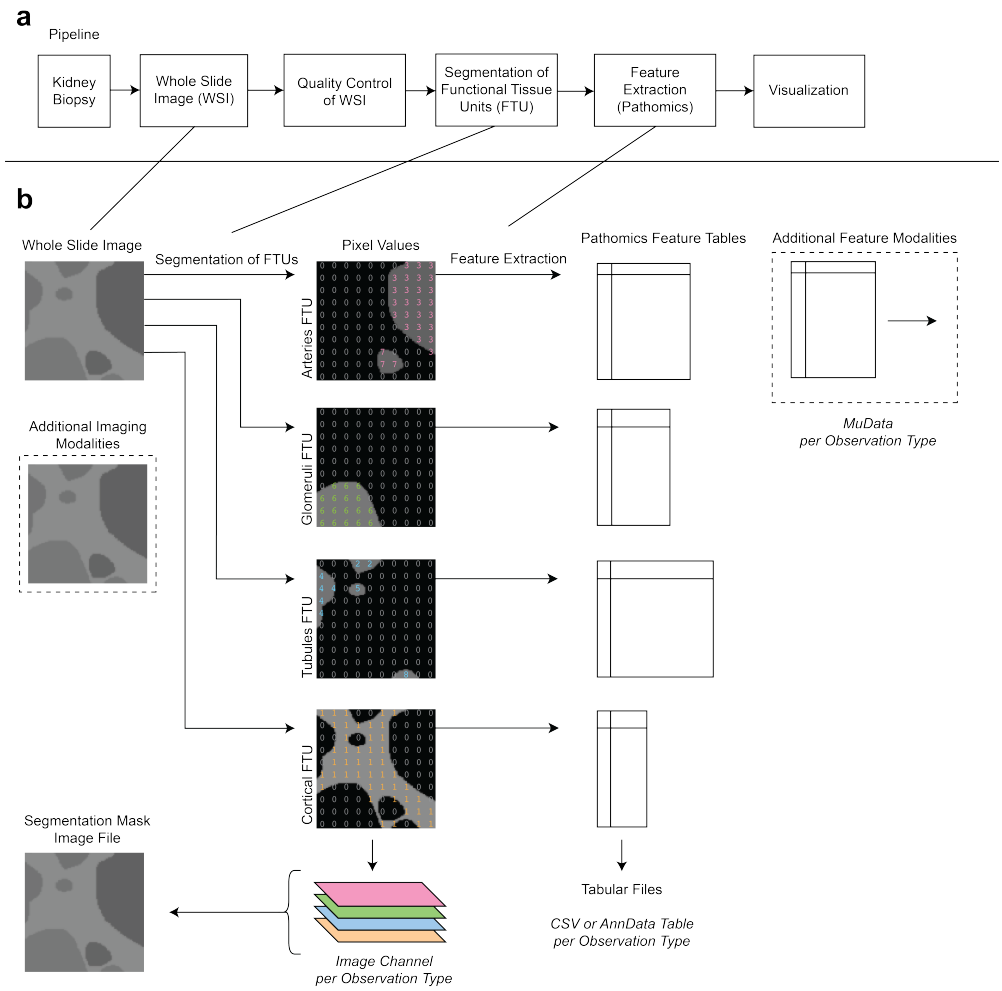

Figure 5.2: a) Overview of pipeline from biopsy and whole slide image (WSI) to segmentation and pathomics to visualization with Vitessce. b) Segmentation masks are represented as multi-channel image files with one channel for each observation type. Pathomics and other features are represented with one table per feature and observation type. Feature tables may contain one row per observation such that different feature tables contain different numbers of rows. Feature table row indices are expected to match the pixel values within the corresponding segmentation mask channel.

pate that other tools and methods at the intersection of computational pathology and spatial 'omics will adopt this format.

Pipeline for segmentation and pathomics feature extraction. We demonstrate our visualization and data representation approaches by using them to visualize the results of a state-of-the-art pipeline for segmentation and feature extraction of functional tissue units in the human kidney. Lucarelli *et al.*[277] describe the development, training, and performance characteristics of this deep-learning based segmentation approach.

This method supports segmentation of multiple observation types, or microcompartments, within the kidney, including cortical and medullary interstitium, non globally sclerotic glomeruli, globally sclerotic glomeruli, tubules, and arteries/arterioles. Additional segmentation networks developed by Jayapandian *et al.*[215] and Ginley *et al.*[169] were used to segment peritubular capillaries (PTC), and pathology-associated regions such as interstitial fibrosis and tubular atrophy (IFTA), respectively.

Once segmented, per-observation pathology features can be extracted using the size and shape of the segmentation itself or using the image pixels from within the segmented region. Features of interest include morphological features, texture features, color features, and distance transform features. Feature extraction is performed using a traditional image analysis approach (*e.g.*, color thresholding) for interpretability.

Application of approach and deployment. We have applied the data storage and interactive visualization approach presented here both to biopsies individually and in an automated fashion across many biopsies from the KPMP and HuBMAP consortia via integration into centralized data processing and dissemination infras-

tructure.

As of January 8, 2025, we have applied the pipeline to 222 (PAS-stained) histology images from 112 KPMP biopsies, all of which are available to explore interactively via the Spatial Viewer within the KPMP Atlas data portal (Figure 5.3d). Of these 112 samples, 11 fit the clinical criteria (based on primary adjudicated category) for Acute Interstitial Nephritis, 16 for Acute Tubular Injury, 39 for Diabetic Kidney Disease, and 24 for Hypertensive Kidney Disease (the categories for the remaining samples were "Other" or "Cannot be determined"). This broad deployment of the approach showcases the ability of this pipeline to extract segmentations from both reference and disease samples at the atlas scale.

In the form of a case study, we next demonstrate how characteristics of an individual biopsy can be identified via the segmentation and extracted-feature visualization in Vitessce (Figure 5.3). We first visualize the whole slide image for a kidney biopsy of interest. Next, we visualize the segmentations for the peritubular capillaries (PTCs) as a layer on top of the image by toggling their visibility using the layer controller. As we zoom in to view the segmentations in more detail, higher-resolution data loads. By default, the PTC segmentations are all mapped to a single color, but we can instead map quantitative features onto the segmentations using a quantitative color map. For example, we can select "area" from the list of features for PTCs, and we can reference the legend in the top right corner to distinguish capillaries with larger area (in yellow) to those with smaller area (in dark purple). As the biopsy at hand contains interstitial fibrosis and tubular atrophy (IFTA), we can view regions of IFTA by toggling the visibility of this segmentation channel. We can adjust the transparency levels of the PTC and IFTA segmentation channels to view the image and all segmentations simultaneously. Upon viewing the IFTA regions and PTCs with color-

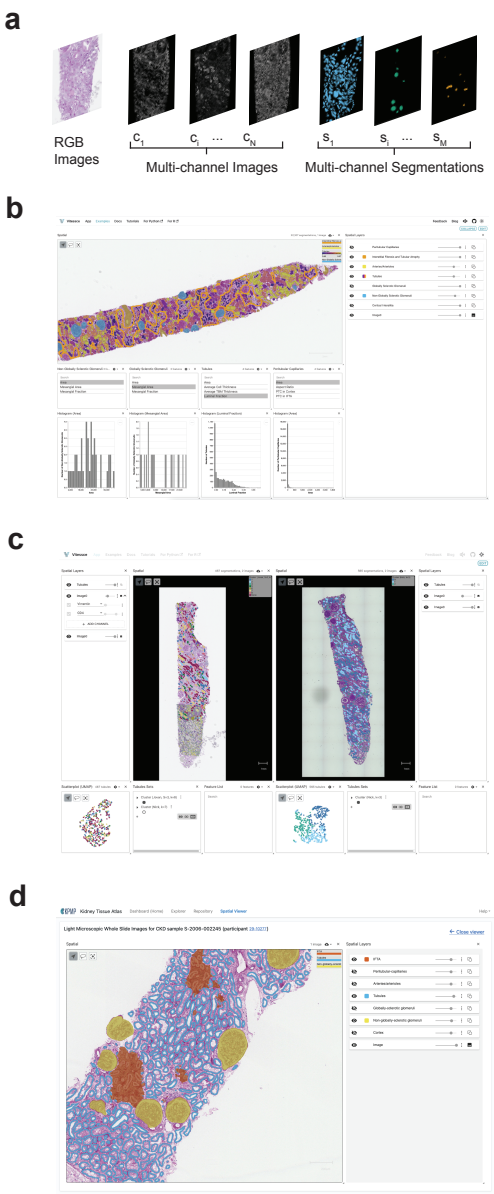

Figure 5.3: a) Schematic of superimposed approach to visualization of image and segmentation layers. b) Screenshot of Vitessce visualization interface during exploration of a single biopsy with segmentations of functional tissue units (FTU) displayed and extracted features visible in both histograms and color mappings. c) Screenshot of Vitessce visualization interface during comparison of two imaging modalities and associated FTU segmentation masks. d) Screenshot of visualization interface embedded in the KPMP Kidney Tissue Atlas web portal, displaying segmentation masks for a sample of interest.

mapped area values, we hypothesize that the distribution of PTC area differs within IFTA regions versus outside. We can use linked interactive statistical plots to quickly compare these distributions, stratifying PTCs based on whether they lie within cortical IFTA, cortical non-IFTA, or total cortex regions.

5.5.3 Discussion

Application of machine learning and artificial intelligence in pathology has brought about revolutionary advances and the emergence of the field of computational pathology[69,394]. Meanwhile, spatial 'omics approaches are advancing in parallel. As it is of increasing interest to perform both histology and spatial 'omics assays on identical or serial tissue slices, it is becoming increasingly important to be able to analyze these data modalities in tandem during the downstream data interpretation stages.

To enable the visualization of such data modalities together, we contribute a new standard for storage of computational pathology segmentation and pathomic feature data (informed by standards from the single-cell domain). We propose the reuse of existing file formats because there is much prior work in development of formats for imaging and 'omics data. OME-TIFF[76] and OME-NGFF[311] are two file formats that support multiplex multiscale images with rich experimental metadata. These formats also support storage of images that represent segmentation bitmasks. In the single-cell analysis ecosystem there are many formats designed around tables of observations and features that are accompanied by metadata, such as AnnData[427], MuData[94], SingleCellExperiment, and Seurat[377]. More recently, formats have emerged that are designed to contain both spatially-resolved data such as images and segmentations as well as observation-by-feature tables and their metadata (but not necessarily more than one type of observation)[291]. At the time of writing, the ecosystem of

data analysis methods is fragmented in terms of supported file formats which motivates our higher-level approach that supports a combination of multiple underlying open-standard file formats. For example, CSV and AnnData can be interchanged for tabular data while OME-TIFF and OME-NGFF can be swapped for imaging and segmentation data, so long as there is a correspondence between data tables and types of segmented entities.

We demonstrate our approach on KPMP biopsies segmented by an exemplary pipeline[215,169]. However, the visualization approach that we propose is generic and could be reused with any segmentation pipeline that outputs results in the open standard formats as described in Results (Section 5.5.2). Further, scripts could be developed to convert alternative data representations to those that we propose, or plugins can be developed for visualization tools to add compatibility for loading data from additional formats. This work demonstrates the importance of considering that tissues are composed of multiple functional tissue units when developing computational tools and data structures, despite the focus to date on cells (and subcellular components) and the established ecosystem of single-cell analysis tools. As deep learning-based segmentation methods advance for additional types of functional tissue units (FTUs) within other organs, visualization of the resulting FTU segmentations will become more valuable for purposes including exploration and communication. Network-based analyses will become increasingly important and therefore overlaid network visualizations are an important future direction for this work. Other follow-up work will be needed to support tasks such as proofreading (*i.e.*, detection and correction of segmentation errors), human-in-the-loop annotation, and support of additional dimensions (*e.g.*, 3D, temporal) and scales.

5.5.4 Methods

Further method details can be found in Appendix I. This appendix describes the implementation of WebGL shaders to render segmentations, perform color mappings, and dynamically computed stroked and filled representations for individual segmented entities.

5.6 Interactive visualization of spatial query results

The identification of cellular neighborhoods within spatial 'omics datasets is of great interest, and represents the aim of at least one of the aforementioned classes of spatial data analysis methods (Section 5.1). Such neighborhoods of cells may represent higher-order tissue structures (FTUs) or suggest druggable cell-cell interactions. In relation to high-level goals of constructing cell atlases, a relevant goal includes the improved understanding of spatial interactions between abnormal cells and other cells to promote or suppress disease processes (Table 1.1).

Existing algorithmic approaches for neighborhood identification are largely non-interactive (*i.e.*, lacking support for a human-in-the-loop). We developed an extension of Vitessce that supports human-in-the-loop execution of a cellular neighborhood identification algorithm called SpatialQuery. This extension enables users to specify method parameters and define spatial queries of interest. Through multiple usage scenarios, we demonstrate that this system provides useful visual analysis features for spatial 'omics data.

5.6.1 SpatialQuery algorithm

We collaborated with the lab of Martin Hemberg to develop an interactive visualization system which incorporates the Hemberg lab's SpatialQuery[35] algorithm for identification of cellular neighborhoods. This algorithm identifies neighborhoods which take the form of spatially co-occurring cell types (Section 5.1). The algorithm takes as input an AnnData object and specification of the columns containing cell type annotations and spatial coordinates. First, SpatialQuery partitions cells using a k-dimensional (KD) tree data structure, which subsequently enables lookups for cells within a given radius r of one or more queried cells (or alternatively, the n nearest neighbor cells of the queried cells). Next, SpatialQuery constructs a frequent-pattern (FP) tree, which enables usage of the FP-growth algorithm[185] to extract frequently-occurring sets of cell types. The FP tree is constructed by extracting the set of cell types in proximity to every other cell (based on radius r or nearest neighbors n), which form the *transactions* in FP tree terminology. The FP-growth algorithm then identifies sets of co-occurring cell types ordered based on their *support* value, defined as the fraction of transactions in which the cell types co-occur, among all transactions. Notably, the KD- and FP-tree data structures enable SpatialQuery to be highly efficient, lending itself to usage in an interactive visualization context.

5.6.2 Visualization of SpatialQuery results

SpatialQuery supports identification of frequently-occurring spatial patterns of cell types, either in a single field of view (FOV) or across multiple FOVs. Here, we focus on the single FOV setting, though analysis across multi-FOVs represents important future work. We visualize the spatial localization of cell types using a categorical color

mapping, and plot cells according to their spatial coordinates.

To demonstrate SpatialQuery, we use a Slide-seq[364] dataset profiling a human kidney sample from the HuBMAP consortium (HBM838.LDFP.578). In the HuBMAP Data Portal, we can view the default Vitessce visualization for this dataset, which contains cell type annotations as well as multiple views: dimensionality reduction scatterplot, spatial plot, and a heatmap (Figure 5.4). Once we import the dataset into a Jupyter notebook, we can use the Vitessce Python API to customize the visualization and integrate the Spatial Query Manager view (Figure 5.5). In this case, we also configure the visualization to include the spatial view and cell set manager view. In the spatial view, cells are colored according to the colors of the currently-selected cell sets. These cell sets represent either cell type annotations computed by HuBMAP data processing pipelines or the results of execution of SpatialQuery (upon querying via the Spatial Query Manager view). As SpatialQuery can return multiple frequent patterns for a given query, we leverage the hierarchical support of the Vitessce cell set manager view to allow the user to select among multiple identified co-occurring sets of cell types (Figure 5.5, Bottom Left).

In this kidney sample, execution of SpatialQuery with the default parameters (Figure 5.5, Bottom Right) results in one identified pattern consisting of two cell types: Cortical Thick Ascending Limb Cell (C-TAL) and Proximal Tubule Epithelial Cell Segment 1 (PT-S1). This finding is consistent with a study by Lake *et al.*[250] which finds that the C-TAL and PT-S1 cell types are both primarily found in the cortex region (as opposed to the outer or inner medulla regions) of the kidney.

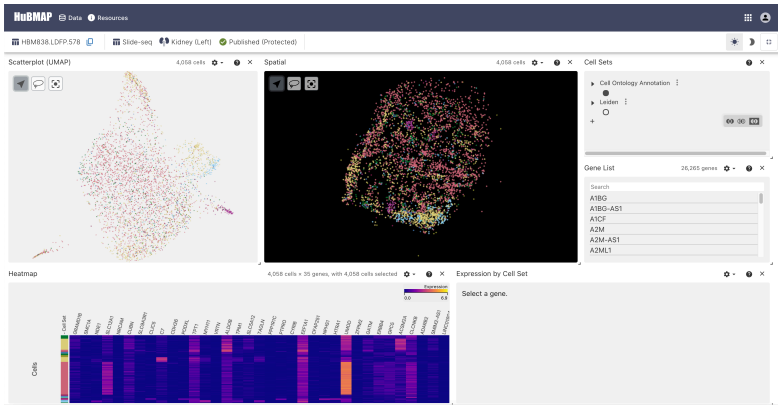

Figure 5.4: Visualization of Slide-seq dataset in the HuBMAP Data Portal.

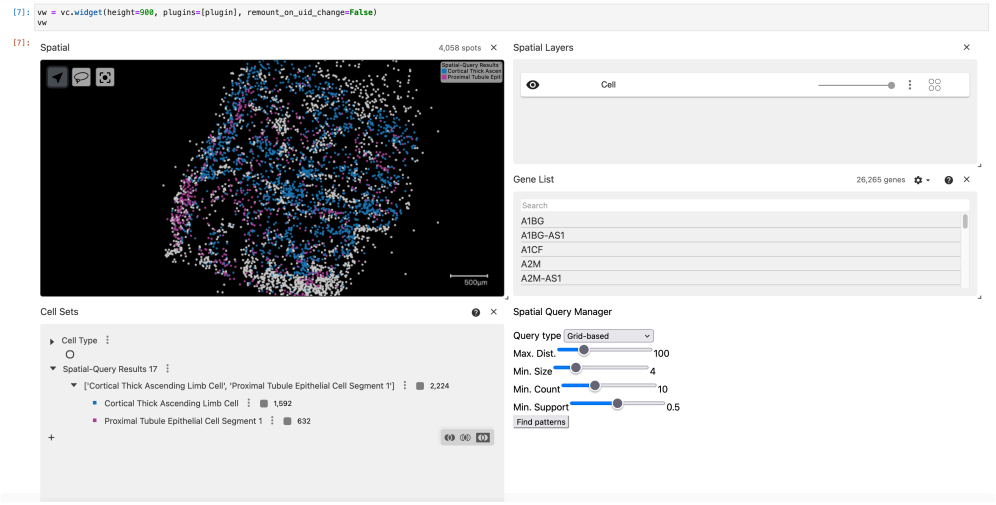

Figure 5.5: SpatialQuery and Vitessce in a Jupyter notebook. Top Left) Spatial view, displaying cells at their spatial coordinates and colored according to the colors of the selected cell sets. Bottom Left) Cell Set Manager view, displaying not only cell type annotations computed by HuBMAP, but also the frequent patterns identified by the SpatialQuery algorithm. Bottom Right) Spatial Query Manager view, containing controls for setting SpatialQuery parameters, selection of the query type, and a button for query execution.

Implementation

Vitessce can display interactive visualizations in multiple contexts, including within Jupyter notebooks in Python environments (Figure 3.1). In a Python environment, Vitessce can visualize data that is simultaneously accessible within Python data analysis code. As the SpatialQuery algorithm is implemented in Python, we developed an approach which uses the Vitessce notebook widget to execute SpatialQuery functions and visualize the results (Figure 5.6).

We developed a custom Vitessce view to control SpatialQuery: the Spatial Query Manager. This view enables users to set query parameters and then execute their queries. The synchronization of values between JavaScript and Python environments is facilitated by Anywidget[289]. For cell type-centric queries, the current selections of the existing Cell Set Manager view are used. The Spatial Query Manager view is implemented using the JavaScript plugin API of Vitessce directly within the Jupyter notebook. Further, as the view implementation consists of a React component, we developed a mechanism by which a Python string containing JSX syntax can be transformed into standard JavaScript code directly within Python, without the need to install any additional tools[21]. We anticipate that this software architecture can be reused to implement plugin views for alternative spatial data analysis methods.

5.6.3 Discussion

This section presents an extension of Vitessce which enables interactive analysis of spatially colocalizing cell types derived via the interpretable SpatialQuery method.

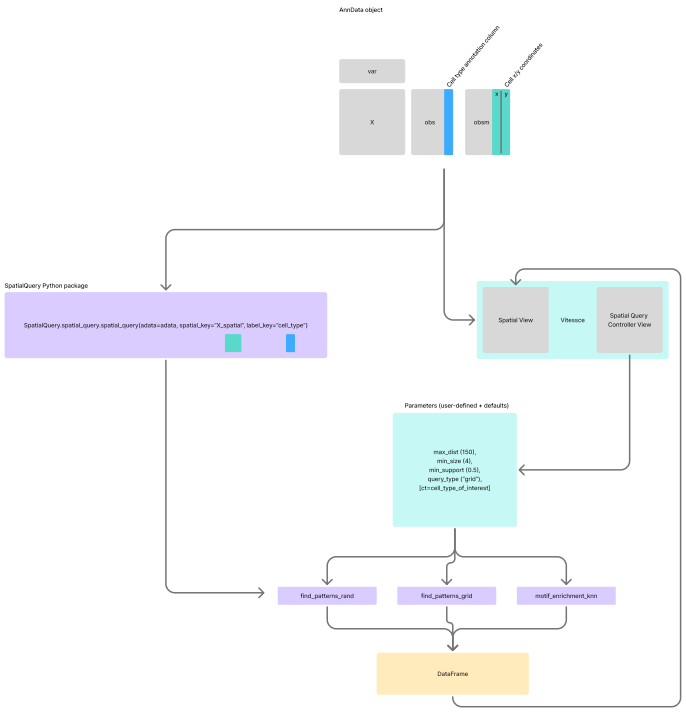

Figure 5.6: Implementation of a Vitessce plugin for the SpatialQuery method.

Alternative visual representations for SpatialQuery results. This work has primarily focused on the integration of the SpatialQuery algorithm into a visual analytics context. More work remains to improve the visual presentation of SpatialQuery results. Whereas we have used color to encode the SpatialQuery results, this could be complemented by alternative visual encodings, including non-spatially-resolved visualizations of the statistical results. For example, pairwise relationships between cell types could draw upon the heatmap visualization designed by Somarakis *et al.*[393]. Additional potential visualization designs include those which would enable comparison of different method parameters side-by-side (as opposed to only iteratively).

Visual analysis across multiple fields of view. In this project, we have used SpatialQuery to analyze data from a single field of view (FOV). However, the SpatialQuery toolkit contains functions for analysis of multiple FOVs, both to find patterns across them (integrative in nature) as well as to perform differential pattern analysis. While the results of such analyses could naively be visualized using small multiples or representative samples (Section H.6.1), alternative visualization approaches should be explored.

5.7 Discussion

This chapter presents multiple projects in which we have built tools for interactive visualization of bioimaging and spatial single-cell data.

5.7.1 Limitations and Future Work

Vitessce integrations with alternative analysis methods. Many opportunities remain to develop interactive visualizations that assist users during spatial data analysis (or facilitate communication and presentation of results from such analysis). Our development of interactive visualizations for bioimaging data, segmentations of functional tissue units (FTUs), and spatially co-occurring sets of cell types (SpatialQuery) are only the tip of the iceberg. For example, there is potential to combine the FTU segmentation work with the capabilities SpatialQuery, enabling differential queries of co-occurring cell types within versus outside FTUs of interest. This could be achieved through modification of the SpatialQuery plugin or through tighter integration of SpatialQuery into Vitessce (*e.g.*, implementation of SpatialQuery in JavaScript on the client-side, so that segmentation data loaded by Vitessce can be considered in its algorithm). There are opportunities to integrate Vitessce with alternative classes of data analysis methods (Section 5.3.1), including methods which compute isodepths, reuse geospatial statistics, or operate on pixel-based data to predict gene expression or produce attention heatmaps. While methods such as SpatialQuery consider only spatial coordinates and cell type labels, analysis methods which consider the full gene expression matrix, including graph-based methods, may provide deeper insight into the biomarkers driving important spatial patterns.

Improving interpretability via image channel groupings based on cell types. Multiplexed imaging techniques such as CODEX and imaging mass spectrometry (IMS) generate images with tens to hundreds of channels. While Vitessce currently enables users to select one or more image channels of interest, and map

these to different colors, this does pose some limitations. Users may not know ahead of time which biomarkers they are interested in. Instead, users may have a cell type of interest in mind. In the future, we intend to leverage ontology term mappings between the cell ontology and feature ontologies (*e.g.*, UniProt or HGNC), to account for the fact that cell type-to-biomarker relationships are not necessarily one-to-one. Users could supply data in the form of ontology-term-edge tuples `(cellTypeTerm, biomarkerTerm)` which would facilitate these mappings. This would require that cell types and image channels have been mapped to ontology terms, for instance, using an Organ Mapping Antibody Panel[356] or a Shapley additive explanation approach[152]. Using these mappings, selections of cell types from the existing Vitessce Cell Set Manager view interface can be synchronized with the selection of image channels containing the corresponding biomarkers.

Addressing scalability. Vitessce currently uses the Viv library for loading and rendering of bioimages (Section 5.4), which, through pyramidal- and tiling-based approaches, scales to large images whose storage requires hundreds of gigabytes. However, experimental techniques such as MERFISH[112] and 10x Genomics Xenium use imaging to localize individual RNA transcripts, resulting in datasets containing tens of millions of RNA transcript points (consisting of the transcript/gene identity and its spatial coordinates). Through discussions with biologists who perform MERFISH experiments, we heard that these individuals are interested in visualizing the point data and underlying microscopy images simultaneously, to facilitate quality control of the point identification process, among other tasks. In contrast to pixel-based data, which is organized on a regular grid, point-based data is not straightforward to aggregate into a pyramidal data structure, complicating efficient loading and rendering. The

SpatialData format has emerged to address some of these issues[291], and we propose an alternative approach in which the OME-NGFF image format can be reused for efficient storage of point-based data (Section I.3). Approaches that facilitate effective visualization must also consider data access patterns (Section 1.8.11).

In addition to the data scalability challenge, there is also a challenge of visualization scalability, as the human visual system cannot distinguish individual data points when millions are rendered simultaneously (*i.e.*, overplotting); instead, we interpret higher-order patterns that emerge. One way to address this visualization scalability challenge is to use a semantic zooming approach to render different visual representations based on the user's current zoom level. The semantic zooming technique is also used by Celldega[16] for spatial single-cell data and by Gosling[281] for genomics data, and serves as a promising approach that could be integrated into Vitessce.

Visualization of temporally-resolved and volumetric data. In introductory sections 1.5.7 and 1.8.3, we motivated the consideration of the temporal data dimension and the third (Z-axis) spatial dimension. While we have begun to address the latter by expanding the volumetric rendering capabilities of Vitessce and integrating these into mixed reality environments[314], more work remains to effectively visualize temporally-resolved and volumetric single-cell data. As more datasets of these types emerge, visualization tools which support these data types will become increasingly important.

5.8 Conclusion

This chapter highlights advancements that we have made in developing interactive visualization tools tailored for bioimaging and spatial single-cell data, multi-channel

segmentations of functional tissue units and other entities, and dynamic queries of spatial cell type co-occurrence. The integration of Vitessce with algorithms such as SpatialQuery exemplifies the potential for interactive, human-in-the-loop techniques, enabling users to quickly iterate during data exploration. Several opportunities remain for future research, including building query capabilities for a broader set of analysis methods, improving scalability to support larger datasets, and developing visualization techniques for temporal and volumetric data. Addressing these challenges will enable richer visual analysis of relationships within spatially-resolved datasets and can further bridge the application of statistical methods to the generation of biologically-relevant insights.

6

Visual analysis of single-cell chromatin accessibility

6.1 Preface

Portions of this chapter have been adapted from previous publications which I have co-authored:

- The chromatin landscape of healthy and injured cell types in the human kidney[170]
- Cistrome Explorer: an interactive visual analysis tool for large-scale epigenomic data[282]

The chromatin landscape study was led by Debora L. Gisch, with contributions from me and many others. The integration of genomic visualizations in Vitessce and the HuBMAP data portal was an effort with contributions from not only me but the entire HuBMAP Data Portal team, in particular, Nikolay Akhmetov, Tabassum Kakar, Lisa Choy, John Conroy, Tiffany Liaw, and Morgan Turner. The Cistrome Explorer project was led by Sehi L'Yi, with contributions from me, Ariaki Dandawate, Len Taing, Chen-Hao Chen, Myles Brown, Clifford A. Meyer, and Nils Gehlenborg.

6.2 Introduction

In eukaryotic cells, processes of gene regulation are controlled by a complex interplay between DNA-binding proteins (transcription factors) and chromatin organization, which together define the epigenome[97]. Chromatin is organized hierarchically, from the entire chromosome at the most coarse level, to nucleosomes at the finest level. Nucleosomes are 147-base pair regions of DNA that are compactly wrapped around a set of eight histone proteins called a histone octamer[340]. Importantly, the epigenome is dynamic, enabling the cell to adapt its transcriptional processes based on stimuli or developmental stages. Thus, epigenomic studies which map chromatin states and transcription factor (TF) binding sites genome-wide generate vast amounts of data that are challenging to analyze. Yet genome-wide epigenomic maps are powerful for

identification of promoters, enhancers, and repression and activation of gene expression[473].

Different sequencing-based experimental techniques have been developed to map different aspects of the epigenome. These techniques include whole genome bisulfite sequencing (WGBS), chromatin immunoprecipitation followed by sequencing (ChIP-seq), DNase I hypersensitivity followed by sequencing (DNase-seq), cleavage under targets & release using nuclease (CUT&RUN), and assay for transposase-accessible chromatin using sequencing (ATAC-seq)[170,304]. Bisulfite sequencing identifies methylated DNA, which is believed to lead to repression of gene expression long-term (in contrast to methylation of histone proteins which is more readily reversible)[105]. ChIP-seq uses antibodies targeted to proteins of interest to enrich for DNA fragments that are bound to these proteins, which can be mapped genome-wide through sequencing of the fragments following enrichment[340]. When targeted to a transcription factor of interest, ChIP-seq can identify binding sites and sequence motifs[340]. When targeted to histone proteins in particular post-translational modification states, ChIP-seq can identify positions of the modified nucleosomes[473]. DNase-seq identifies regions of DNA that are easily digested by DNase I, called DNase I hypersensitive sites[90], which correspond to positions where nucleosomes have been displaced and therefore indicate the positions of regulatory elements including promoters, enhancers, and transcription factor binding sites.

To date, there has been a focus on developing analytical and visualization methods for single-cell transcriptomics data, despite a wealth of data and regulatory information to be gleaned from chromatin accessibility (scATAC-seq) and complementary epigenomics modalities. This lack of visualization focus stems from the more complicated nature of the data: the genomic axis, larger scale (genome-wide vs. gene-

level/transcriptome-wide), and modeling difficulties of the data. In this chapter, I describe interactive visualization tools that we have developed for exploration of scATAC-seq and other epigenomics data, including in cell atlas contexts.

6.3 Related Work

The closest related work to ours consists of genome browser visualization tools, which display one or more genomic sequences. We refer to Nusrat *et al.*[325] for an overview of genomic visualization techniques. Other related work includes graphical interfaces for analysis and querying of epigenomics databases.

6.3.1 Genome Browsers

Genome browser tools typically display genome-mapped data using a linear coordinate system. The UCSC Genome Browser is a foundational tool in genomics which was developed more than two decades ago and remains widely used today[230]. Other widely-used genome browser tools include the Integrative Genomics Viewer[362] and JBrowse[388]. These tools are flexible and offer many features, but were not designed for visual analysis of chromatin accessibility data from many cells or samples.

6.3.2 Interactive Tools for Epigenomics Data Analysis

Several tools have been designed for interactive visualization of epigenomics data. The WashU Epigenome Browser[474] and EpiViz[110] offer epigenomics-specific features, such as visualization of long-range interactions or chromatin domains from Hi-C data, yet remain limited in the number of samples that can be visualized at one time. HiGlass was initially designed for scalable visualization of genomic contact matrices,

such as from Hi-C experiments, but also contains many classic genome browser features such as support for one-dimensional sequence tracks[231]. These tools lack support for visualization of many (tens to hundreds) of epigenomic tracks simultaneously, alongside sample-level metadata such as cell type annotations or provenance information.

6.4 Kidney Chromatin Landscape Browser

Our collaborators in the Kidney Precision Medicine Project (KPMP) used several epigenomics profiling techniques to identify regulatory regions with key roles in kidney injury[170]. This effort was motivated by the previous identification of altered gene expression patterns in the injured kidney[250], thus epigenomics profiling could help to define the upstream mechanisms of these altered states. This profiling resulted in the construction of an atlas of methylation, histone modifications, and open chromatin.

6.4.1 Experimental Methods

This kidney atlas includes a total of 25 samples, from both biopsies and nephrectomies comprising both healthy and injured states, with different subsets of samples profiled using different experimental techniques. The epigenetic techniques used include: (1) WGBS from laser microdissected glomeruli (GLOM) and tubulointerstitium (TI), (2) single nucleus multiome (10x Genomics combined snATAC-seq and snRNAseq), (3) bulk ATAC-seq, and (4) CUT&RUN for histone modifications of active chromatin (H3K27ac, H3K4me3, H3K4me1) and repressive chromatin (H3K27me3). Gisch *et al.* further applied spatial transcriptomics techniques for orthogonal and spatial validation of findings[170].

6.4.2 Interactive Browser

We developed a kidney chromatin landscape browser[27] to view the aligned datasets across the genome (Figure 6.1). The browser application is based on HiGlass[231] and allows users to explore the study results, with features for interactively viewing cell type-specific profiles. The application initially loads six genomic tracks: the two WGBS profiles (for GLOM and TI) and the four CUT&RUN profiles (H3K27ac, H3K4me3, H3K4me1, H3K27me3). Users can zoom and pan along the genome or can navigate to a genomic interval of interest. With the cell type selection controls at the top of the application, users can dynamically append aggregate scATAC-seq profiles for cell types or subtypes of interest.

In contrast to a typical HiGlass-based application, this browser does not load genomic profile data from a specialized server. Instead, we used the HiGlass plugin API to develop a plugin data fetcher for BigWig files[24]. This reduces the cost and complexity of the application, as the BigWig files can be stored in an object storage system or other static file server.

6.5 Extension of Vitessce for visualization of chromatin accessibility data

Today there exist several types of multimodal experimental techniques which enable profiling both the transcriptome and the epigenome in the same cells[420]. These include SNARE-seq[114] and the commercially-available 10x Genomics Multiome, both of which enable profiling gene expression and chromatin accessibility, transcriptome-wide and genome-wide, respectively. While the aforementioned Kidney Chromatin Landscape Browser (KCLB) contains visualizations of chromatin accessibility modality

Figure 6.1: Kidney Chromatin Landscape Browser interface. (Top) Controls for selection of cell types or subtypes of interest. (Bottom) Interactive genome browser tracks rendered using HiGlass[231]. As the user zooms in, higher-resolution data loads.

data produced by 10x Genomics Multiome experiments (Section 6.4.1), that browser does not link this data to the corresponding transcriptomics profiles. Therefore, we integrated the same genome browser track visualization approach into Vitessce (Chapter 3) for integrated visualization of data from both experimental modalities. As in the KCLB project, we use HiGlass[231] and custom HiGlass data fetcher plugins[25] to implement this functionality.

6.5.1 HuBMAP Data Portal Integration

In the Human BioMolecular Atlas Program (HuBMAP), multiple tissue mapping centers are generating data using 10x Genomics Multiome experiments[391]. Similar to Gisch *et al.*[170], these data generators intend for this multimodal data to provide insight into the regulatory mechanisms controlling important gene expression programs. Further, as HuBMAP is profiling non-diseased tissue samples, the consortium intends

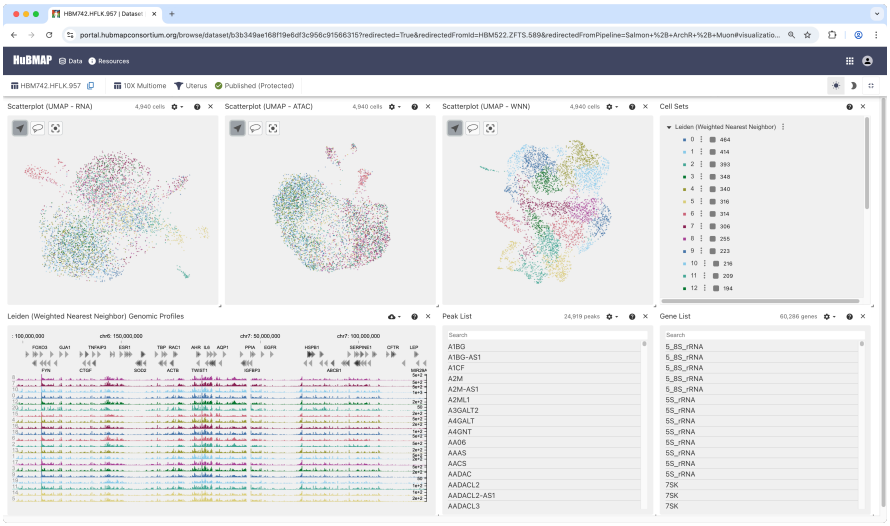

Figure 6.2: Visualization of a 10x Genomics Multiome experiment in the HuBMAP Data Portal. (Top Left) Three dimensionality reduction scatterplots, considering transcriptomics-only, chromatin accessibility-only, and a weighted nearest neighbors-based integration of both feature types. (Bottom Left) Interactive genome browser tracks rendered using HiGlass[231]. (Right) Controls for selecting cell clusters, chromatin accessibility peaks (mapped to genes), and genes of interest.

for this data to serve as a reference with which future researchers can compare data from disease contexts.

In collaboration with the HuBMAP Data Portal and the HuBMAP Integration, Visualization, and Engagement (HIVE) teams, we integrated interactive multimodal visualizations of 10x Genomics Multiome experiments into the HuBMAP Data Portal (Figure 6.2). In these visualizations, the genome browser track view is linked to multiple dimensionality reduction scatterplot views and the cell set manager view. Selections of cell clusters and cell types are coordinated across all views. This implementation has involved not only extension and configuration of Vitessce (*i.e.*, client-side), but also the development of new data processing pipelines[29] that are executed within the HuBMAP infrastructure for all 10x Multiome datasets.

6.6 Cistrome Explorer: an interactive visual analysis tool for large-scale epigenomic data

The Cistrome Data Browser (Cistrome DB) is a web-based resource containing over 47,000 mouse and human ChIP-seq, DNase-seq, and ATAC-seq datasets that have been uniformly processed to identify cis-regulatory binding sites for trans-acting factors along the genome[271,304,473]. Cistrome DB integrates with the UCSC Genome Browser and WashU Epigenome Browser for visualization of one or more user-selected samples from the database, yet without support for integrated metadata visualizations nor scalable to hundreds of selected samples. However, Cistrome DB does provide functionality for searching across the entire database. Users can search for samples based on targeted factor, biological source (cell line, cell type, or tissue type), species, or quality control metrics. In addition, the Cistrome Toolkit offers querying capabilities that allow users to ask which factors regulate a gene of interest, which factors bind in a given genomic interval, and which factors have a significant binding overlap with a given peak set (*i.e.*, set of genomic intervals). Responses to these queries may include not only the identified factors but also the relevant samples in the database driving the results.

In collaboration with the developers of Cistrome DB, we sought to integrate visualizations that would scale to hundreds of samples from the database and offer users the ability to execute Cistrome Toolkit queries through interaction with the visualizations directly. We developed a tool called Cistrome Explorer[17] that supports these requirements[282] (Figure 6.3). Cistrome Explorer is implemented using HiGlass[231] for scalable loading and rendering of genomic data, with a substantial amount of additional interactive visualization components developed around the main HiGlass view

to display metadata for the hundreds of samples. Iterative workflows are enabled by integration with the Cistrome Toolkit, for example, allowing users to query which transcription factors bind in a selected genomic interval.

6.6.1 Case study

To illustrate the functionality of Cistrome Explorer, we demonstrate a use case with ATAC-seq data from Cistrome DB (Figure 6.3). MYB is a transcription factor that is important in hematopoietic development[323] and has been found to play a role in estrogen receptor-positive breast cancer[143]. After navigating to the genomic coordinates of MYB, Cistrome Explorer reveals that although MYB is a transcription factor important in development, its promoter is accessible in most cell types. In the immune cell types and in breast cancer, the pattern of the enhancer near the MYB transcription start site (TSS) is complex. Using the Cistrome Toolkit enables us to find potential transcription factors by selecting a genomic interval and executing a query directly from within Cistrome Explorer. One enhancer 4 kb from the TSS, accessible in monocytes, macrophages and dendritic cells, is bound by the hematopoietic regulators SPI1 and IKZF1, while another 17 kb from the TSS is bound by RELA in T cells and B-cell derived lymphoblastoid cell lines. The breast cancer-associated enhancer 6 kb from the TSS is bound by ESR1, consistent with the observation that MYB is regulated by the Estrogen Receptor[143].

6.6.2 Implementation

Cistrome Explorer is implemented using a client-server architecture[17] in which the client is a React web application and the server is a modified HiGlass server[231]. The

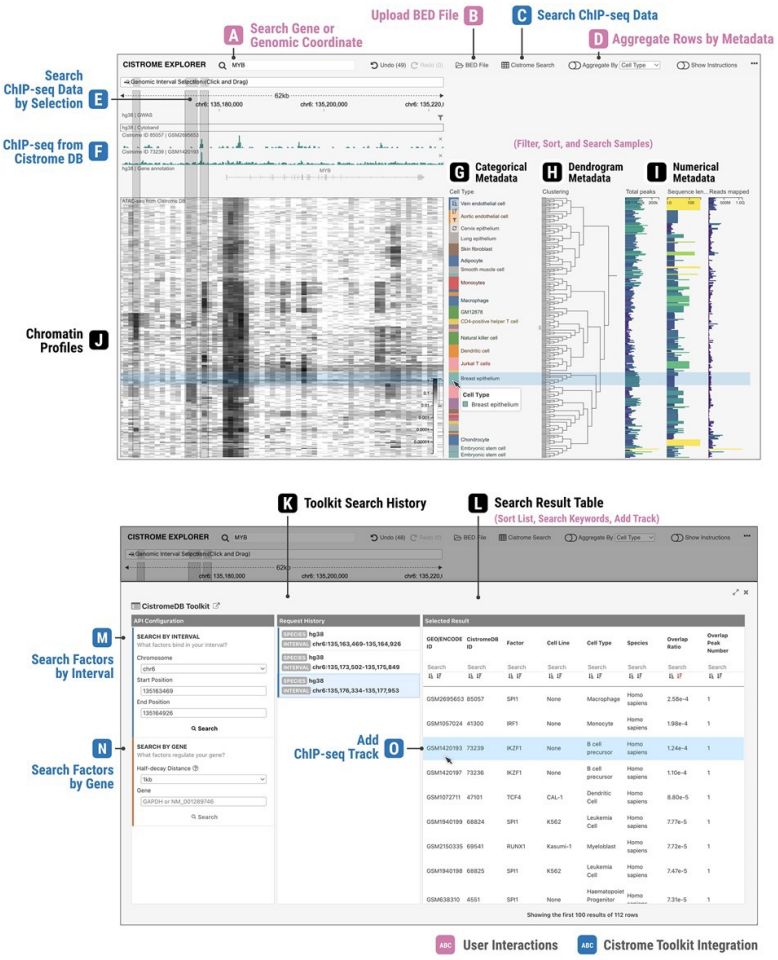

Figure 6.3: Cistrome Explorer interface. (Top) ATAC-seq profiles representing diverse cell types, 228 samples from the Cistrome DB. (Bottom) Cistrome Toolkit analysis, showing a list of transcription factors associated with regions that were interactively selected from the heatmap.

HiGlass server modifications include the ability to dynamically aggregate genomic profiles from user-defined subsets of samples, to retrieve data stored within an S3 object storage system bucket, and to easily deploy the server using a Docker-Compose workflow[18]. The implementation of Cistrome Explorer also required the execution of a Snakemake[308] data processing pipeline to consolidate terabytes of Cistrome DB BigWig files into a HiGlass "multivec"-format file per species and factor[19]. Finally, the client-side interactive visualizations of metadata are implemented using D3[87] and a modified version of the Two[36] JavaScript library, which enables rendering the same plots using either HTML canvas or SVG elements depending on scalability requirements. The states of the HiGlass and D3/Two visualizations are synchronized using the state management features of React.

6.7 Discussion

The three projects described in this chapter enable visual exploration of epigenomics and chromatin accessibility data from single-cell experiments. However, much future work remains, especially related to whether and how to explicitly encode the epigenomic profiles of individual cells visually. While the first two projects (Kidney Chromatin Browser and Extension of Vitessce) display genome-wide single-cell data aggregated to the cell type-level and the third project (Cistrome Explorer) approaches the problem of simultaneous visualization of hundreds of samples with their metadata, none of these three projects explicitly visualizes the (non-aggregate) genomic profiles of individual cells. This problem remains challenging due to both data sparsity and rendering scalability, yet these preliminary projects provide lessons that can inform future directions.

Cistrome Explorer is the most complex of the three aforementioned projects, offering query functionality to answer key user questions that arise during visual analysis, such as which transcription factors bind in a user-selected genomic interval. Cistrome Explorer also offers the most advanced metadata visualizations, with support for legends for sample provenance, bar plots for quality control metrics, and dendrograms for hierarchical clustering results. Additionally, Cistrome Explorer supports on-the-fly aggregation of genomic profiles based on metadata such as cell type and tissue type. Through future work, these features could be integrated into the Kidney Chromatin Landscape Browser, Vitessce, and the HuBMAP Data Portal. The coordinated multiple views in Vitessce would allow for straightforward implementation of dynamic aggregation based on user-defined groups of cells (*e.g.*, the lasso tool in the scatterplot could determine which cells' genomic profiles to aggregate). Finally, while Cistrome DB would not be considered to be a single-cell atlas resource, its focus on processing, cataloging, and querying publicly-available epigenomics data does present similar challenges, especially in terms of the scale and heterogeneity of the data.

While these projects involved close collaboration with experts in analysis of epigenomics data, they did not involve formal user studies to evaluate the described interactive tools. Further, since our survey in Chapter 2 did not identify a large number of epigenomic atlas publications, those findings lack depth into patterns of single-cell epigenomics visualizations in the atlas literature. Thus, there are opportunities to complete formative interview studies with data analysts and to perform a focused survey of figures in papers describing single-cell chromatin accessibility (scATAC-seq) experiments.

As the scale (in terms of cell count) of scATAC-seq experiments continues to increase, performance challenges for interactive visualization tools will grow more pro-

nounced. Specifically, tools that aim to perform dynamic aggregation of thousands of per-cell genomic profiles may need to use strategies such as downsampling or streaming. Regarding the implementation, recent web technologies such as WebAssembly could play an important role in addressing these challenges. Some tools in the chromatin accessibility data analysis ecosystem are moving towards formats such as AnnData and MuData[94]. Consideration of data storage formats that are optimized for visualization use cases will also be important in this domain, as it has been for addressing challenges in visualization of single-cell spatial and imaging data.

6.8 Conclusion

This chapter has presented three projects that advance the visual analysis of epigenomic and chromatin accessibility data, including for single-cell and multimodal datasets. From the Kidney Chromatin Landscape Browser to Cistrome Explorer, these projects demonstrate how interactive visualization tools can empower researchers to explore regulatory mechanisms using this data. While significant progress has been made, especially in multimodal data integration, metadata visualization, and integrated query functionality, challenges remain, including the need for methods that can more directly and effectively represent per-cell epigenomic variation, as well as additional user evaluations, will be essential for ensuring these tools meet user needs. By evolving in tandem with experimental techniques and integrating with knowledge bases such as Cistrome DB, visual analytics platforms can continue to play a critical role in epigenomic data analysis.

7

Conclusion

7.1 Thesis summary

This thesis has been informed by both the single-cell biology literature and close interactions with biologists, pathologists, clinicians, visualization experts and other researchers interested in similar problems. Beginning with the basics of single-cell experiments and data analysis (Chapter 1), the thesis then considers how single-

cell atlas data is currently communicated in publication figures (Chapter 2). Next, the thesis introduces Vitessce, a web-based interactive visualization framework for single-cell data, including spatially-resolved and multimodal data (Chapter 3). Chapter 4 describes two applications of Vitessce which enable visual comparisons of subsets of single-cell atlas data, in the forms of query-versus-reference and case-versus-control comparisons. Chapter 5 describes applications of Vitessce for specific types of spatially-resolved data: segmentations of functional tissue units and spatial enrichments of co-occurring cell types. Chapter 6 details the interactive visualization of large-scale chromatin accessibility datasets, including how such techniques have been integrated into Vitessce. Finally, Appendices A through I contain methodological and technical details underlying these projects.

This thesis research has not only contributed targeted visual analytics approaches intended for single-cell data analysts to use directly, but also along the way we have made standalone contributions to topics in information visualization (Chapter 2 and Appendices A-C). These projects have resulted in the ideation of numerous future directions, which have been woven into multiple discussion sections within this thesis. Despite the focus of this work on a small set of topics in the single-cell domain, we anticipate that the methodologies herein might propel an expanded research agenda which considers important emerging single-cell topics (*e.g.*, perturbation screening, immune repertoire sequencing, consideration of genetics; Section E.2) and beyond (*e.g.*, in domains such as computational chemistry or climate research).

7.2 Research outlook

We supplement the discussions of follow-up research topics intertwined with the preceding thesis chapters with a broader research outlook for the field. First, there are important topics in single-cell data analysis which my research has not directly addressed, yet have been articulated by our collaborators in the domain and which are well-suited to be addressed through visualization. For example, during the Polyphony project (Chapter 4), we identified the unmet need for tools which help users to understand the quality of a given reference atlas dataset and whether a particular reference dataset is a good fit for integration with a given query dataset. Recently-developed metrics for benchmarking data integration approaches[279] could be integrated into a visual analytics system to help answer such questions. Exciting types of data and analyses are emerging in the domain, such as the aforementioned perturbation screening and immune repertoire sequencing data types, which present opportunities for the development of new visualization approaches. Opportunities to build tools that better support single-cell proteomics and metabolomics data visualization and improve the interpretability of this data (*i.e.*, beyond direct mass-to-charge ratios) remain, and could incorporate techniques from existing tools such as Metaspace[336]. Beyond support for additional experimental data types, future work in visual analytics should be geared towards improving the interpretability of specific bioinformatic methods for single-cell data, enhancing the usage of powerful methods such as LEMUR[51] which is designed for analysis of multi-condition single-cell data. Yet in addition to tools for recent bioinformatic methods, our interviews with biologists and clinicians revealed that even well-established analysis methods remain challenging to use without programming expertise and often require usage of multiple disjoint interactive tools.

Spatially-resolved data, an extremely active area for both data generation and bioinformatic tool development efforts, remains underutilized by biologists and clinicians who currently primarily look to this data type for spatial validation of findings from non-spatial analyses. Furthermore, as atlas-scale single-cell datasets containing profiles of samples from many tissue donors has emerged, so too have synergies between single-cell data analysis and patient-level data analysis[202]. Finally, throughout this thesis research we have identified the need to develop single-cell visualization tools which better enable exploration of atlas data via both top-down (*i.e.*, from atlas to experiment) and bottom-up (*i.e.*, from experiment to atlas) workflows.

Two aspects of this work that will undoubtedly persist are 1) the challenge of linking fresh experimental data with existing ontologies and biological literature & knowledge bases, and 2) the challenge of ever-increasing dataset sizes[374]. However, in the spirit of the nested model[316], future researchers can continue to stave off impending dataset size challenges by formulating clever approaches to data transformation that fit the user goals identified in this thesis (Table 1.1) or through similar formative interviews or literature reviews. For instance, rather than spending effort to increase the scalability of dimensionality reduction scatterplots, researchers should look to alternative approaches that are more *visually* scalable, such as the contour plot described in Section 4.5.5, or should develop a new approach, for example, based on the metrics of Manz *et al.*[290]. Efforts to develop a taxonomy and map the design space for single-cell data visualization (for both experiment- and atlas-scale data; Section 2.5.1) could also be helpful in identification of useful visualization designs to pursue in interactive settings. Intersections with existing more general taxonomies such as the comparative visualization taxonomy of Gleicher[171] would similarly be promising to explore, as they suggest, for instance, an opportunity to use relatively

less-explored comparative designs involving superposition and direct encoding. In a similar vein, future researchers should continue to consider prior research which may have been overlooked and identify opportunities to combine prior conceptual approaches with modern technical capabilities to address domain problems. For example, our work in Use-Coordination (Appendix A) merges a conceptual model from two decades ago with a current software architecture approach. There will always be emerging technologies to apply and domain problems to address, using both old and new conceptual frameworks. It has been fruitful to pursue research at the intersection of data visualization and the application area of single-cell biology, and I would encourage other researchers to explore the countless other interdisciplinary niches at which contributions can push our collective scientific understanding forward.

A

Use-Coordination

A.1 Preface

Portions of this chapter have been adapted from a previous publication:

- Use-Coordination: Model, Grammar, and Library for Implementation of Coordinated Multiple Views[227]

I was the primary author of the Use-Coordination manuscript, with contributions

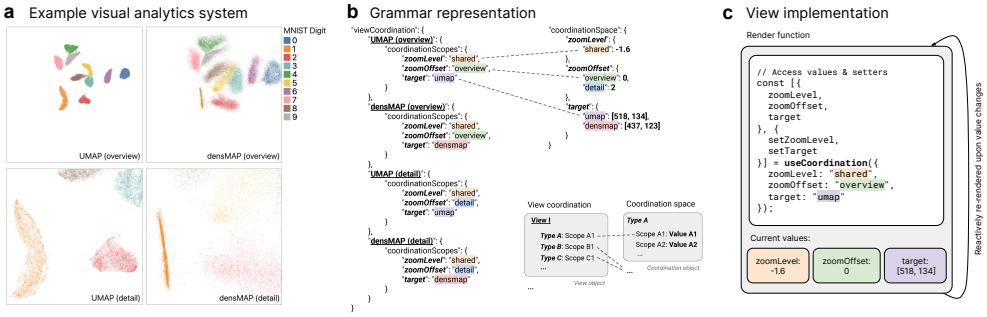

Figure A.1: Example of a visual analytics system that implements coordinated multiple views with the `use-coordination` software library. (a) The system supports comparison of two dimensionality reduction results using four scatterplots linked on subsets of properties. (b) The coordination specification defines which views are linked to which properties using a JSON grammar representation. Dotted lines illustrate that the first view "UMAP (overview)" is mapped to particular coordination scopes in the coordination space. (c) Views use their coordination scope mappings to obtain values and setter functions from the parent system that manages the coordination specification. Developers can assume that the render function for each view will re-execute upon changes to the values in the mapped coordination scopes.

from Trevor Manz and Nils Gehlenborg.

A.2 Abstract

Coordinated multiple views (CMV) in a visual analytics system can help users explore multiple data representations simultaneously with linked interactions. However, the implementation of coordinated multiple views can be challenging. Without standard software libraries, visualization designers need to re-implement CMV during the development of each system. We introduce `use-coordination`, a grammar and software library that supports the efficient implementation of CMV. The grammar defines a JSON-based representation for an abstract coordination model from the information visualization literature. We contribute an optional extension to the model and grammar that allows for hierarchical coordination. Through three use cases, we show that `use-coordination` enables implementation of CMV in systems containing

not only basic statistical charts but also more complex visualizations such as medical imaging volumes. We describe six software extensions, including a graphical editor for manipulation of coordination, which showcase the potential to build upon our coordination-focused declarative approach. The software is open-source and available at https://use-coordination.dev.

A.3 Introduction

Coordinated multiple views (CMV) is a visual analytics technique in which interactions in one visualization are automatically reflected in other linked visualizations. CMV is especially powerful during exploratory data analysis as it can help users to discover relationships, make comparisons, and simultaneously view details plus context.

Research on CMV began as early as the 1970s with methods for linking static views[73]. This was followed by a focus on the development of theoretical models which would enable flexible coordination implementations[73,299,322,88]. Several such models were introduced alongside software implementations that are now obsolete. A survey by Roberts *et al.*[361] characterized data structures and underpinning technologies for CMV as an "afterthought."

The translation of theoretical models into shared open-source software implementations has remained under-studied by the visualization research community. As a consequence, visualization tool developers must devise custom implementations each time they want to integrate a technique such as CMV. This places a burden on not only developers, but also on users who may encounter a wide range of *ad hoc* implementations of techniques.

To address this problem, we contribute the `use-coordination` grammar and software library (Figure A.1). Our approach implements the Abstract Model for coordination proposed by Boukhelifa *et al.*[88]. Additionally, we formalize an extension of the model that enables the specification of a hierarchy of coordination. We define the grammar in the form of a JSON-based domain-specific language (DSL) that is declarative and programming language-agnostic. The `use-coordination` library is implemented in JavaScript and can be integrated into React-based web applications, and we provide an alternative implementation for Python. The JavaScript library is accompanied by modules for visualization and editing of coordination specifications using their graph-based representation, for integration with provenance tracking approaches[128], and for construction of specifications imperatively via an object-oriented API. We anticipate `use-coordination` being useful to researchers aiming to support CMV in visual analytics systems by leveraging a common implementation that is grounded in a theoretical model from the information visualization literature. The software is open-source and available with accompanying documentation at https://github.com/keller-mark/use-coordination. We demonstrate the utility of the library through use cases with statistical plots and medical imaging visualizations.

A.4 Design Goals

We had several design goals based on our experiences implementing CMV in visual analytics tools for biomedical data[228,119] (Chapters 3 and 4):

DG1. Declarative grammar representation. Declarative representations of state enable serialization for use cases such as database storage, programming lan-

guage portability, reproducibility, provenance tracking, and multiplayer interactions. We aim for a coordination model that has a declarative representation which can be serialized to a JSON-based DSL format.

DG2. Reactive updates. Today's most popular user interface development frameworks are designed for reactive component-based architectures. In these architectures, a tree of view components re-render reactively, in response to changes in state values. The reactive paradigm simplifies the mental model for developers who do not need to worry about component state becoming stale thanks to efficient component tree diffing algorithms that are abstracted by the framework. In JavaScript, this class of frameworks includes React[5], Vue[10], Svelte[8], and Solid[7]. The reactive architecture contrasts with earlier publish-subscribe architectures that rely on explicit event subscriptions. Integrating event-based state management approaches into reactive frameworks can be unnatural, forcing the developer to revert to a more complicated mental model in which there is a risk of component state becoming stale (*e.g.*, due to accidental lack of publication or subscription). We aim for a coordination model and library that are designed for reactive component rendering architectures. As a corollary, view implementations should remain independent of one another such that views do not need to communicate directly to achieve coordination.

DG3. Decoupled from data representation and view layout. For maximum flexibility, coordination should be decoupled from the processes of data representation (and by extension, loading and transformation) and view layout. Not only does this allow developers to use methods of their choice (*i.e.*, view layout via CSS or a specialized JavaScript library) and visualize diverse data types (*e.g.*, tables, images,

trees, graphs), it also follows the Unix philosophy of modular software tools that have a singular focus.

DG4. Arbitrary coordination types. Coordination should not be limited to a predefined set of properties. Developers should be able to coordinate any part of the state of an application for maximum flexibility.

A.5 Related Work

CMV techniques have been studied by many visualization researchers over the years. We refer to Roberts[361] and Scherr[382] for two early surveys.

A.5.1 Coordination Models

Roberts[361] describes multiple model architectures for coordination, including data-centric and model-view-controller (MVC) architectures. Data-centric approaches are powerful but to our knowledge all existing approaches in this category are limited to tabular data representations. In contrast, Boukhelifa *et al.*[88] contribute the Abstract Model for coordination which is not data-centric and therefore more general than prior approaches. Multiple visualization systems, including CViews[89] and ManiVault[425], implement this model, but there is not a standalone implementation nor a standard declarative representation for the model state. Further, existing implementations of CMV use event-based approaches for information propagation, which are unnatural to use with reactive user interface frameworks such as React.

A.5.2 Grammars for Coordination

Harth *et al.*[190] and Chen *et al.*[113] both describe the goal of enabling CMV across views implemented using different plotting libraries using JSON- and natural language-based grammars, respectively. Both approaches are event-based with predefined sets of supported actions and both make assumptions about tabular data types. Neither encodes key rudiments of the Abstract Model[88] such as coordination types and coordination scopes. Though both are open-source, neither is available as a documented reusable software library through a package repository such as NPM.

A.5.3 Interaction Orchestration

Interaction orchestration can be considered as a special case of CMV in which all coordination types represent interactions such as selections and brushing. The Vega-Lite Grammar of Interaction[378] supports declarative specification of coordinated selections comprising points and intervals. The Grammar of Interaction is powerful but remains limited to the scope of a Vega-Lite specification (*i.e.*, unable to orchestrate across multiple specifications or outside Vega-Lite), does not support arbitrary value schema, and supports only single- or all-view coordination. The Dynamically Interactive Visualization (DIVI)[390] approach infers configurations for coordinated interactions from views defined as SVGs. The usage of SVG enables implementation of CMV with minimal developer overhead, but comes with scalability limitations and precludes coordination of non-SVG views such as tables or input elements. Both the Vega-Lite and DIVI approaches are implemented using event-based models.

A.5.4 Data Flow Models

A number of data flow approaches for coordinated visualization have been proposed, including VisFlow[461] and sMolBoxes[416]. Data flow systems visually encode operations on data, helping users to understand which views are linked on which data elements and rendering properties. In contrast, more general visual analytics systems, even those which implement CMV, do not typically include such a visual encoding. Similar to data-centric coordination models, data flow models make assumptions about data representations (*e.g.*, that data is tabular). To our knowledge, there are not standard declarative representations for data flow models. We note that research on visualization for data flow models is complementary to research on coordination and may inform how coordination model state is communicated to users.

A.6 The Use-Coordination Model and Grammar

The `use-coordination` model can be seen as an extension of the Abstract Model contributed by Boukhelifa *et al.*[88,89]. The model has also been informed by our earlier usage of the Abstract Model to implement CMV in the Vitessce visual analytics framework[228]. We formalize the model as a JSON-based grammar in terms of each rudiment of coordination:

Coordination Type. A coordination type is a name for a property being coordinated (*e.g.*, `zoom` or `colormap`).

Coordination Scope. A coordination scope is a named instance of a coordination type. For example, a visual analytics system with two different scatterplots display-

ing different projection results may have two coordination scopes for the `zoom` coordination type, `zoom.UMAP` and `zoom.TSNE`, enabling the user to zoom in these plots independently. We do not associate any semantics with coordination scope names, although particular visual analytics systems or view implementations may.

Coordination Value. Each coordination scope is mapped to a value, for example `zoom.UMAP: 10`.

Coordination Object. A coordination object encapsulates the coordination scopes and values for a given coordination type. A coordination object may contain zero or more coordination scopes. Coordination scope names must be unique within a coordination object but may be reused across coordination objects.

Coordination Space. The coordination space contains the coordination objects for all coordination types in the system (Figure A.1b).

View Coordination. Views typically correspond to plots or other standalone widgets in a visual analytics system. Each view may have a mapping from coordination types to coordination scopes (Figure A.1). Depending on their implementation, views may support a subset of the coordination types that are present in the coordination space. During rendering, each view performs a lookup (in the coordination space) for the coordination values that correspond to its current coordination scope mappings.

Hierarchy of Coordination. Prior work has discussed the need for hierarchical coordination, though did not formalize a multi-level model. Boukhelifa and Rodgers note, "...theoretically, any process could be coordinated with anything,"[89] and Mc-

Donald *et al.* state, "The hierarchical or graph structure of the Scene will often reflect a similar hierarchy in the Subjects,"[299]. Our grammar extends the Abstract Model[88] to support a hierarchy of coordination. First, we allow each view to be associated with more than one scope per coordination type (*i.e.*, an array of scopes). Next, a `coordinationScopesBy` property can be specified for each view, which defines scope → scope mappings, including between scopes within different coordination objects. For example, scopes representing bars selected in a bar plot can be mapped to scopes representing colors, enabling users to define color encodings per selected bar. To our knowledge, such an extension of the Abstract Model[88,89] has not previously been formalized. Further details about the JSON representation for hierarchies of coordination are available in Section 2 of our Supplemental Materials.

In accordance with our design goals, intentionally absent from the grammar are events (DG2), layout- and data- related properties (DG3), and view-view mappings (DG5).

A.7 Software Implementation

We contribute an open-source software library called `use-coordination`, which implements the model and grammar defined in Section A.6. The library follows the React "hooks" paradigm in which composable functions abstract away complex state management features. Developers of visual analytics systems can use the library by defining a *coordination specification* (*i.e.*, an instance of the grammar) and defining each view as a React component instance that is associated with a unique identifier. Within each view, developers can use the provided hook functions to perform lookups of coordination values from the coordination space and to obtain "setter" functions

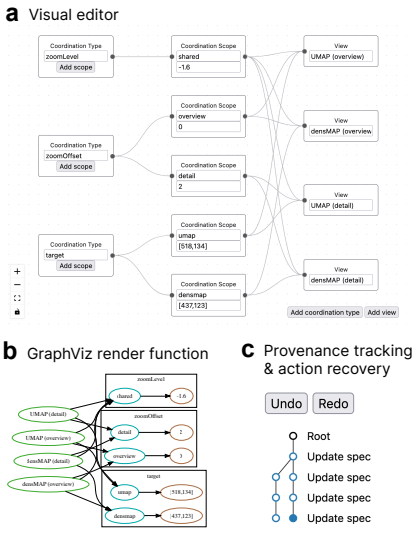

Figure A.2: Extensions of `use-coordination`. a) Our visual editor enables users to modify the coordination specification without knowledge of JSON. b) The coordination specification can be rendered to the GraphViz DOT format for static visualization. c) User interactions in a visual analytic system can be tracked through optional integration with Trrack[128].

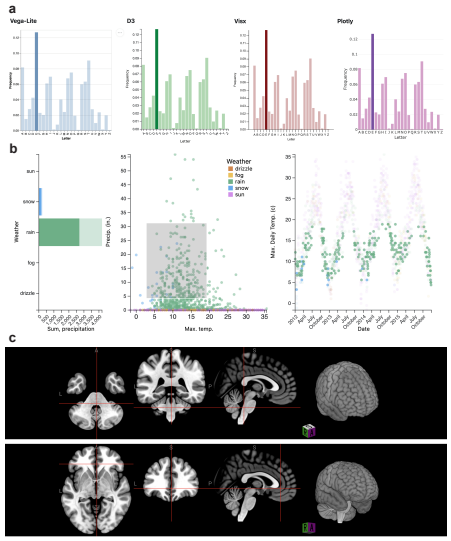

Figure A.3: Use cases. a) Coordinated selections across Vega-Lite, D3, Visx, and Plotly bar plot implementations. b) Seattle weather example in which brushing interactions in one plot (the middle scatterplot) are coordinated with the other two plots. c) Coordination of 2D (crosshair position) and 3D (azimuth) parameters of a medical imaging volume.

that enable updating the values in the mapped coordination scopes.

Software Extensions

The portability and declarative nature of the `use-coordination` grammar have proven to be powerful in enabling the straightforward development of additional software modules (Figure A.2).

Python and non-React implementations. Our approach and JSON representation generalize beyond React and JavaScript. First, we demonstrate an implementation of the `useCoordination` hook function for Solid[7] using its *signals* pattern for reactivity[37]. Second, we provide `with-coordination`, a Python implementation that coordinates Jupyter widget views[39]. Importantly, the software modules that follow in this section are designed around the shared JSON grammar, such that each module is compatible with all implementations.

Programmatic specification. On top of the declarative JSON representation, we have developed an object-oriented API for incremental construction and manipulation of coordination specifications. This API enables users to define objects corresponding to views and coordination scopes. Subsequently, users can pass these objects to functions such as `linkViews`, avoiding errors associated with direct JSON manipulation. At any point, users can convert their specification object to JSON.

Node-link diagrams. The `use-coordination` model takes the form of a graph data structure, which can be visualized as a node-link diagram. We provide a function to convert coordination specifications from JSON to GraphViz[147] `DOT` format for

visualization (Figure A.2b).

Visual editor. We provide a visual flow-based editor based on the graph structure of the model. This editor is implemented as a React component using the ReactFlow[6] library, enabling it to be embedded into visual analytics applications so that end-users can modify the coordination specification *in situ* (Figure A.2a).

Provenance tracking. The serializable nature of the JSON-based coordination specification lends itself to integration with provenance tracking and action recovery approaches. We provide a utility function to help integrate `use-coordination` with Trrack[128] (Figure A.2c).

Schema validation. Coordination types can optionally be associated with schema to which their values must conform. We implement a React provider component which supports validation via Zod[11] schema. Schema validation errors can be caught and handled by the parent application to display friendly error messages and prevent invalid data from propagating to views.

A.8 Use Cases

We demonstrate `use-coordination` through three use cases. Based on our experiences using the library, we also provide detailed usage guidance in Section 3 of our Supplemental Materials.

A.8.1 Use Case 1: Multiple Plotting Libraries

Visual analytics systems may integrate multiple plotting libraries. There is often a need to link elements across plots regardless of how they are implemented, a use case which `use-coordination` supports. We implement a bar plot using four different plotting libraries: Vega-Lite[378], D3[87], Visx[9], and Plotly[4] (Figure A.3a). With `use-coordination`, we coordinate bar selections across all four plots. The user can dynamically modify which subsets of plots are coordinated together via our visual editor extension (Figure A.2).

A.8.2 Use Case 2: Seattle Weather

We demonstrate that `use-coordination` can be used to link brushing interactions across plots. We reproduce Figure 1 of Snyder and Heer[390] in which the Seattle weather dataset is visualized using a bar plot and two scatterplots (Figure A.3b). As the user brushes (in the middle scatterplot view) a region containing data points corresponding to low temperature and high precipitation, the data points are highlighted in all three plots.

A.8.3 Use Case 3: Medical Imaging

Our approach is not limited to coordination of visualizations of tabular data. We visualize a medical imaging volume using two Niivue[3] instances (Figure A.3c). We link one of three 2D parameters (X-axis crosshair) and one of two 3D parameters (azimuth) via `use-coordination`. This coordination enables the user to simultaneously visualize six 2D slices (two of which are identical) and view the volume from two different angles.

A.9 Discussion

With `use-coordination`, we contribute the concept of a declarative coordination specification that is decoupled from both data representation and view layout. We provide a software ecosystem around such specifications oriented towards visual analytics system developers. The coordination specification is grounded in an established coordination model[88] whose influence on prior works in the visualization field demonstrates wide applicability.

The most closely related works are from Harth *et al.*[190] and Chen *et al.*[113], yet both differ from `use-coordination` in terms of design goals and conceptual models. These works are based on event- or action-based models which stand in contrast to the declarative and reactive approach that we adopt (DG1, DG2). In comparison to Harth *et al.*, our work also differs in that it is decoupled from data representation and view layout (DG3). Compared to the natural language-based grammar of Chen *et al.*, we developed a declarative JSON-based DSL (DG1). This declarative approach makes it easy to implement multiple systems around the DSL. For instance, our grammar has the potential to serve as a compilation target for natural language-based approaches involving large language models.

The simplicity of our approach comes with trade-offs. For example, we allow for coordination types to have arbitrary names and value schema (DG4) and we decouple coordination from view layout (DG3). While this freedom enables the integration of `use-coordination` with any selection, interaction, and layout approaches, it could present challenges to reuse of coordination specifications across visualization system boundaries or to meta-analyses that rely on standardization of coordination type names with associated semantics. Despite this limitation, we emphasize that vi-

sualization developers could define their own set of standard coordination types with domain-specific semantics (*e.g.*, `biomarker-selection` for biological data visualization). Our grammar does not explicitly include aspects such as transitions/easing or composite brushing[294], which would be up to the developer to implement (coordinated or otherwise). We note that developers may define more fully-featured libraries on top of `use-coordination` that are coupled to particular view layout approaches or coordination type semantics.

A.10 Conclusion

In this work, we present the `use-coordination` model, grammar, and software library. We demonstrate that standardizing the representation of coordination through a declarative JSON-based grammar enables the development of software extensions that simplify the implementation and reasoning about CMV. Our use of reactive software paradigms and initial extensions (Section A.7) reduce redundant work and lower barriers to entry across multiple platforms and languages. An important question for future work is how to effectively communicate coordination to end-users. We anticipate our work serving as a tool to address this and other human-centered questions.

`Use-coordination` is open-source and our software packages are available through the NPM and PyPI package repositories. The software is accompanied by tests, documentation, and examples. Appendix B contains further details about the JSON grammar and the model that we have developed to represent hierarchies of coordination.

B

Supplementary Information for Use-Coordination

B.1 Preface

This chapter contains supplementary information for Appendix A. The contents have been adapted from the supplementary information associated with the corresponding

publication:

- Use-Coordination: Model, Grammar, and Library for Implementation of Coordinated Multiple Views[227]

B.2 JSON representation

This section documents our JSON representation for coordination specifications.

B.2.1 Grammar formalization

We provide formal grammar specifications using both extended Backus–Naur form (EBNF) (Listing 1) and TypeScript type definitions (Listing 2). In contrast to EBNF which expresses syntax alone, the TypeScript type definitions additionally express some grammar semantics that are enforced during type-checking. Specifically, coordination types and coordination scopes that are referenced within the `viewCoordination` property are required to exist within the `coordinationSpace` property of the coordination specification.

```
<coordinationSpec> ::= "{"
  ("key" ":" (<string> | <number>) ",")?
  ("coordinationSpace" ":" <coordinationSpace> ",")?
  ("viewCoordination" ":" <viewCoordination> ",")?
"}"
<coordinationSpace> ::= "{"
  (<coordinationType> ":" <coordinationObject> ",")*
"}"
<coordinationObject> ::= "{"
  (<coordinationScope> ":" <coordinationValue> ",")*
"}"
<viewCoordination> ::= "{"
  (<viewId> ":" <viewObject> ",")*
"}"
<viewObject> ::= "{"
  ("coordinationScopes" ":" <coordinationScopesMapping> ",")?
```

```
  ("coordinationScopesBy" ":" <coordinationScopesByMapping> ",")?
"}"
<coordinationScopesMapping> ::= "{"
  (<coordinationType> ":"  <coordinationScopeStringOrArray> ",")*
"}"
<coordinationScopesByMapping> ::= "{"
  (<coordinationType> ":" "{"
    (<coordinationType> ":" "{"
      (<coordinationScope> ":" <coordinationScopeStringOrArray> ",")*
    "}")*
  "}")*
"}"
<coordinationScopeStringOrArray> ::= <coordinationScope> | "[" (<coordinationScope> ",")* "]"
<coordinationType> ::= <objectKey>
<coordinationScope> ::= <string>
<coordinationValue> ::= <objectValue>
<viewId> ::= <objectKey>
<json> ::= <primitive> | <container>
<primitive> ::= <number> | <string> | <boolean>
<string> ::= <quote> (<alphanumeric> | <symbol>)* <quote>
<quote> ::= "\""
<alphanumeric> ::= <digit> | <alphabetic>
<digit> ::= [0-9]
<alphabetic> ::= [a-z] | [A-Z]
<symbol> ::=  "|" | " " | "-" | "!" | "#" | "$"
| "&" | "(" | ")" | "*" | "+" | "," | "-" | "."
| "/" | ":" | ";" | ">" | "=" | "<" | "?" | "@"
| "[" | "\\" | "]" | "^" | "_" | "`" | "{" | "}"
| "%
<number> ::= <integer> | <float>
<integer> ::= <digit>+
<float> ::= <digit>+ "." <digit>+
<boolean> ::= "true" | "false"
<container> ::= <object> | <array>
<array> ::= "["  (<json> ",")* "]"
<object> ::= "{" (<member> ",")* "}"
<member> ::= <objectKey> ":" <objectValue>
<objectKey> ::= <identifier> | <string>
<objectValue> ::= <json>
<identifier> ::= <identifierStart> <identifierEnd>*
<identifierStart> ::= <alphabetic> | "$" | "_"
<identifierEnd> ::= <identifierStart> | <digit>
```

Listing 1: Formalization of the grammar using extended Backus-Naur form (EBNF). We assume JSON5 syntax features for brevity (trailing commas allowed in arrays & objects and JavaScript identifiers allowed as object keys). The <json> non-terminal (line 34) may be substituted for the complete JSON specification.

```
type CoordinationSpace = {
  [coordinationScopeName: string]: {
    [coordinationScope: string]: unknown;
  };
};
type CoordinationScopesMapping<T extends CoordinationSpace> = {
  [K in keyof T]?: keyof T[K] | (keyof T[K])[];
};
type CoordinationScopesByMapping<T extends CoordinationSpace> = {
  [K1 in keyof T]?: {
    [K2 in keyof T]?: {
      [K3 in keyof T[K1]]?: keyof T[K2] | (keyof T[K2])[];
    };
  };
};
type ViewCoordination<T extends CoordinationSpace> = {
  [viewId: string]: {
    coordinationScopes?: CoordinationScopesMapping<T>;
    coordinationScopesBy?: CoordinationScopesByMapping<T>;
  };
};
type CoordinationSpec<
  CoordinationSpaceT extends CoordinationSpace,
  ViewCoordinationT extends ViewCoordination<CoordinationSpaceT>,
> = {
  key?: string | number;
  coordinationSpace?: CoordinationSpaceT;
  viewCoordination?: ViewCoordinationT;
};
function defineSpec<
  CoordinationSpaceT extends CoordinationSpace,
  ViewCoordinationT extends ViewCoordination<CoordinationSpaceT>,
>(
  spec: CoordinationSpec<CoordinationSpaceT, ViewCoordinationT>,
): CoordinationSpec<CoordinationSpaceT, ViewCoordinationT> {
  return spec;
}
```

Listing 2: Formalization of the grammar using TypeScript (v4.7.4) type definitions. Type-checking of coordination specifications can be achieved by passing a spec object into the defineSpec function (line 30).

B.2.2 Properties

coordinationSpace

The `coordinationSpace` property takes the form of a JSON object whose keys are the coordination types (as strings) and values are coordination objects (Listing 3). The keys of the coordination objects are coordination scopes (as strings) and values are the coordination values (*i.e.*, the values associated with each scope). The types of values that each coordination scope can take can be as simple as a single number or as complex as an array or object.

```
{
    "coordinationSpace": {
        "zoomLevel": {
            "shared": 3.3
        },
        "zoomOffset": {
            "overview": 0,
            "detail": 2
        },
        "target": {
            "umap": [0.57, -0.24],
            "densmap": [0.19, -0.01]
        }
    },
    // ...
}
```

Listing 3: Example coordination space which contains three coordination objects.

viewCoordination

The `viewCoordination` property takes the form of a JSON object whose keys are the view IDs (as strings) and values are objects with the property `coordinationScopes` (object that maps coordination types to coordination scopes for the parent view) (Listing 4).

```
{
    // ...,
    "viewCoordination": {
        "umapOverview": {
            "coordinationScopes": {
                "zoomLevel": "shared",
                "zoomOffset": "overview",
                "target": "umap"
            }
        },
        "densmapOverview": {
            "coordinationScopes": {
                "zoomLevel": "shared",
                "zoomOffset": "overview",
                "target": "densmap"
          }
        },
        "umapDetail": {
            "coordinationScopes": {
                "zoomLevel": "shared",
                "zoomOffset": "detail",
                "target": "umap"
          }
        },
        "densmapDetail": {
            "coordinationScopes": {
                "zoomLevel": "shared",
                "zoomOffset": "detail",
                "target": "densmap"
            }
        }
    }
}
```

Listing 4: Example view coordination property which contains four view objects, each with three coordination type-scope mappings.

`KEY`

The `key` property is an optional unique identifier for the specification, defined as a string or number. This property is intended to be used in situations in which the top-level `CoordinationProvider` component is used as a React "controlled" component

(*i.e.*, a parent of the provider is the source of truth for the specification and internal updates to the specification are communicated via the `onSpecChange` prop of the provider). For example, if a change in the value of `key` is detected, the specification will be re-validated (which is not guaranteed to occur upon other changes to the specification for performance reasons).

B.3 Hierarchy of coordination

We represent hierarchies of coordination using three key concepts: multi-scope mappings, multi-level scope mappings, and meta-coordination.

The base `useCoordination` hook function does not provide access to coordination values within hierarchies. Instead, we provide additional hook functions for access to values within hierarchies:

- `useCoordinationScopes` returns an ordered list of coordination scopes for a given view ID and coordination type. Scopes are omitted if their value in the coordination space is `null`.
- `useCoordinationObject` returns a mapping from (multiple) coordination scopes to their values for a given view ID and coordination type.
- `useCoordinationL1` returns nested coordination values and setter functions corresponding to coordination type-specific coordination scopes for each coordination type, one level deep. This requires a view ID, primary coordination type, and list of secondary coordination types as parameters.
- `useCoordinationScopesL1` returns a list of primary coordination scopes and a mapping between primary and secondary coordination scopes for a given view

ID, primary coordination type, and secondary coordination type.

- `useCoordinationL2` returns nested coordination values and setter functions, two levels deep. This requires a view ID, primary coordination type, secondary coordination type, and list of tertiary coordination types as parameters.

B.3.1 Multi-scope mappings

Multi-scope mappings enable a view to be mapped to more than one coordination scope per coordination type (Listing 5, Lines 11 and 16).

```
{
    "coordinationSpace": {
        "barSelection": {
            "S0": "A",
            "S1": "B"
        }
    },
    "viewCoordination": {
        "vegaLite": {
            "coordinationScopes": {
                "barSelection": ["S0", "S1"]
            }
        },
        "d3": {
            "coordinationScopes": {
                "barSelection": ["S0", "S1"]
            }
        }
    }
}
```

Listing 5: Example coordination specification in which each view is mapped to multiple coordination scopes for the `barSelection` coordination type.

B.3.2 Multi-level scope mappings

Views can use multi-level scope mappings to specify coordination type-scope mappings that are (conceptually) nested beneath other coordination type-scope mappings (Listing 6). Multi-level coordination is achieved by views that specify both `coordinationScopes` and `coordinationScopesBy` properties.

The `coordinationScopesBy` property takes the form of a JSON object whose keys are primary coordination types, and values are objects whose keys are secondary coordination types. The secondary coordination types are mapped to objects whose keys are primary coordination scopes and values are secondary coordination scopes (Listing 6, Lines 20-21 and 33-34).

```
{
    "coordinationSpace": {
        "barSelection": {
            "S0": "A",
            "S1": "B"
        },
        "barColor": {
            "C0": "#ff0000",
            "C1": "#ff00ff"
        }
    },
    "viewCoordination": {
        "vegaLite": {
            "coordinationScopes": {
                "barSelection": ["S0", "S1"]
            },
            "coordinationScopesBy": {
                "barSelection": {
                    "barColor": {
                        "S0": "C0",
                        "S1": "C1"
                    }
                }
            }
        },
        "d3": {
```

```
            "coordinationScopes": {
                "barSelection": ["S0", "S1"]
            },
            "coordinationScopesBy": {
                "barSelection": {
                    "barColor": {
                        "S0": "C0",
                        "S1": "C1"
                    }
                }
            }
        }
    }
}
```

Listing 6: Example of multi-level coordination in which each view maps each `barSelection` scope to a `barColor` scope.

B.3.3 Meta-coordination

Meta-coordination enables views to share coordination type-scope mappings while view implementations remain unaware of the existence of other views. Meta-coordination lifts coordination scope mappings into the `coordinationSpace` using a special `metaCoordinationScopes` coordination type (Listing 7). Meta-coordination and multi-level coordination can be used together with the `metaCoordinationScopesBy` coordination type (which is to `metaCoordinationScopes` as `coordinationScopes` is to `coordinationScopesBy`) (Listing 8). Without meta-coordination, arrays of multi-scope mappings would not be possible to share among multiple views, therefore preventing array modifications performed by one view from being shared with any other view(s).

In theory, meta-coordination can be recursive because `metaCoordinationScopes` takes the form of a coordination type within the `coordinationSpace`. However, our `useCoordination` hook functions are not implemented to support recursive meta-coordination.

```
{
    "coordinationSpace": {
        "zoomLevel": {
            "shared": 3.3
        },
        "zoomOffset": {
            "overview": 0,
            "detail": 2
        },
        "target": {
            "umap": [0.57, -0.24],
            "densmap": [0.19, -0.01]
        },
        "metaCoordinationScopes": {
            "A": {
                "zoomLevel": "shared"
            }
        }
    },
    "viewCoordination": {
        "umapOverview": {
            "coordinationScopes": {
                "metaCoordinationScopes": ["A"],
                "zoomOffset": "overview",
                "target": "umap"
            }
        },
        "densmapOverview": {
            "coordinationScopes": {
                "metaCoordinationScopes": ["A"],
                "zoomOffset": "overview",
                "target": "densmap"
          }
        },
        "umapDetail": {
            "coordinationScopes": {
                "metaCoordinationScopes": ["A"],
                "zoomOffset": "detail",
                "target": "umap"
          }
        },
        "densmapDetail": {
            "coordinationScopes": {
                "metaCoordinationScopes": ["A"],
                "zoomOffset": "detail",
                "target": "densmap"
```

```
            }
        }
    }
}
```

Listing 7: Minimal example of meta-coordination which lifts the type-scope mapping for the `zoomLevel` coordination type outside of the view scope mappings and into the coordination space, adding a layer of indirection. This example also demonstrates that views can be mapped to an array of `metaCoordinationScopes` which are applied first to last such that later meta-coordination scopes take precedence (if multiple contain mappings for the same coordination type).

```
{
    "coordinationSpace": {
        "metaCoordinationScopes": {
            "A": {
                "barSelection": ["S0", "S1"]
            }
        },
        "metaCoordinationScopesBy": {
            "A": {
                "barSelection": {
                    "barColor": {
                        "S0": "C0",
                        "S1": "C1"
                    }
                }
            }
        },
        "barSelection": {
            "S0": "A",
            "S1": "B"
        },
        "barColor": {
            "C0": "#ff0000",
            "C1": "#ff00ff"
        }
    },
    "viewCoordination": {
        "vegaLite": {
            "coordinationScopes": {
                "metaCoordinationScopes": "A",
                "metaCoordinationScopesBy": "A"
            }
        },
        "d3": {
            "coordinationScopes": {
```

```
                "metaCoordinationScopes": "A",
                "metaCoordinationScopesBy": "A"
            }
        },
        "barColorPicker": {
            "coordinationScopes": {
                "metaCoordinationScopes": "A",
                "metaCoordinationScopesBy": "A"
            }
        }
    }
}
```

Listing 8: Example that combines multi-level and meta-coordination.

We did not investigate alternative approaches to implement meta-coordination such as lexical scoping (as opposed to name-based). However, we note that a lexical scoping approach may prevent possible recursive meta-coordination.

B.4 Usage guidance

Through our experience implementing tools with use-coordination, we have compiled recommendations about how to define coordination types and their value schema.

B.4.1 Types of coordination types

We have identified several common types of coordination types that are used when building visual analytics systems.

Visual properties. These properties define visual marks or encoding-related information, such as colormaps or zoom levels.

Dataset identifiers. When there are multiple datasets available to users within a visualization system, a `dataset` coordination type can indicate which dataset to use

when rendering a particular visualization. A `dataset` value may also indicate a user-selected feature or subset of the data, such as a column of a dataframe to display in a histogram.

Ephemeral interaction parameters. These coordination types can be used upon interactions such as hovers or clicks, and then can be used to render tooltips or highlights. The values are typically identifiers that originate from a dataset.

Behavior modifiers. Behavior modifier coordination types can be combined with other coordination types to determine how values should be used. For example, if a first coordination type defines a range `[low, high]` to use for filtering values, a second behavior modifier coordination type might specify whether to interpret these values as `relative` or `absolute`. Behavior modifiers can be necessary to introduce as requirements for a visual analytics system evolve over time, if coordination types defined early-on must become more nuanced or take on additional functions.

Interface properties. These properties store state that is used to render the user interface, such as whether a dropdown is open.

B.4.2 Granularity of coordination types

Generality of coordination types. In the most extreme case of generality, coordination types could correspond to primitives like ”string” or ”integer”. However, in the base (*i.e.*, non-hierarchical) coordination model, each view can only be mapped to a single scope per coordination type, preventing specification of multiple string- or integer-type values for the same view. Therefore, we recommend coordination types

that are as general as possible while remaining semantically meaningful.

Specificity of coordination types. When defining coordination types, the developer must think outside the box to envision use cases in which coordination would make sense. It should not make sense for the values for two separate coordination types to be coordinated (in any envisioned use case); otherwise, these coordination types would be too specific and should have been merged into a single more general coordination type.

B.4.3 Coordination values

Value complexity should consider the smallest reasonable atomic unit to coordinate. System developers may be tempted to nest values such as image channel settings within a coordinated JSON object. However, this prevents 1) coordination of a subset of sub-values, and 2) may have performance implications, requiring React to perform more re-renders than necessary.

Start with base model. It is easiest to implement views that support the base (*i.e.*, non-hierarchical) coordination model. In complex visual analytics applications, the need for multi- and multi-level coordination can be identified via views that contain conceptual sub-views. Conceptual sub-views can take the form of juxtaposition or superposition, as in genome browser tracks or image layers, which each may have their own set of properties that make sense to coordinate independently of their sibling sub-views.

Globality/stability of data element references. References to data elements that appear in coordination values should avoid exposing implementation details such as internal indices. McDonald *et al.*[299] articulate this point well:

"Nearly all implementations of brushing scatterplots of which we are aware, including our own early work, share the essential feature (and failing)...that the correspondence between plots is represented by matching array indices. Unfortunately, it precludes even the most trivial extensions. For example, suppose the user wishes to simultaneously analyze three data sets: all the patients in a clinical trial of some drug, the male patients in the trial, and the female patients. Typical commercial implementations of brushing scatterplots either restrict the user to a single data set, or brush separate data sets separately, *i.e.*, brushing in a plot of all patients has no effect on plots of male patients or plots of female patients. If we think of brushing scatterplots in terms of direct manipulation, then it's obvious what wrong with the 'standard' implementation: There's no reliable representation for the identity of the underlying entity that's being brushed."

C

Bibliometric Analysis of Dimensionality Reduction Algorithm Usage

C.1 Preface

Portions of this chapter have been adapted from an in-progress manuscript titled: A Critical Analysis of the Usage of Dimensionality Reduction in Four Domains[104]. This

project was led by Dylan Cashman with contributions from myself, Hyeon Jeon, Bum Chul Kwon, and Qianwen Wang. While I contributed to all aspects of this project, here I focus on one aspect where I was the primary contributor: bibliometric analysis of dimensionality reduction method usage across subject areas. I was the sole data analyst and author of this section of the manuscript. These results are relevant to this dissertation because they inform us about the extent to which dimensionality reduction is used in biology and medicine (including single-cell biology) and how this usage compares to other fields.

C.2 Background

In the aforementioned survey on usage of dimensionality reduction (DR) in four domains (Biology, Chemistry, Physics, and Business), we sought to understand how results from the PCA, UMAP, and t-SNE algorithms are visualized in these domains (which are notably outside of visualization and computer science). However, logical follow-up questions are: How representative are the three selected DR methods (PCA, UMAP, and t-SNE) and the four subject areas (Biology, Chemistry, Physics, and Business) compared to all others? Are these three DR methods the most-used (among all DR methods)? To what extent does DR method usage differ by subject area? Especially in light of this survey having been conceptualized by computer scientists and bioinformaticians (as opposed to authors being embedded in all four domains of interest), these questions are fundamental to be able to contextualize the survey results.

To address these questions, I performed a bibliometric analysis of the Semantic Scholar citation network database[233], focusing on the last decade of publications across 20 subject areas and considering 78 DR methods[151]. I mapped each DR method

to a single publication in which it was first described, and then analyzed the extent to which each DR publication was cited by papers in each subject area (both overall and temporally), performing field size normalization to enable comparison using a percentile ranking approach[86]. The main takeaway of this analysis was that while t-SNE and PCA were within the top three most-cited DR methods by papers in 18 and 16 of the 20 subject areas, respectively, UMAP was only within the top three methods for two subject areas: Biology and Medicine.

C.3 Bibliometric Analysis Across Subject Areas

We aimed to use the academic citation network to assess usage of a large set of dimensionality reduction methods across all domain subject areas. We queried for all academic papers (excluding preprints) published since 2013 which cite those publications initially introducing one of 78 dimensionality reduction methods ("DR papers"). The 78 DR papers were manually curated by extending Table 1 of Espadoto *et al.*[151] (Table C.1). We queried for citing papers (*i.e.*, papers directly citing a DR paper) using the Semantic Scholar Academic Graph[233]. We excluded general methods such as "neural networks" which were considered in the original table from[151] but are used broadly as black boxes without investigation of latent space, as well as methods for which a single originating publication to reference could not be identified (Table C.3). We excluded citing papers in Computer Science, Mathematics, and Engineering from our analysis, as we are interested in understanding how DR methods are used in other subject areas.

Semantic Scholar maps papers to subject areas (each paper can be mapped to multiple subject areas) using two different methods. The first method uses subject area

information provided by the external source of the paper (*e.g.*, academic publishers or other publication aggregators). The second method uses a machine learning classifier that predicts subject areas based on paper title and abstract (English-language only). We use the latter (machine learning-based) method of subject area classification for consistency. We exclude papers which lack a subject area classification from our analysis. Note that due to the one-to-many nature of the mapping between papers and subject areas, the same paper may count towards multiple subject areas in our analyses (Figure C.6).

We present results in the form of counts, proportions, and percentile rankings. We use proportions and percentile rankings to address the problem that citation counts are not comparable across subject areas of different sizes. Further, these approaches mitigate biases that may exist in the Semantic Scholar database, which may be more representative of certain subject areas than others. We compute standard error for proportions using the binomial distribution. We compute percentile rankings of papers using the CP-EX method described by Bornmann and Williams[86], using the entire Semantic Scholar corpus to build cumulative percentages of papers with each citation count value in each subject area and year. A DR method which is cited by N papers from subject area S which has percentile rank P for year Y indicates that $P\%$ of papers published in subject area S in year Y received less than N citations. In contrast to proportions, the percentile ranking metric is not influenced by introduction of new DR methods (*i.e.*, using percentile rankings, we would not observe a decrease for a consistently-used method A simply due to introduction of some method B, despite a reduction in A's share of usage, Figure C.13).

C.3.1 Bibliometric Analysis Results

We first sought to quantify the usage of the 78 DR methods within each subject area based on direct citations. We find that there are 136,956 unique papers published between 2013 and 2023 which cite at least one of the DR papers in the dataset. Of those 136,956 papers, 74,720 have been mapped to at least one subject area via one of the two subject area mapping methods. 74,357 papers (99.5% of those with at least one subject area) have been mapped to at most three subject areas using either method (Figure C.4). The differences between the external subject area mappings and the predicted ("s2-fos-model") mappings are shown in Figure C.5. Using the latter subject area assignments, there are 21,249 citing papers assigned to at least one of 20 subject areas other than Computer Science, Mathematics, and Engineering (Figure C.7). We use these 21,249 papers and subject areas for the results that follow in this section.

Citation counts stratified by subject area and cited DR method are shown in Figure C.8. Through summation of citing paper counts within subject areas, we observe that the top ten areas citing the 78 DR papers are Biology (first), Environmental Science, Medicine, Psychology, Physics, Business, Economics, Chemistry, Materials Science, and Agricultural and Food Sciences (tenth). Instead summing across subject areas (*i.e.*, ignoring subject areas), we find that t-SNE, PCA, and UMAP are the three most highly-cited methods.

Acknowledging that citation counts are influenced by field size, we next consider metrics that are standardized to enable comparison across subject areas. Considering the number of citing papers as a proportion of all papers in each subject area, we instead see that the top ten areas are Environmental Science (1 in 596 papers cites a

DR method), Biology, Psychology, Geography, Linguistics, Geology, Physics, Business, Economics, and Chemistry (1 in 2,722 papers cites a DR method) (Figure C.7). When we stratify citation counts by cited method and subject area and convert them to rankings, it becomes apparent that while t-SNE and PCA appear in the top three methods for 18 and 16 of the 20 subject areas, respectively, UMAP is only within the top three methods for Biology and Medicine (Figures C.1 and C.2). Other notable findings are that N-MDS is the top-ranked method in Environmental Science and Geography, FA is the top-ranked method in Psychology and Education, and LDA is the top-ranked method in Agricultural and Food Sciences.

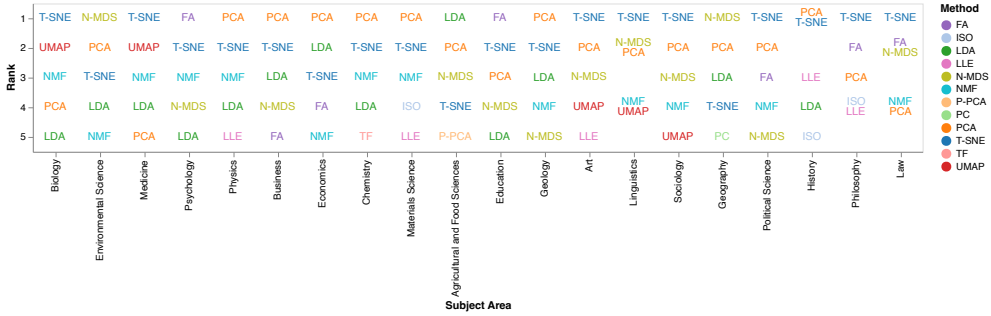

Figure C.1: Rankings based on papers citing dimensionality reduction methods (2013-2023). Methods are ranked based on the number of citing papers in each subject area. Subject areas are ordered left-to-right based on descending total count of papers citing the 78 considered dimensionality reduction methods. Ties are displayed in alphabetical order. Full method names can be found in Table C.1.

For ease of interpretation and visualization, we next consolidate lesser-cited DR methods by mapping those that do not appear in the top three methods cited within any subject area to the category 'Other'. This results in nine DR method categories: t-SNE, PCA, UMAP, NMF, LDA, N-MDS, FA, LLE, and 'Other' (Figure C.9). Using these nine top-used DR method categories, we next considered percent usage of each DR method within each subject area (Figure C.10). We find that the percent-

age of citations mapped to the 'Other' method category is largest (compared to the eight remaining categories) in the subject areas Environmental Science, Physics, Business, Economics, Chemistry, Materials Science, Education, Geology, History, and Philosophy, which suggests that there is not a single overwhelmingly dominant DR method used in these fields. Together, the top three DR methods per field are cited more than the bottom 75 methods in Biology, Medicine, Psychology, Agricultural and Food Sciences, Linguistics, Sociology, Geography, and Law (Figure C.10). This may suggest that in these subject areas only a small set of DR methods are established or accepted, or that data characteristics in such areas are amenable to certain DR methods.

We next consider how the 78 DR methods have been cited over time. We use percentile rankings of citation counts stratified by subject area and year to enable temporal comparison. Focusing on t-SNE, PCA, and UMAP in Biology, Physics, Business, and Chemistry, we find that percentile rankings for t-SNE, PCA, and UMAP increased between 2013 and 2022 in Biology, Physics, and Chemistry. In Business, percentile rankings for t-SNE and UMAP increased while those for PCA stayed consistent (Figures C.11 and C.12). These results are consistent with the publication of t-SNE in 2008 and UMAP being preprinted in 2018. The general trend between 2013 and 2022 across subject areas is that percentile rankings of DR methods have increased over time.

Bibliometric Analysis Discussion

This analysis is based on a one-to-one mapping between DR methods and DR papers. Therefore, the results may be sensitive to the DR paper mappings and do not capture cases in which methods have been introduced multiple times under different names or

in different fields independently. Another caveat is that this bibliometric analysis is limited to direct citations of DR papers, which ignores indirect citations (*e.g.*, papers which only cite a software implementation of a DR method) and non-citations (*e.g.*, papers which use and discuss a DR method without a formal citation). It may be the case that once a DR method becomes established or well-known in a subject area, researchers become less inclined to add a citation, which could obscure the impact of the most-used methods. Likewise, it may be the case that recently-established software citation practices such as Zenodo DOI usage could impact our results, including the temporal analysis.

As future work, this bibliometric analysis could be extended to map each DR method to multiple papers, accounting for methods introduced multiple times, as well as considering papers describing software implementations of methods. A coarse approach would be to consider citing papers from specific subject areas (*e.g.*, Computer Science, Mathematics, and/or Engineering) to be additional DR papers for the cited method (as opposed to domain papers). A more sophisticated approach could take advantage of citation intent, either computationally[123] or manually, as well as consider more than just direct citations. Further, datasets containing 'informal' or 'soft' citations could be used[144,213]. The bibliometric analysis could also be extended to consider whether citing domain papers contain a visualization which uses the cited DR method (*i.e.*, differentiating papers which plot DR results from those that do not). The analysis should be repeated using an alternative data source (*e.g.*, Scopus or Web of Science) to assess its robustness and reproducibility. In the future, the analysis could be extended temporally.

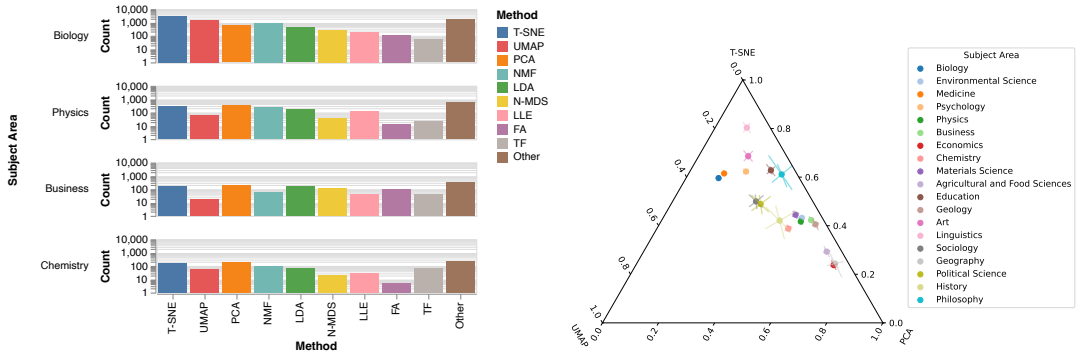

Figure C.2: Left: Counts of papers citing dimensionality reduction methods (2013-2023), including only the Biology, Physics, Business, and Chemistry subject areas. Subject areas are ordered by descending total citing paper counts. Methods that are not ranked in the top five for at least one of the four subject areas are mapped to the "Other" category. Expanded figure including all subject areas can be found in Figure C.9. Right: Ternary plot displaying proportional usage of t-SNE, PCA, and UMAP based on citing papers in each subject area (2013-2023). Standard error bars are displayed for all three axes of the ternary plot. Full method names can be found in Table C.1.

C.4 Code Availability

Scripts and other source code used for data processing and analysis described in Section C.3 are available at https://github.com/keller-mark/hd-vis-scripts.

C.5 Supplementary Figures

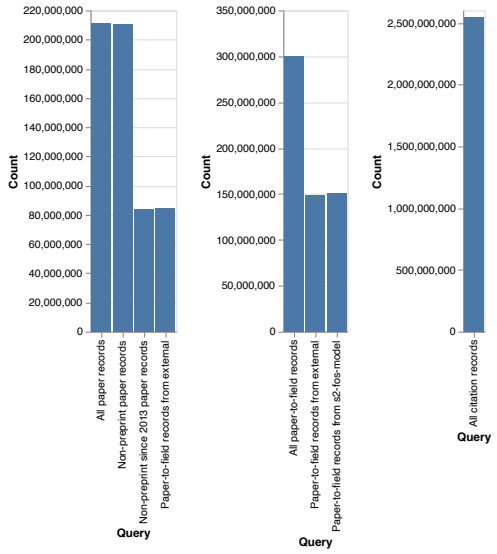

Figure C.3: Overview of record counts in database derived from Semantic Scholar Academic Graph. Left: queries of table containing paper records. Middle: queries of table containing paper-to-subject area mappings. Right: query of table containing citation mappings (citing-to-cited mappings).

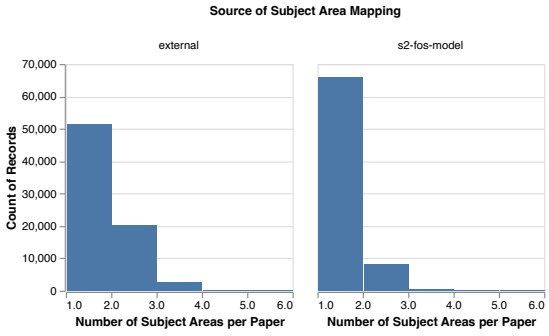

Figure C.4: Distribution of number of subject area assignments per paper for papers citing dimensionality reduction methods (2013-2023). Each paper in the Semantic Scholar Academic Graph may be assigned to multiple subject areas. Left: distribution of subject area assignments per paper when using external subject area sources. Right: distribution of subject area assignments per paper when using the Semantic Scholar predictions based on title and abstract.

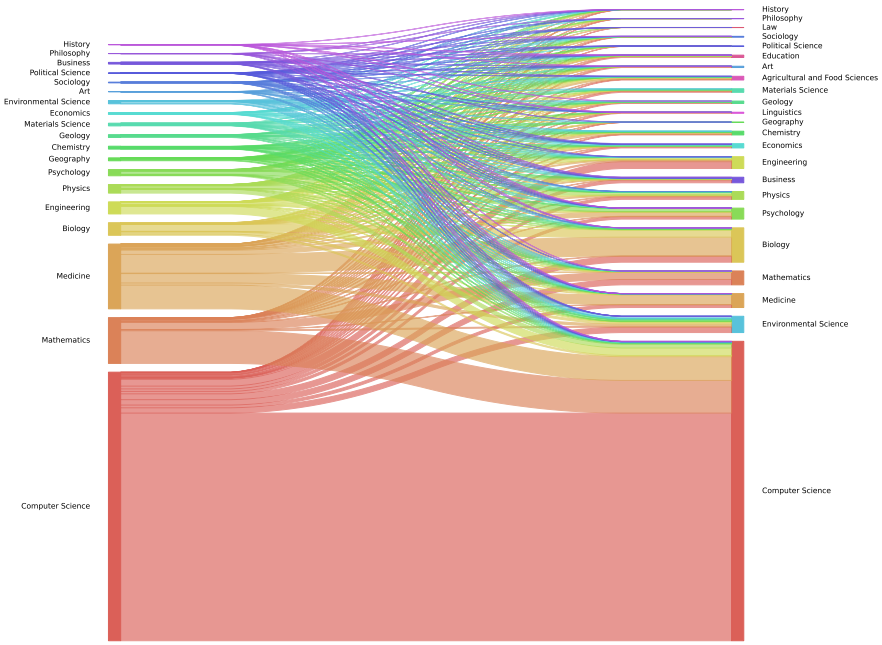

Figure C.5: Flow of subject area assignments from externally-sourced (left, "external") to internally-sourced (right, "s2-fos-model"). To account for papers assigned to multiple subject areas, each paper is considered six times in the sankey diagram.

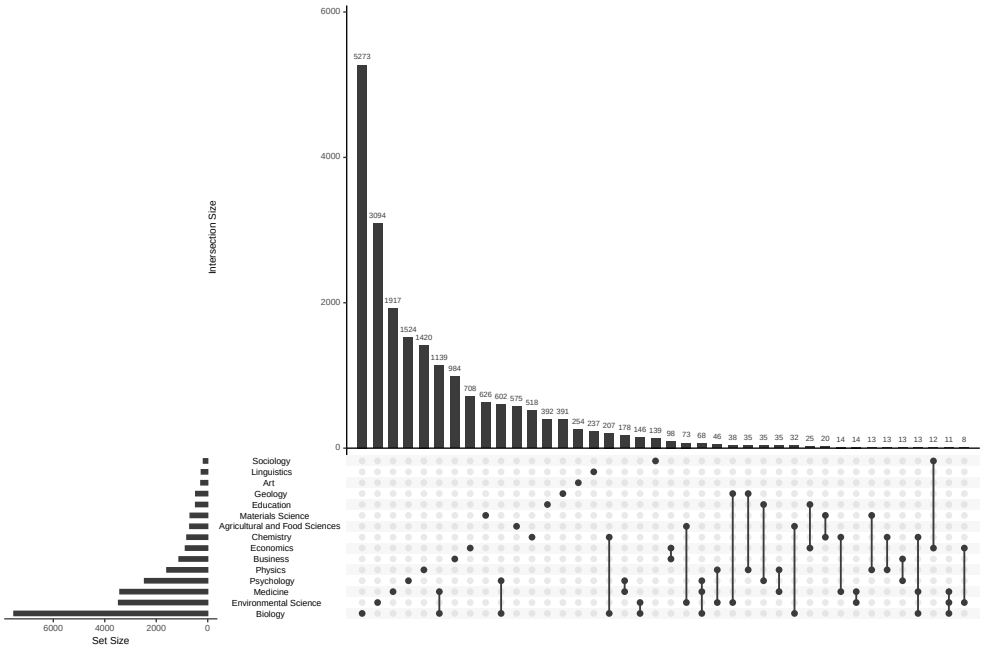

Figure C.6: Counts of papers assigned to sets of one or more subject areas. Top 15 subject areas shown.

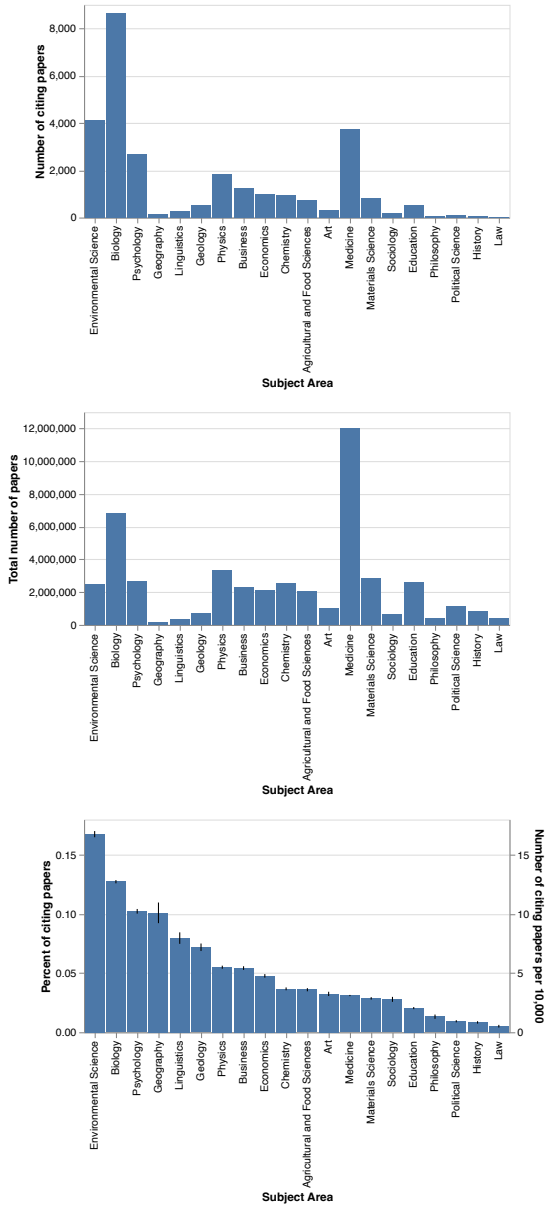

Figure C.7: Counts of papers in the Semantic Scholar Academic Graph stratified by subject area (2013-2023). Top: count of papers citing one of the 78 dimensionality reduction methods considered in the bibliometric analysis. Middle: count of all papers assigned to each subject area. Bottom: percentage of papers in each subject area citing one of the considered dimensionality reduction methods. The Subject Area axes are sorted by descending percentage.

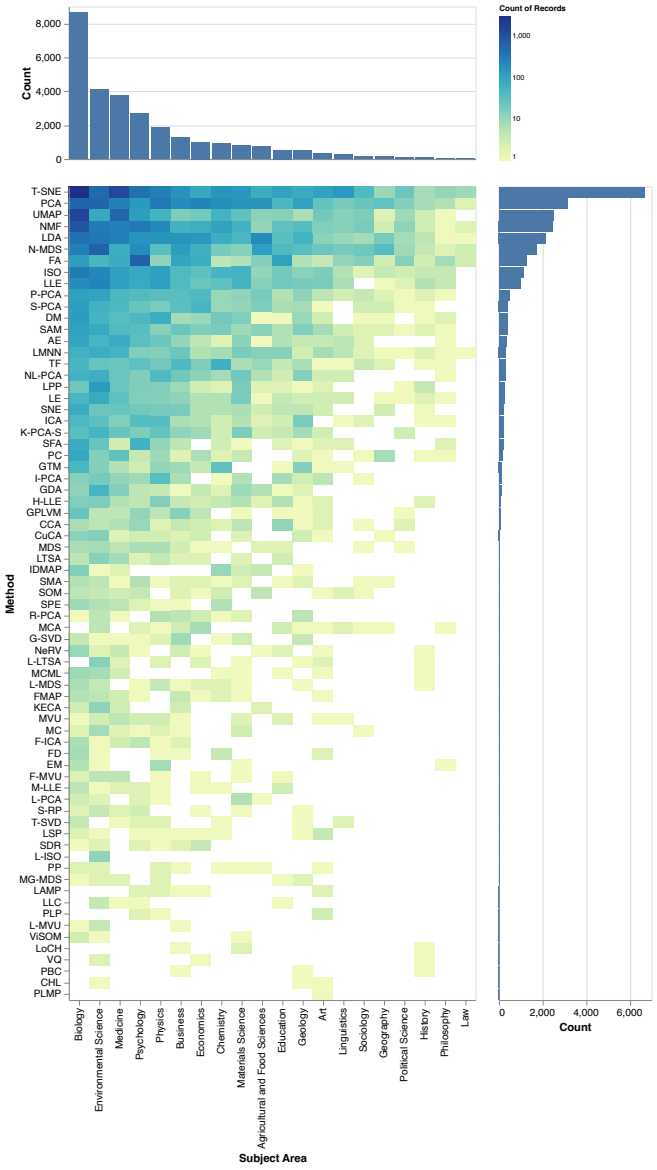

Figure C.8: Count of papers citing dimensionality reduction methods, stratified by subject area (2013-2023). The quantitative colormap is log-scaled. The Method and Subject Area axes are sorted by descending summed counts. Full method names can be found in Table C.1.

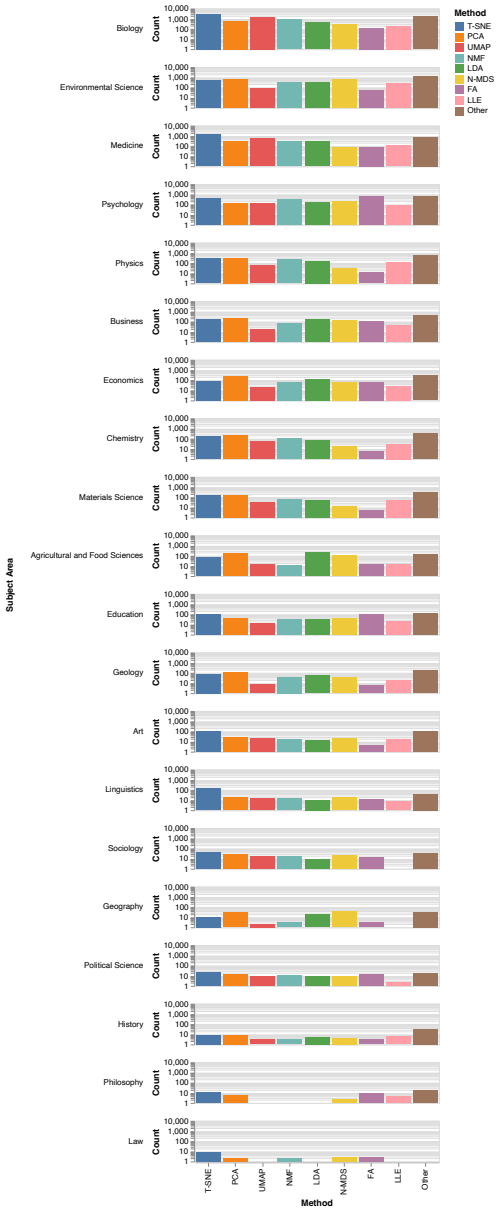

Figure C.9: Count of papers citing dimensionality reduction methods, stratified by subject area (2013-2023). The y-axis is log-scaled. Methods that are not ranked in the top three in at least one subject area are mapped to the "Other" category. Full method names can be found in Table C.1.

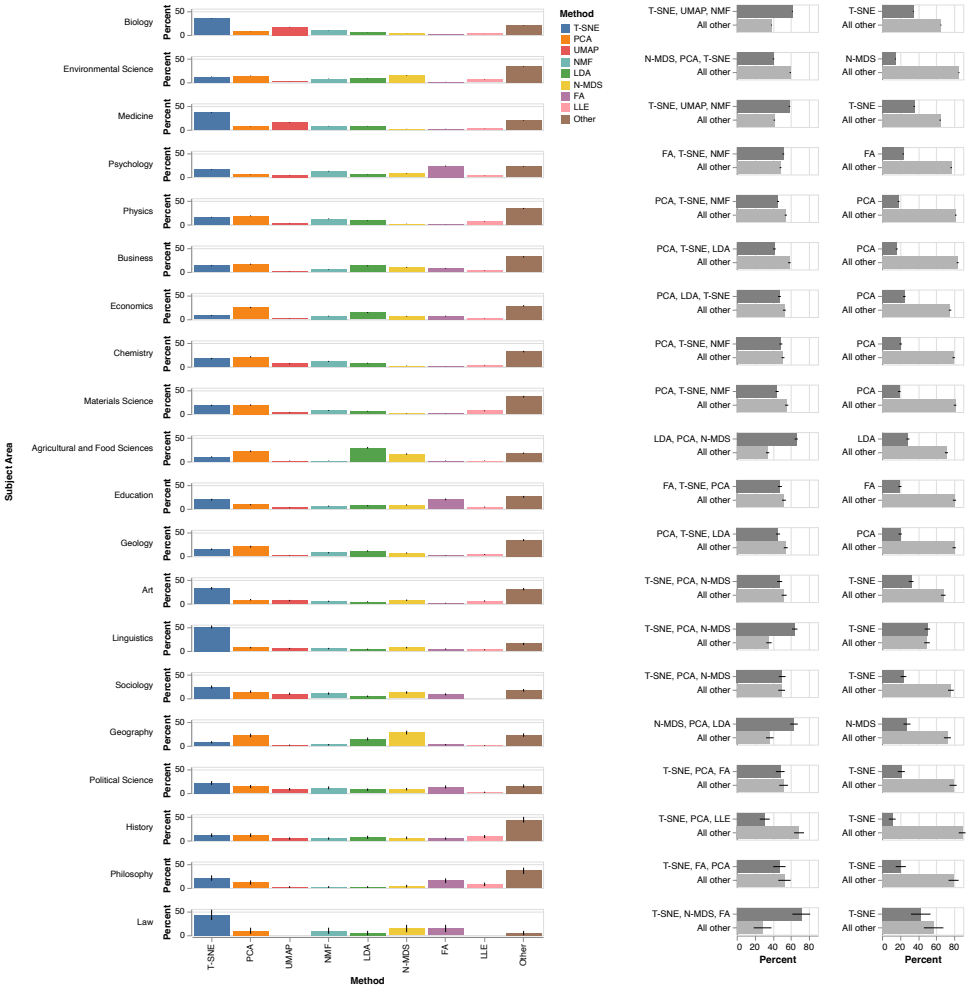

Figure C.10: Percentage of papers citing dimensionality reduction methods, stratified by subject area (2013-2023). Left: Methods that are not ranked in the top three in at least one subject area are mapped to the 'Other' category. Middle: Methods that are not ranked in the top three within the subject area are mapped to the 'All other' category. Right: All methods except the top method within the subject area are mapped to the 'All other' category. Full method names can be found in Table C.1.

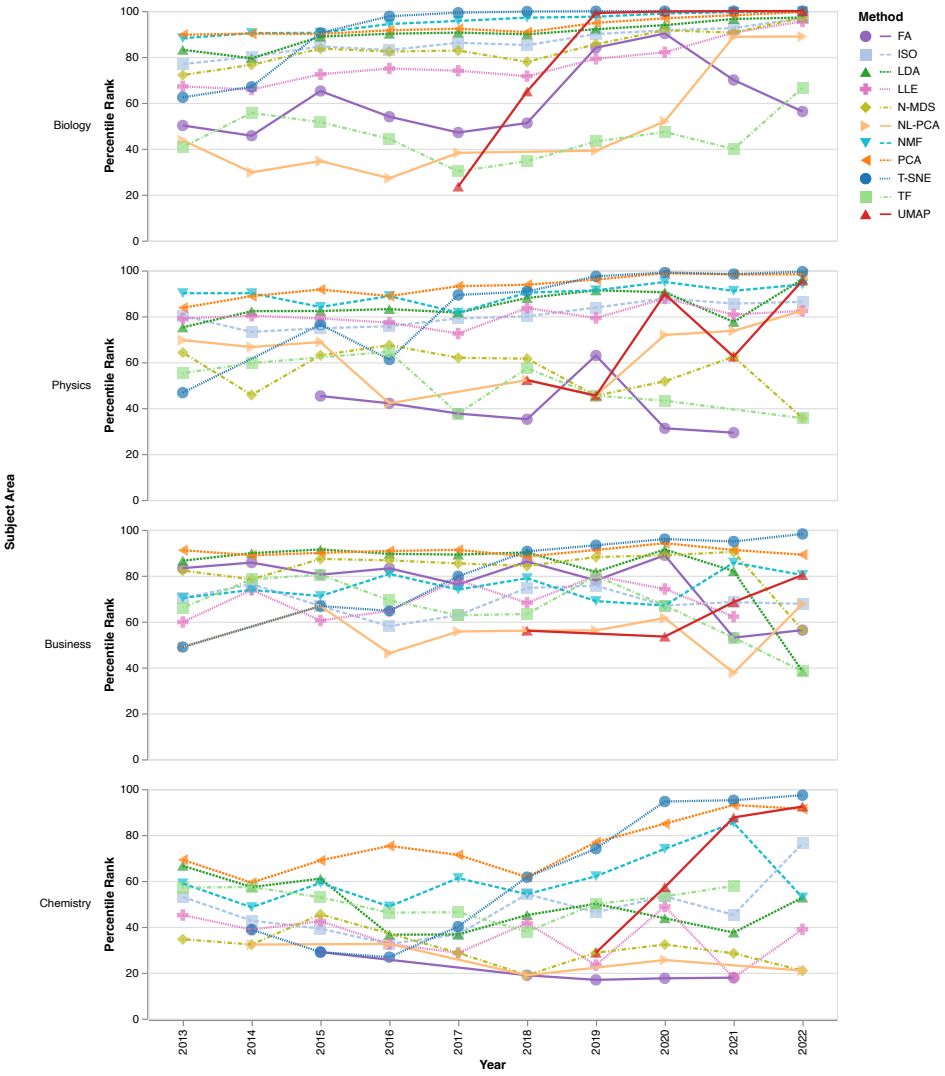

Figure C.11: Percentile rankings of dimensionality reduction methods over time based on citation counts in each subject area and year. Includes only Biology, Physics, Business, and Chemistry subject areas. Expanded figure including all subject areas appears in Supplement C.12. Methods are included if they are ranked in the top five for at least one subject area and year. Full method names can be found in Table C.1.

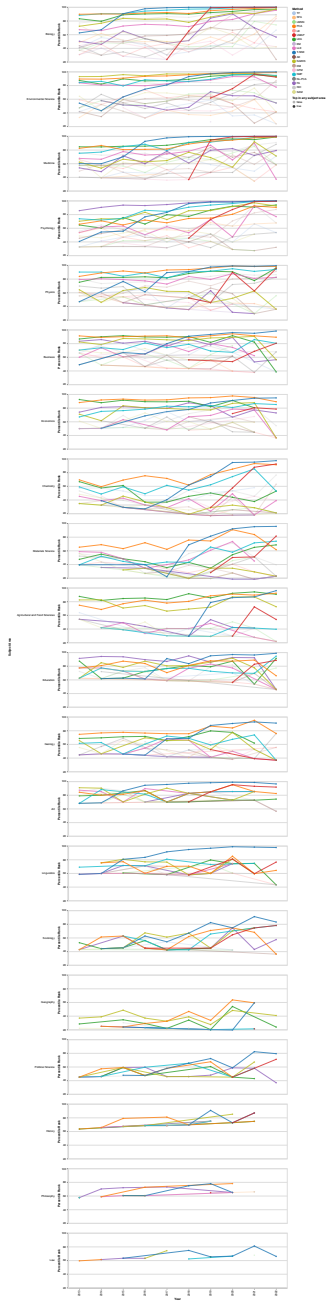

Figure C.12: Percentile rankings of dimensionality reduction methods over time based on citation counts in each subject area and year. Methods are included if they are ranked in the top three for at least one subject area and year. Opacity is used to highlight methods which are ranked in the top three for at least one subject area over the entire time period. Full method names can be found in Table C.1.

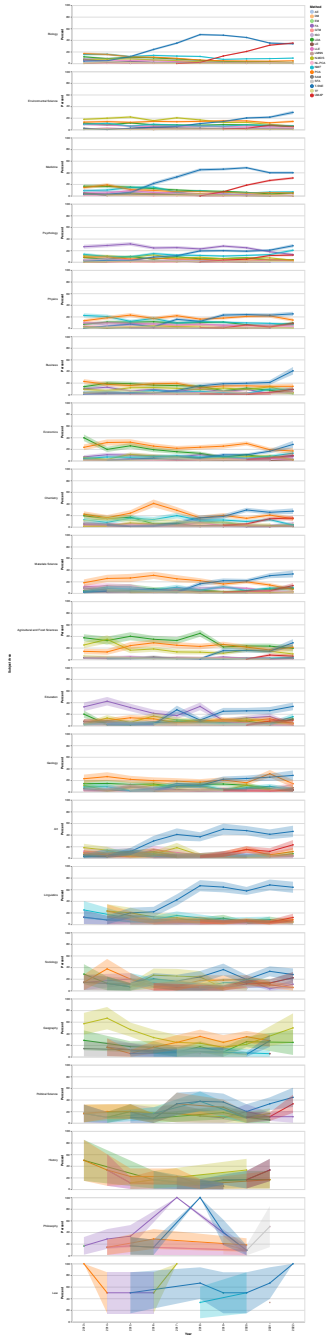

Figure C.13: Proportional usage of dimensionality reduction methods over time based on citation counts in each subject area and year. Methods are included if they are ranked in the top three for at least one subject area and year. Full method names can be found in Table C.1.

Acronym	Name	Reference
AE	Autoencoder	205
CCA	Canonical correlation analysis	234
CHL	Chalmers	106
CLM	ClassiMap	262
CuCA	Curvilinear component analysis	133
DM	Diffusion maps	124
EM	Elastic maps	175
FA	Factor analysis	395
FD	Force-Directed	408
FMAP	FastMap	156
GDA	Generalized Discriminant Analysis	162
GPLVM	Gaussian process latent variable model	255
GTM	Generative topographic modeling	79
ICA	Independent component analysis	221
F-ICA	FastICA	211
NL-ICA	Nonlinear ICA	224
IDMAP	Interactive Document Map	307
ISO	Isomap	409
L-ISO	Landmark Isomap	117
KECA	Kernel entropy component analysis	216
KLP	Kelp	68
LAMP	Local Affine Multidimensional Projection	218
LDA	Linear discriminant analysis	159
LE	Laplacian eigenmaps	74
LLC	Locally linear coordination	407
LLE	Locally linear embedding	370
H-LLE	Hessian LLE	142
M-LLE	Modified LLE	469
LMNN	Large-margin nearest neighbor metric	439
LoCH	Local convex hull	155
LPP	Locality preserving projection	195
LSP	Least square projection	342
LTSA	Local tangent space alignment	470
L-LTSA	Linear local tangent space alignment	468
MC	Manifold charting	91
MCA	Multiple correspondence analysis	75
MCML	Maximally collapsing metric learning	172
MDS	Metric Multidimensional scaling	126
L-MDS	Landmark MDS	387
MG-MDS	Multi-Grid MDS	96
N-MDS	Nonmetric MDS (Kruskal)	244
MVU	Maximum variance unfolding	440
F-MVU	Fast MVU	285
L-MVU	Landmark MVU	441
NeRV	Neighborhood retrieval visualizer	424
NMF	Nonnegative Matrix Factorization	256
PBC	Projection by clustering	342
PC	Principal curves	192
PCA	Principal component analysis	345
I-PCA	Incremental PCA	369
K-PCA-S	Kernel PCA (Sigmoid)	384
L-PCA	Localized PCA	160
NL-PCA	Nonlinear PCA	243
P-PCA	Probabilistic PCA	412
R-PCA	Robust PCA	225
S-PCA	Sparse PCA	478
PLMP	Part-Linear multidimensional projection	343
PLP	Piecewise laplacian-based projection	344
PP	Projection pursuit	209
RBF-MP	Multidimensional projection with Radial Basis Function	59

Table C.1: Dimensionality reduction methods considered in Section C.3, adapted from Table 1 of Espadoto *et al.*[151]. Continued in Table C.2.

Acronym	Name	Reference
RP	Random projections	360
G-RP	Gaussian random projection	131
S-RP	Sparse random projection	131
SAM	Sammon mapping	375
R-SAM	Rapid sammon (Pekalska)	347
SDR	Sufficient dimensionality reduction	239
SFA	Slow feature analysis	447
SMA	Smacof	259
SNE	Stochastic neighborhood embedding	204
T-SNE	t-distributed stochastic neighborhood embedding	418
SOM	Self-organizing maps	360
ViSOM	Visualization-induced SOM	458
SPE	Stochastic proximity embedding	49
G-SVD	Generalized SVD	333
T-SVD	Truncated SVD	184
TF	Tensor factorization	189
UMAP	Uniform manifold approximation and projection	300
VQ	Vector quantization	222

Table C.2: Table C.1, continued.

Acronym	Name	Reason
DML	Distance metric learning	Unable to identify reference
FS	Feature selection	Too general
LR	Linear regression	Too general
MAF	Maximum autocorrelation factors	Unable to identify reference
ML	Manifold learning	Too general
t-NeRV	t-NeRV	Unable to identify reference
NLM	Nonlinear mapping	Unable to identify reference
NN	Neural networks	Too general
K-PCA-P	Kernel PCA (Polynomial)	Unable to identify reference
K-PCA-R	Kernel PCA (RBF)	Unable to identify reference
PLSP	Piecewise least square projection	Unable to identify reference
PM	Principal manifolds	Unable to identify reference

Table C.3: Dimensionality reduction methods found in Table 1 of Espadoto *et al.*[151] that were not considered in Section C.3.

D

Zarr: Multidimensional Array Storage Format

D.1 Preface

The increasing size of datasets is a challenge for not only the single-cell domain but also other fields. Data size challenges are being felt by analysts of earth-scale satel-

lite imagery datasets and high-resolution microscopy datasets[313]. A common practice is to upload these datasets to cloud object storage systems. However, traditional file formats such as HDF5 can be difficult to read partially, which poses an obstacle for analysis performed from remote compute resources, as well as for web-based visualization that is limited by browser memory caps.

D.2 Zarr

Zarr is a storage format for large array data that is designed to support partial reads and writes on both local and remote file systems[306]. Zarr is conceptually similar to HDF5 in that a Zarr store can contain arrays, groups (*i.e.*, folder-like containers for multiple arrays), and JSON metadata. However, Zarr enables large arrays to be split into multiple small files that can be read and written in parallel. The Zarr specification is programming language-agnostic and relatively simple, which has enabled developers to implement read/write libraries for multiple languages including C, C++, Java, JavaScript, Julia, Python, and Rust[46].

D.3 Zarr in R

Due to the widespread use of R in bioinformatics, the lack of an established Zarr implementation for R poses a barrier to adoption of Zarr-based formats by the bioinformatics community. One approach to address this challenge is to develop R bindings to lower-level C/C++ libraries, which is the approach taken by the Rarr project[31] which supports reading 14 data types, writing 4 data types, and 5 compression methods. As a learning exercise and experiment, I ported the most popular Zarr implementation (Zarr-Python) to an R package called Pizzarr. Pizzarr currently supports

both reading and writing of 15 data types and 6 compression methods. While the pure R implementation of Pizzarr is likely slower than Rarr's C bindings, the Pizzarr API is more fully-featured and will be more familiar to users of Zarr-Python and the code more readable and maintainable[28].

E

Supplementary Information for Chapter 1

E.1 Preface

This chapter contains supplementary information for Chapter 1.

E.2 Other single-cell topics

While not directly relevant to the research described in this thesis, we highlight these topics for readers as important research areas which could benefit from future efforts to apply interactive visualization techniques.

E.2.1 Immune Repertoire Sequencing Methods

The immune system is key to many biological processes, including inflammation, aging/senesence, and response to pathogens, allergens, and immunotherapy, and vaccines. B cells and T cells are two immune cell types which have receptors that recognize foreign molecules (antigens). The diversity of antigens that can be recognized by B and T cell receptors (BCR and TCR) is controlled by the process of V(D)J recombination whereby different gene segments in variable exons of the immunoglobulin (Ig) and receptor genes are randomly assembled. As BCR and TCR genes are highly expressed, the BCR and TCR sequences can be profiled using full-length scRNA-seq techniques (as opposed to techniques which sequence transcripts only partially) and jointly analyzed with the rest of the transcriptome[338,84].

E.2.2 Perturbation Screening Methods

Perturbation screening methods use single-cell profiling techniques following CRISPR-based genetic perturbations of cells (*i.e.*, using CRISPR to knock out, knock down, or activate genes). Perturb-seq is a specific experimental technique in which a pooled CRISPR screen is performed prior to single-cell RNA sequencing[138]. In such a pooled screen, thousands of single-guide RNA (sgRNA) constructs target different genes for perturbation, within a single experiment. Importantly, in Perturb-seq, the vector en-

coding the sgRNA also contains an expressed guide barcode, enabling scRNA-seq readouts which identify not only the cell identity (as in standard scRNA-seq) but also the identity of the guide RNA. Recently, Ota *et al.*[330] have developed a method which links Perturb-seq experimental data with loss-of-function (LoF) information to model the effects of LoF on gene expression. The Perturb-seq approach enables scalable and cost-effective profiling of the transcriptome-wide effects of thousands of perturbations.

E.2.3 Integration of Genetics and Molecular Data

Expression quantitative trait locus (eQTL) analysis using single-cell RNA sequencing (scRNA-seq) data aims to identify genetic variants associated with gene expression variability at the single-cell level. This type of analysis requires expression profiles with matched genotype data (*i.e.*, from the same individuals). Similar analyses can be conducted with data from alternative experimental modalities, including chromatin accessibility data (to identify chromatin accessibility QTLs), as well as with cell-state abundance data (csaQTLs)[373]. Single-cell data can be powerful in such analyses, as many studies have found allelic effects on gene expression that are cell type-specific[456]. Through colocalization analysis, researchers can identify genetic variants that are likely to be causal for gene expression changes (eQTLs) using genome-wide association study (GWAS) data.

F

Supplementary Information for Chapter 2

F.1 Preface

This chapter contains supplementary information for Chapter 2. Portions of this chapter have been adapted from the supplementary information associated with the

corresponding manuscript:

- The State of Single-Cell Atlas Data Visualization in the Biological Literature[229]

F.2 Query Strings

We provide the advanced query strings used to systematically identify publications from the Web of Science literature database.

```
(PY=(2020) OR PY=(2021) OR PY=(2022)
OR PY=(2023) OR PY=(2024))
AND TI=(atlas) AND TI=(human)
AND (TI=(cell) OR TI=(cells)
OR TI=(cellular))
NOT (TI=(tumor) OR TI=(cancer)
OR TI=(carcinoma) OR TI=(mouse)
OR TI=(murine))
```

Listing 9: Web of Science advanced query for single-cell atlas publications. Last queried August 12, 2024.

F.3 Paper Filtering and Annotation Processes

After querying Web of Science, we exported all search results to Bibtex format. We imported the Bibtex file from Web of Science into a new Zotero collection called "Filtering in progress". We used Zotero's "Find Full Text" option to download all paper PDFs. We created a Zotero collection called "Pass filtering" and collections for failures of each of the eligibility criteria (*e.g.*, "Fail is_biology"). We iterated over all of the papers in the "Filtering in progress" collection, moving them to either the "Pass filtering" collection or one of the failure collections, until the in-progress collection was empty.

Once the set of eligible papers was established (the "Pass filtering" collection), we saved the paper PDF files into a folder (using Zotero's "Show in Finder" option) and exported metadata for all passing entries into a tabular file using the Zotero "Export Items" option with "Format: CSV".

We next imported the folder of paper PDF files into a new project in the AtlasTI software. Within each AtlasTI document, we annotated each subfigure using rectangle area quotations (as opposed to text-based quotations). We applied one or more Codes to each quotation. Where hierarchies of codes emerged, we applied the most-specific (*i.e.*, leaf) code, and subsequently created Smart Codes with OR conditions to automatically apply the corresponding more-general codes.

To facilitate analysis and communication of our results, we exported the AtlasTI project to the QDPX format. We transformed the QDPX output to JSON, CSV, and PNG files using a Python script[30]. We created a figure browser website using the Astro static website development framework[34].

F.4 Code Descriptions

We provide descriptions of the codes used to analyze static atlas publication figures.

- **anatomical diagram**: Schematic or illustration of an organism, organ, or functional tissue unit intended to convey anatomical information.
- **area plot**: Area plots encode data using area visual marks.
- **bar plot**: Bar plot (non-stacked), including small multiples of bar plots.
- **beeswarm plot**: A plot which displays a distribution of values using points that are arranged to prevent overlap.

- **box plot**: Box plot, including small multiples of box plots.
- **cell cycle**: A plot which displays predicted cell cycle phase information for each cell.
- **cell type composition**: Plot displaying one or more categorical distributions of cell types, typically also intended to compare proportions of cell types among samples or conditions.
- **chromatin accessibility**: Plots which display results from chromatin accessibility (scATAC-seq) experiments.
- **clustermap**: A heatmap where the horizontal or vertical (or both) axis is aligned to the leaves of a dendrogram.
- **comparison**: Plots in which two or more conditions are being compared, for example via small multiples or directly with a volcano plot or developmental time-series line plot.
- **confusion matrix**: A plot which displays classification model performance using a heatmap in which recall values for each category are encoded using a quantitative colormap.
- **contact matrix**: A type of heatmap in which both axes represent genomic coordinates, and values are colored according to chromatin contacts among each pair of genomic coordinates.
- **cross-organ**: A plot which displays data from multiple organs simultaneously.

- **cross-species**: Plot that includes data from multiple species, intended to facilitate comparison across species. In the context of this report, these plots include both human and non-human data.
- **custom annotations**: Any non-standard annotation to a plot. Includes text and region annotations on scatterplots and images, metadata bar plots alongside heatmaps, or arrows indicating data points of interest.
- **custom coloring**: Plots whose colormap is neither a direct categorical colormap (*e.g.*, for cell types) nor a direct mapping of biomarker abundance (*e.g.*, gene expression) values to a quantitative colormap.
- **data integration**: A plot which displays the results of a data integration algorithm, typically using a dimensionality reduction scatterplot.
- **deconvolution**: A plot which displays the results of cell type deconvolution algorithm, typically to display predicted proportions of cell types that compose the spots of a 10X Visium experiment.
- **dendrogram**: A standalone dendrogram or dendrogram as a component of a complex plot such as a clustermap. Typically used to display hierarchical clustering results or phylogenetic relationships.
- **density plot**: A density plot corresponds to a plot which uses a quantitative colormap to encode point density, and excludes gating scatterplots.
- **developmental**: Plots which contain multiple developmental stages, typically intended to facilitate comparison among stages.

- **diagram**: A schematic intended to illustrate a concept, which does not necessarily encode quantitative information.
- **dimensionality reduction**: Visualization of a low-dimensional embedding of high-dimensional data, typically using a scatterplot.
- **donut plot**: A type of pie chart which lacks an inner circle.
- **dot plot**: Plot in which the radius and color channels of circle marks arranged in a grid are used to encode two quantitative values along two categorical dimensions, typically gene and cell type, which define the horizontal and vertical axes of the grid.
- **epigenomics**: A plot which displays epigenomic data, such as chromatin accessibility, transcription factor binding, or a chromatin contact matrix.
- **evolution**: A plot which encodes evolutionary information, for example using a dendrogram displaying phylogenetic relationships.
- **experimental design**: Figure outlining aspects of experimental design, typically using a combination of flow charts, schematics, diagrams, and illustrations. Sometimes these include quantitative information, but not always.
- **filtering**: A subtype of scatterplot in which a subset of data points are displayed.
- **FISH-like**: A type of spatial plot in which point visual marks correspond to RNA transcript locations. Often these points are overlaid on an image and colored according to the gene corresponding to the transcript.

- **gating scatterplot**: A style of scatterplot in which the X and Y axes correspond to two different biomarkers, intended to facilitate identification of specific subpopulations of cells. Points are often colored to indicate point density, potentially with density contours overlaid.
- **gel**: An image of a gel from a gel electrophoresis experiment.
- **genetics**: A plot which contains GWAS results, eQTL results, genetic risk scores, or allele-specific expression values.
- **genomic**: Plots which contain genome-mapped data, often with genomic coordinates mapped to the X-axis of the plot.
- **gsea**: Plots that display the results of gene set enrichment analysis, using any plot type.
- **heatmap**: Rect visual mark used to encode values within a 2D matrix, with matrix values colored using a quantitative colormap.
- **hexbin**: A type of density plot which encodes the density of data points along two dimensions using hexagonally-shaped bins, with density encoded using a quantitative colormap.
- **highlighting**: A subtype of scatterplot in which a subset of data points are highlighted, typically with these points colored in the foreground and the rest of the points displayed in the backround in gray or less opaque.
- **histology**: A microscopy image of a chemically-stained sample. These are RGB images as opposed to multiplexed/multi-channel images.

- **imaging**: A type of spatial plot in which pixels of an image are displayed. This includes RGB images (*e.g.*, histology), grayscale images, and multiplexed images.
- **immune repertoire**: Plots which encode information from immune repertoire experiments, such as TCR or BCR sequencing.
- **jitter plot**: A plot displaying a distribution of values, with points jittered to convey point density, typically layered with violin or box plots.
- **kaplan-meier**: A Kaplan-Meier plot displays the probability of survival over time, typically used to compare survival rates among conditions or groups.
- **ligand-receptor**: Plots that display ligand-receptor or cell-cell communication information, using any plot type.
- **ligand-receptor circle plot**: A type of node-link diagram in which nodes correspond to cell types and are arranged in a circle, with edges corresponding to ligand-receptor or cell-cell interactions.
- **line plot**: A plot which uses a line visual mark.
- **logo plot**: Sequence logo plots display both nucleotide frequency and information content for nucleotide positions of a binding motif.
- **lollipop chart**: Similar to a bar chart, except bars are drawn using a line that ends with a circle visual mark.
- **manhattan plot**: A type of scatterplot in which the X-axis encodes genomic coordinates and the Y-axis encodes the negative of the log10 of the P-value.

- **model performance**: Plots which encode a model performance metric, such as precision, recall, or F1 score.
- **multi-omics**: A plot which simultaneously displays information from more than one experimental modality, such as both scRNA-seq and scATAC-seq.
- **multi-scale**: Spatial visualization in which multiple physical scales are displayed simultaneously, typically using small multiples or insets, with scale conveyed via scale bars.
- **multiplex**: Spatial visualization displaying multiplexed (*i.e.*, multi-channel) imaging data, where each channel of the image corresponds to a color (*i.e.*, each channel on its own would correspond to a grayscale image, as opposed to RGB).
- **network**: A node-link diagram used to display a network.
- **non-standard metric**: A plot whose colormap or Y-axis encodes a non-standard metric, such as the output of a custom scoring function
- **non-standard plot type**: Plot types that either do not correspond to a single-cell plotting toolkit function or use different semantics than how the same plot type is often used in single-cell figures.
- **parallel coordinates**: A type of line plot in which lines are drawn between corresponding points positioned along two or more quantitative axes that are arranged in parallel.
- **pie chart**: Either a standalone pie chart or small multiples of pie charts, including scatterplots where points contain pie charts, and node-link diagrams in which nodes contain pie charts.

- **pie chart visium**: A type of deconvolution plot in which the predicted cell type proportion for each spot of a 10X Visium experiment is displayed using a per-spot pie chart, typically overlaid on the histology image.
- **ppi network**: A type of network plot in which nodes correspond to proteins, and edges correspond to protein-protein interactions (PPIs).
- **proteomics**: A plot which displays results from a proteomics experiment.
- **pseudoaxis**: A plot which organizes data points along a pseudoaxis to convey a predicted spatial ordering.
- **pseudotime**: Plots which encode pseudotime, typically along the X-axis or using a quantitative colormap.
- **radial plot**: A plot which encodes points using the radius and angle along a polar coordinate system.
- **representative sample**: Spatial visualization intended to display a representative example of some phenomena of interest.
- **sankey plot**: A sankey diagram which displays degrees of flow among states.
- **scatterplot**: Point or circle visual mark used to encode two (typically quantitative) variables on the X and Y axes.
- **signature score**: A plot which encodes the output of a scoring function that combines the expression or abundance of multiple biomarkers.
- **small multiples**: Repeated instances of the same plot type, either within the same subfigure or among subfigure instances within a parent figure.

- **spatial**: Plot in which the X and Y axes correspond to physical space, such as imaging or 10x Visium.
- **spatial transcriptomics**: A type of spatial plot which displays the results of a spatial transcriptomics experiment.
- **spatial validation**: A type of spatial plot which is intended to convey validation of some finding.
- **spatial-3d**: A type of spatial plot which displays a 2D projection of a 3D image/volume.
- **stacked bar plot**: Stacked bar plot, including small multiples of stacked bar plots.
- **statistical significance**: Plots which contain annotations that indicate statistical significance, or which directly encode statistical significance (*e.g.*, with a transformed p-value axis).
- **subclustering**: A subtype of dimensionality reduction scatterplot in which a subset of data points are displayed, after applying the algorithm to the subset of points (as opposed to using the original embedding coordinates resulting from the algorithm applied to the full set of points).
- **subsubfigures**: Subfigures which contain more than one plot, and which are also not small multiples of a single plot type.
- **ternary plot**: A plot which encodes the ratios of three variables using coordinates within an equilateral triangle. Also called a simplex plot.
- **upset plot**: An UpSet plot visualizes multiple intersecting sets.

- **velocity**: Plots which encode RNA velocity algorithm results, typically with a vector field-like grid of arrows overlaid on a dimensionality reduction scatterplot.
- **venn diagram**: A plot which uses overlapping circles to convey the degree of similarity and difference among multiple intersecting sets of items.
- **violin plot matrix**: Small multiples of violin plots in which the violin plots are arranged in a grid.
- **violin plots**: Violin plot, including small multiples of violin plots.
- **visium**: Spatial visualization that displays a grid of 10X Visium spots, typically using circle visual marks, often overlaid on the corresponding histology image.
- **volcano plot**: A style of scatterplot in which points represent biomarkers, the X-axis encodes $\log_2$ fold-change, and Y-axis encodes the negative of the $\log_{10}$ p-value for the biomarker, typically displaying the results of a differential expression test.

G

Supplementary Information for Chapter 3

G.1 Preface

This chapter contains supplementary information for Chapter 3. Portions of this chapter have been adapted from the supplementary information associated with the

corresponding manuscript:

- Vitessce: integrative visualization of multimodal and spatially resolved single-cell data[228]

G.2 Extended Data Figures for Vitessce

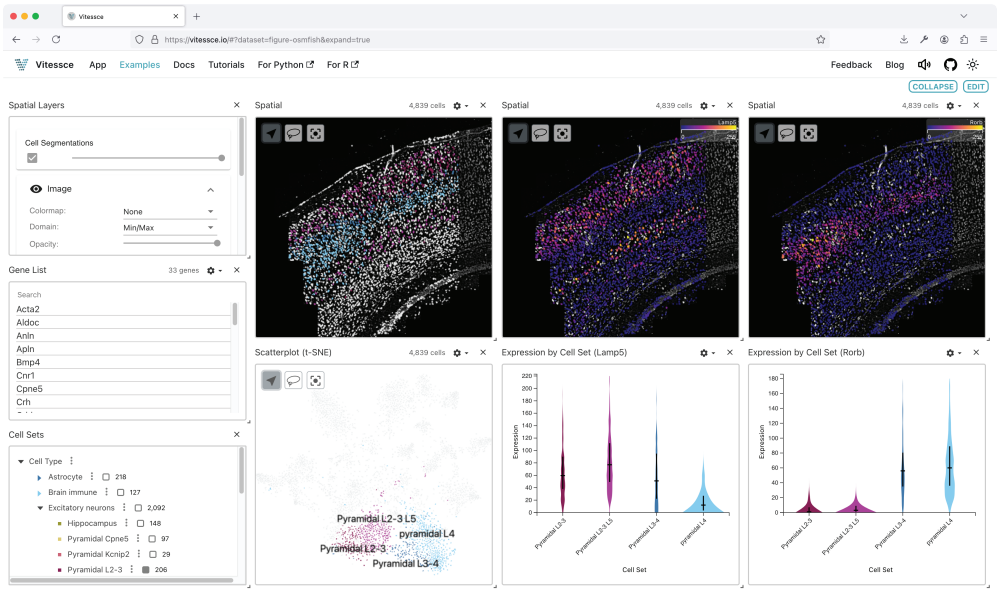

Figure G.1: Visualization of single-molecule fluorescence in situ hybridization (smFISH) data. Codeluppi *et al.*[122] used single-molecule fluorescence in situ hybridization (smFISH) to profile the somatosensory cortex in a mouse brain section. The authors selected 33 marker genes based on previous scRNA-seq findings in the somatosensory cortex and their ability to define cell types. A notable finding from this experiment was the discovery of a transition region defined by the Pyramidal L3/4 excitatory neuron cell type. Using the spatial plot and heatmap in Vitessce, we can reproduce this finding and observe the reported joint expression of markers *Lamp5* and *Rorb* that define the surrounding Pyramidal L2/3 and L4 cell types, respectively.

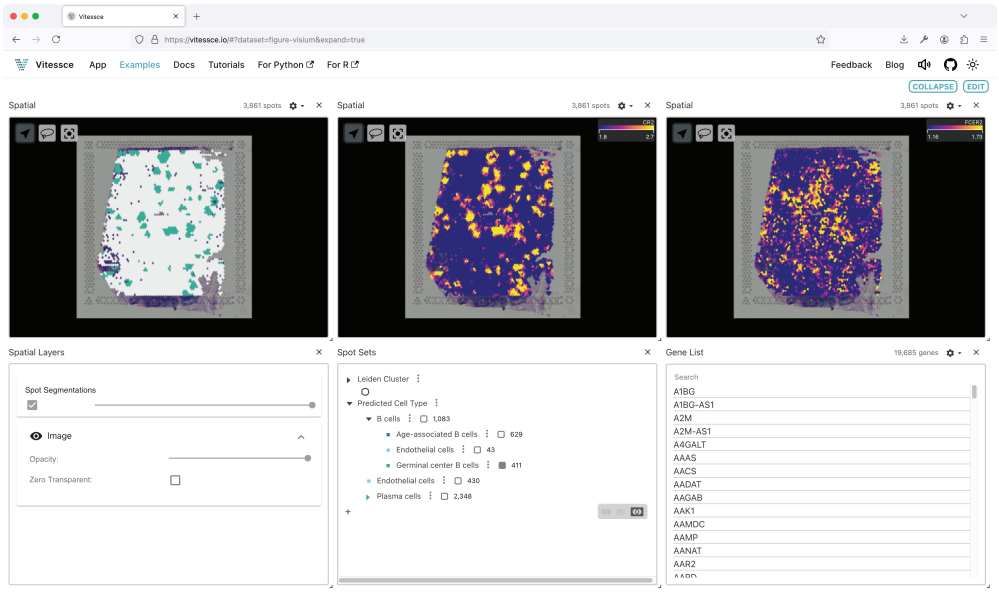

Figure G.2: Visualization of a 10x Genomics Visium dataset. 10x Genomics provides this dataset as a demo of the Visium technology and thus this dataset does not answer a particular biological question. Nonetheless, we can validate that the expected lymph node cell types are present. Using the CellTypist method for semi-automated cell type annotation of immune cells, we identify that *CR2* is highly expressed by spots predicted to contain Germinal center B cells[141]. Using CellPhoneDB[145], we query for known ligands of this receptor, which results in identification of *FCER2*[63]. Using linked spatial views in Vitessce, we can observe that *CR2* is specifically expressed by Germinal center B cell spots while the expression of *FCER2* is less specific but does overlap.

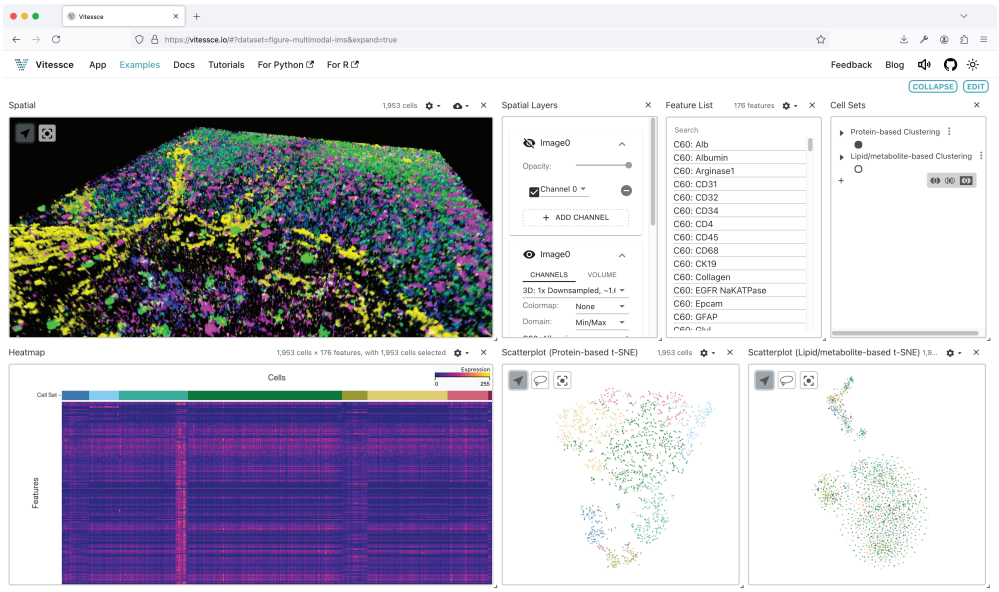

Figure G.3: Visualization of a 3D multimodal mass spectrometry imaging dataset. Tian *et al*.[411] report the development of a mass spectrometry imaging workflow that utilizes both cryogenic $(H_2O)_{n>28K}$-GCIB-SIMS and C_{60}-SIMS techniques to capture metabolites, lipids, and proteins at single-cell resolution in the same tissue section. Applying this workflow to a human liver sample, the authors find that they are able to classify metabolic zones and cell types using lipid and metabolite profiles. Using the volumetric rendering features available in Vitessce, we can explore the 3D multi-channel imaging data alongside the cell-by-feature heatmap and both protein-based and lipid/metabolite-based dimensionality reductions.

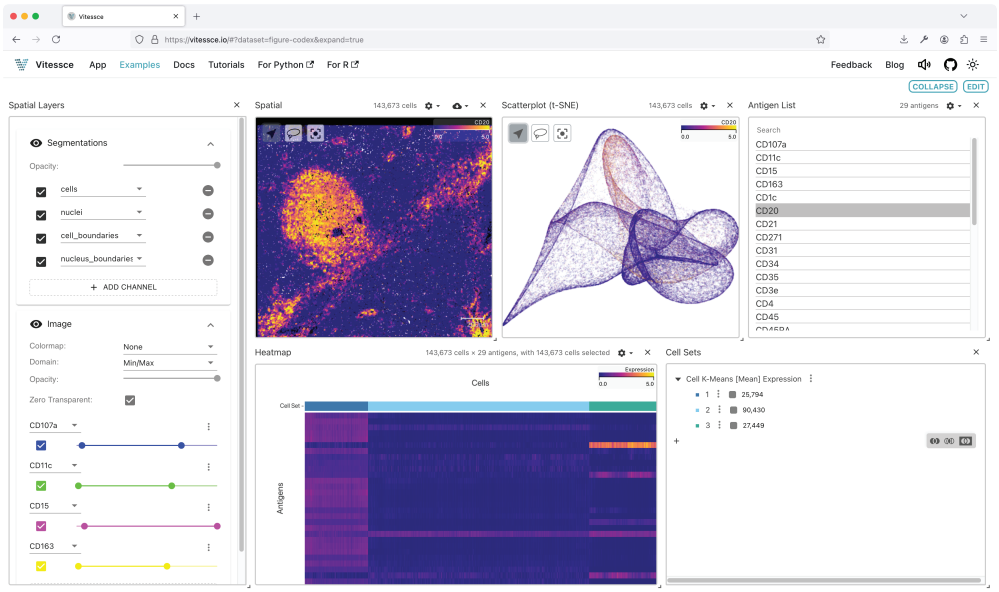

Figure G.4: Visualization of a CODEX dataset. CO-Detection by indEXing (CODEX) is a spatial assay which uses oligonucleotide probes conjugated to antibodies specific to a set of antigens of interest. Through multiple rounds of hybridization, up to 50 different antigens may be imaged. We present an example of using Vitessce to explore a human spleen tissue sample in which 29 antigens were targeted. The dataset contains the 29-channel microscopy images, cell segmentations, cell-by-antigen quantification, and unsupervised clustering results. To interact with this dataset, we configure Vitessce with a spatial plot, heatmap, and controllers to select image channels, antigens, and cell clusters of interest. With multiple representations of the same data, we can choose to begin exploration by focusing on one of several entity types. Because this dataset lacks cell type annotations, we are interested in the protein markers which define each cluster. We approach the visual analysis by selecting a cluster in the cell set controller, then searching for cluster-specific patterns in the spatial view. With clustering assignments encoded as cell segmentation mask colors, it appears that cells assigned to cluster 6 within the 'Cell K-Means [Mean-All-SubRegions] Expression' localize to a small set of compact regions in the spatial view, suggestive of cellular neighborhoods in the spleen. When we identify cluster 6 in the heatmap view, it is clear that the marker with the highest relative expression in this cluster corresponds to the B cell marker CD20. We can confirm that CD20 is enriched in this spatial region by hiding the cell segmentation masks in the spatial plot, uncovering the underlying image data. Using the spatial layer controller, we can select the CD20 channel and verify that the fluorescence signal indeed appears in the expected regions. The heatmap and image views can help us to find other markers correlated with the CD20 spatial expression pattern.

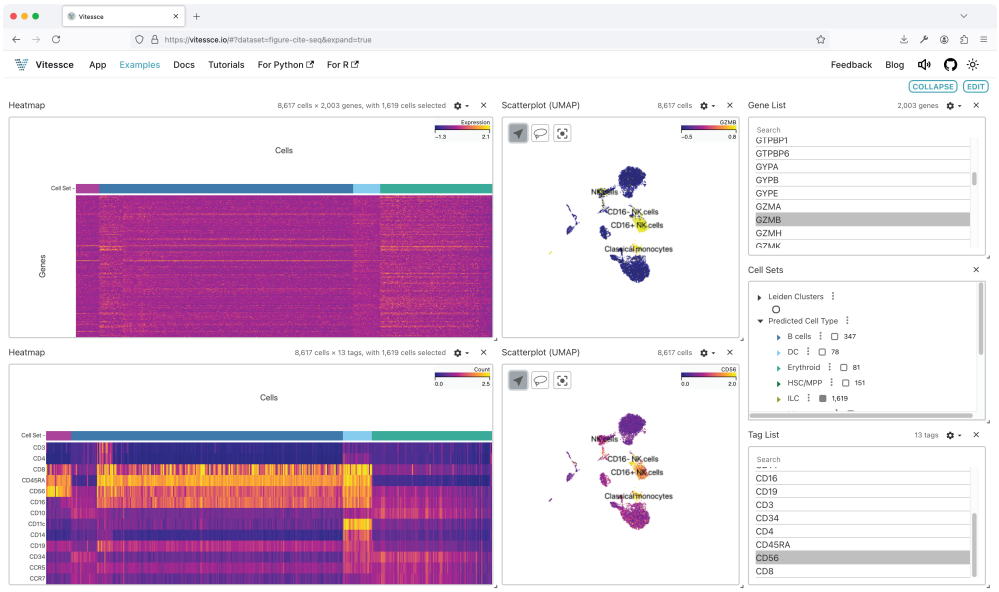

Figure G.5: Visualization of a CITE-seq dataset. Stoeckius *et al*. report the development of the CITE-seq method for measuring gene expression and surface protein abundance in the same cells[398]. The authors validate their technique on a sample of cord blood mononuclear cells (CBMCs) by measuring the abundance of well-characterized immune cell type markers. Using linked scatterplots and heatmaps to visualize protein abundance and gene expression levels simultaneously, we can reproduce the authors' multimodal characterization of the Natural Killer (NK) cell type based on CD56 levels and the expression of genes *GZMB*, *GZMK*, and *PRF1*.

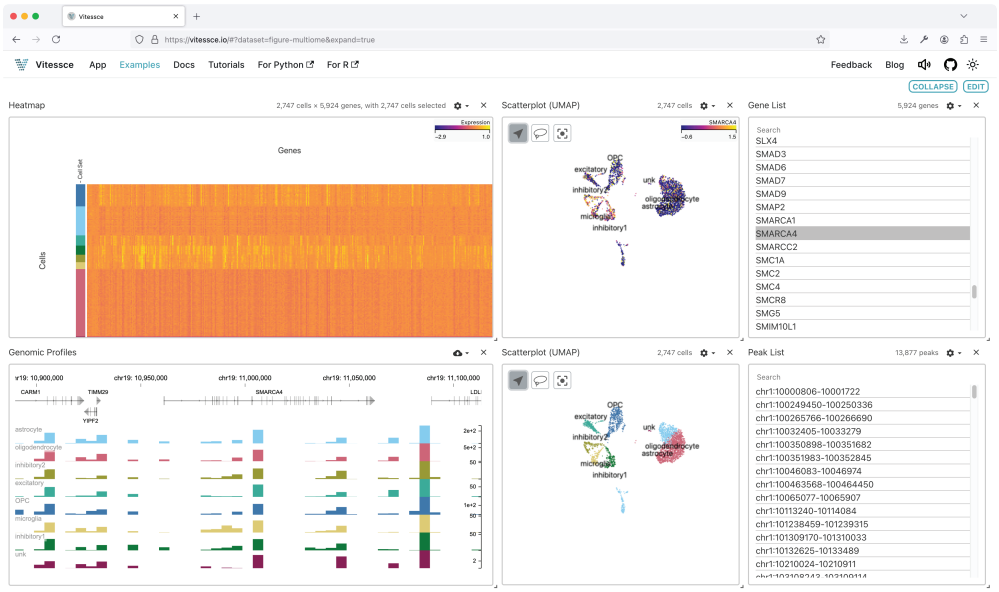

Figure G.6: Visualization of a 10x Genomics Multiome dataset. This dataset is provided by 10x Genomics as a demo of the Multiome technology and thus is not intended to answer a particular biological question. Nonetheless, we can use this dataset to validate that the expected cerebellum cell types are present. Using the heatmap, we identify the gene *SYT1* based on its expression pattern in the cell cluster corresponding to the microglia cell type. Querying for transcription factors of *SYT1* using Cistrome Toolkit[271,473], we can identify *SMARCA4* as the top result from a human neuron sample, with a regulatory potential score of 0.50982. Navigating to this gene on chromosome 19 in the genome browser view, we can observe a footprint-like pattern in the track for microglia, which we might want to validate in follow-up analyses.

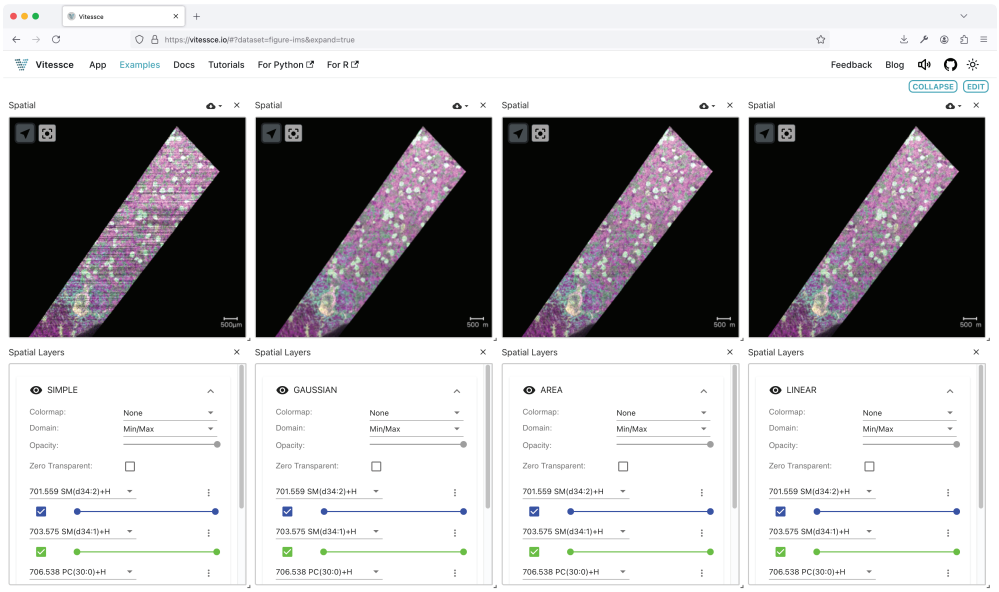

Figure G.7: Visualization of a MALDI IMS dataset processed multiple ways. Vitessce can be used for visual comparison of the outputs of multiple data processing methods within datasets containing multiple technical conditions, supporting the workflows of algorithm developers and computational biologists. For example, computational biologists tasked with analyzing imaging mass spectrometry (IMS) data may be interested in the consequences of using different interpolation methods for generation of pyramidal image files. Pyramidal image files contain downsampled lower resolution versions of the same image which are generated by pre-aggregating pixels, facilitating multi-scale visualization in a web browser. The choice of a downsampling pre-aggregation interpolation function may affect the visual properties of lower resolution images. The open microscopy image processing toolkit bioformats2raw implements several common interpolation methods: SIMPLE, GAUSSIAN, AREA, LINEAR, CUBIC, and LANCZOS. We use the multiscale image rendering capabilities of Vitessce to compare the application of these methods to an IMS dataset. The zoom levels and centers of each spatial plot are coordinated to facilitate comparison of the same image region at multiple scales. Vitessce helps to discover that the SIMPLE interpolation method introduces regularly-spaced horizontal streaks into the lowest resolution version of the image. Zooming in prompts Vitessce to load a higher resolution image in which the streaks are absent, indicating that they are artifacts of the downsampling procedure. This insight may steer us away from using the SIMPLE method when processing IMS data.

G.3 Using Vitessce

Vitessce is available in different forms to make it useful for multiple audiences. Researchers can use the R and Python packages to explore local or remote datasets dur-

ing data analysis. The Vitessce website and online configuration editor are designed for sharing visualizations with collaborators and debugging. Software developers can incorporate Vitessce into other tools and write plugins by using it as a JavaScript package.

G.3.1 Vitessce in Python environments and Jupyter Notebooks

Vitessce can be used as a Python package in scripts and Jupyter notebooks. Visualizations can be configured and rendered directly into notebooks using an interactive widget that is compatible with multiple notebook environments including JupyterLab, Jupyter Notebook (classic), and Google Colab. The implementation of the widget is based on Anywidget[289]. Installation instructions and API documentation can be found at https://python-docs.vitessce.io. A set of tutorial notebooks is available at https://github.com/vitessce/vitessce-python-tutorial.

G.3.2 Vitessce in R environments, RMarkdown, RStudio, and Shiny apps

Vitessce can be used as an R package in scripts, RMarkdown documents, and Shiny apps. Visualizations configured in R can be rendered using an interactive widget in the RStudio Viewer pane, RMarkdown documents, pkgdown websites, and Shiny apps. The widget is implemented as an R htmlwidget. Installation instructions and API documentation can be found at https://r-docs.vitessce.io.

G.3.3 Vitessce in JavaScript and web environments

Vitessce is implemented in JavaScript as a React component with corresponding APIs for configuration and registering plugins (described below). The JavaScript package can be used in websites, other React components, or other JavaScript packages. Installation instructions and API documentation can be found at http://vitessce.io.

G.3.4 Vitessce as a website and online configuration editor

To quickly configure a Vitessce visualization from a web browser, we provide a web application to write and edit Vitessce configurations using JSON and JavaScript syntax. This method of configuring a Vitessce instance does not require Python, R, or JavaScript package installation. This resource can be found at http://vitessce.io.

G.3.5 Python, R, and JavaScript configuration APIs

Vitessce configurations are defined using a JSON representation that specifies the view layout and points to local or remote data files via URLs. In addition to the declarative JSON representation, we have developed Python, R, and JavaScript APIs to enable users to define configurations programmatically. These configuration APIs support definition of datasets, files, views, the view layout, and view coordinations within the native object-oriented paradigms of each language.

G.3.6 JavaScript plugin API

The Vitessce JavaScript package contains functions for defining plugin view types, coordination types, data types, and file types. Once plugins have been defined, they

must be registered by providing a name that will be used to refer to the implementation in Vitessce configurations. Plugin view types must be implemented as React components. We provide examples and tutorials for using the plugin API on the Vitessce documentation website.

G.4 Multimodal configuration

Vitessce supports arbitrary multimodal datasets by adopting the observation-by-feature matrix conventions used by data analysis packages in the single-cell ecosystem including MultiAssayExperiment, Seurat, and MuData. In this model, observations are entities being measured, such as cells, molecules, spots, beads, or nuclei. Features are the characteristics being measured about entities such as genes, chromatin accessibility peaks, or surface proteins. Feature values are the quantities being measured, such as expression levels, counts, or intensities. Identifiers for types of observations, features, and feature values can be defined in the Vitessce configuration for both data and views. Vitessce then matches views to data accounting for observation type, feature type, and/or value type identifiers. For most views and data types, all three properties are used, but a subset or a superset may be used depending on which properties are relevant to a particular view. For example, the heatmap considers all three properties when loading data because the visualization contains features, observations, and values. A particular heatmap can be uniquely identified by these properties. In contrast, the feature list view is uniquely identified by only feature type and therefore observation type and value type are not considered when loading data for the view.

G.5 Data organization

In order to separate data loading from rendering and support multimodal experiments, Vitessce views load data corresponding to datasets and data types such as arrays of spatial coordinates per observation, dimensionality reduction coordinates per observation, images, and observation-by-feature matrices. Views may load data corresponding to one or more datasets and one or more data types. These data types may be aligned on certain axes (*e.g.*, to support shared observation or feature sets) or not (*e.g.*, to support comparison of multiple datasets).

Data types are loaded independently such that their data may be contained in the same file or split across independent files, allowing multiple file formats to be used to load each dataset. Vitessce defines multiple file types which correspond to data type-/file format pairs. If a file format supports multiple data types, a joint file type may be defined to simplify the configuration (*i.e.*, allowing URLs and file options to be defined once for a file while being exposed as multiple data types internally). For example, AnnData objects, which may contain multiple observation-by-feature matrices, dimensionality reductions, and spatial coordinates, can be configured using a higher-level joint file type that is specific to AnnData. However these specifics are abstracted from the implementations of views which simply perform lookups for data corresponding to individual data types.

G.5.1 On-demand loading of data subsets

To scale to large datasets, data loading is deferred in multiple cases: multi-scale images, multi-scale bitmask-based segmentations, genome-mapped data, and per-feature subsets of observation-by-feature matrices. The loading of highly-multiplexed multi-

scale (*i.e.*, pyramidal) image files is implemented using Viv. As described by Manz *et al.*[288], Viv loads images as data tiles corresponding to the current viewport zoom level (*i.e.*, resolution) and target position (*i.e.*, X/Y), as well as selections for channel (C), temporal (T), and Z axes. In Vitessce, this approach applies not only to primary images, but also to image bitmask files for cell and organelle segmentations, which can also be stored in multi-resolution formats.

Scalability to large observation-by-feature matrices in Vitessce takes advantage of a similar data tiling approach. These matrices can be stored in multiple Zarr-based file formats, in particular when using AnnData[427], MuData[94], and SpatialData[291]. Zarr supports tiled ("chunked") and compressed multi-dimensional arrays that can be served as directories of static files termed "stores." Benchmarking conducted by Moore *et al.*[311] demonstrates that accessing Zarr data is at least as fast as accessing the same data stored in HDF5 and TIFF, and in high-latency cloud storage scenarios Zarr outperforms HDF5 by an order of magnitude.

The Zarr chunk strategy can be configured to optimize for particular use cases. In Vitessce, performance is optimized when many observations and few features are stored in each chunk, enabling use cases such as quickly loading the expression values for a particular gene (*i.e.*, one feature) across all cells (*i.e.*, all observations). The tradeoff is that the same chunk strategy might result in poor performance under a different use case such as loading expression values for all genes for only one cell. We additionally compare the full sizes of Zarr and CSV-based AnnData objects in Figure G.8.

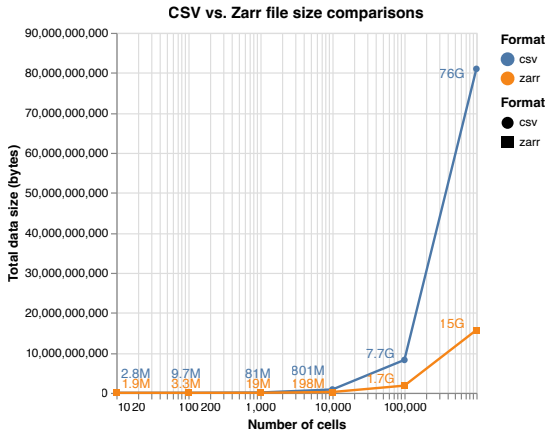

Figure G.8: Comparison of CSV- and Zarr-based file sizes. Each point corresponds to an AnnData object containing 17,811 genes and a variable number of cells, downsampled to between 10 and 1 million.

Genome browser tracks visualized using HiGlass also load data tiles on-demand based on the current browser viewport and zoom level, as described by Kerpedjiev *et al.*[231]. In Vitessce, we extend this mechanism to support genomic data from Zarr stores (in addition to the existing file formats supported by HiGlass) to eliminate the dependency on a specialized HiGlass Server. This extension is implemented as a HiGlass plugin data fetcher.

G.6 Coordinated multiple views

Vitessce adopts the coordinated multiple views technique from the field of information visualization to enable comparison tasks such as overview & detail, focus & context, difference views, and small multiples[361,382].

G.6.1 Coordination model

The coordination model proposed by Boukhelifa *et al.*[88] is used to link subsets of views on visualization, interaction, and data properties. In this model, views are not directly linked to one another, but instead to named property values referred to as coordination scopes. The properties that views can be linked on are referred to as coordination types. As a result, views are coordinated when they are linked to the same coordination scope for a given coordination type.

G.7 JavaScript implementation

Vitessce is implemented as a configurable React component in JavaScript. Below the root React component, visualization and control views are also implemented as React components. Views for visualization may use any web technologies including WebGL, SVG, CSS, and the HTML canvas element. The main Vitessce component is distributed in a JavaScript package which also exports an object-oriented configuration API and plugin APIs for defining custom views as React components, file types as JavaScript classes, and coordination types.

G.7.1 Implementation of coordinated multiple views

We have implemented the aforementioned coordination model in JavaScript and incorporated it into the Vitessce configuration schema using a JSON representation. As we anticipate that this may be useful for general implementation of coordinated multiple view visualizations using JavaScript, we provide the implementation as a standalone JavaScript package that can be used in conjunction with the plugin API.

G.7.2 Implementation of views

Vitessce configurations define a set of views that contain interactive visualizations. View implementations are independent of one another and use custom React hook functions to access values from the coordination model and load data. The views currently available include a scatterplot, spatial plot, heatmap, cell set size bar chart, gene expression histogram, gene expression per cell set violin plots, cell set manager, gene selection, image layer controller, and genomic profile per cell set genome browser. Views for visualization are implemented using custom JavaScript code and existing JavaScript libraries for statistical and geospatial data visualization. A list of the major open-source software packages used and a list of views currently available are provided in Supplementary Tables 1 and 2, respectively.

G.7.3 Implementation of custom DeckGL layers

The implementations of several views including the heatmap, scatterplot, and spatial views leverage the extensibility of the DeckGL API for WebGL-based data visualization. DeckGL exposes not only high-level JavaScript APIs ("layers") for rendering points, lines, polygons, and text, but also abstractions for developing custom layers with associated custom WebGL shader programs. The heatmap is implemented using a custom layer that performs aggregation on the GPU of neighboring values in an observation-by-feature matrix when multiple values correspond to only a single pixel on the screen. This eliminates the aliasing and Moiré patterns that would otherwise occur at low zoom levels of a large matrix in the heatmap view, while preserving smooth zoom and pan interactions. The custom heatmap layer also contains logic which restricts pan interactions to the matrix area and determines how to display

axis ticks based on zoom level and text length. The spatial view is implemented using multiple custom layers, including one which renders image bitmasks by extending from Viv layers. The spatial and scatterplot views both use a custom layer which efficiently maps feature values to colors on the GPU using quantitative colormap functions written in WebGL.

G.7.4 Implementation of data loading

Data is loaded using HTTP from static files on local or remote web servers. Files corresponding to one or more datasets are specified via an array of URLs and file types in the configuration. Certain file types accept options that specify details of the internal file organization to enable lookups (*e.g.*, to load particular arrays within Zarr stores by specifying their relative paths). Each file type is loaded by a corresponding JavaScript class that defines a required load function and optional functions to load data subsets (where file formats allow). Data loading classes may perform validation, in particular when file formats such as JSON are used which may vary widely in their contents.

G.8 Processing of data for use cases

Data for use cases shown in Figure 3.2 was processed using Python scripts, Jupyter notebooks, and Snakemake pipelines. Vitessce visualizations were configured using the Vitessce Python package. Configurations were exported to JSON files and uploaded to the GitHub Gist service to enable referencing them by URL. Files obtained from the HuBMAP Data Portal were processed by automated pipelines developed within the HuBMAP Infrastructure and Engagement Collaboratory (HIVE).

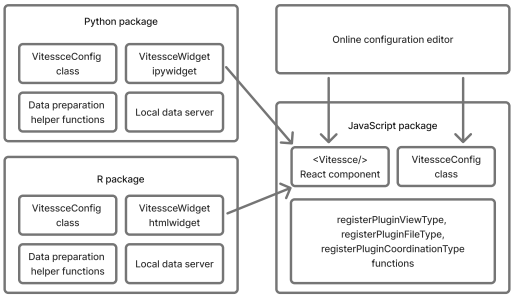

Figure G.9: Implementations of Vitessce for different programming languages and computational environments. The Python package, R package, and the online configuration editor depend on the Vitessce React component distributed via the JavaScript package.

G.9 Software Architecture

Vitessce is an open-source framework for interactive visualization of single-modality, multimodal, and/or spatially-resolved single-cell datasets that can be used in multiple programming languages and computational environments (Figure G.9). Visualizations are implemented in JavaScript and other web technologies (*e.g.*, SVG, WebGL) that are designed to run in a web browser. Python and R packages enable usage of Vitessce without knowledge of JavaScript. These Python and R packages contain widgets for visualization in interactive computing environments (*e.g.*, Jupyter Notebooks and Shiny apps), server modules for accessing local datasets via HTTP, configuration modules for defining visualizations using standard object-oriented syntax of each language, and helper functions for saving data structures to formats that are compatible with Vitessce. The online configuration editor embeds the Vitessce React component and a JavaScript code editor for using Vitessce directly in a web browser without any software installation requirements. A plugin API can be used to define new visualiza-

tion components and data loaders in JavaScript.

G.9.1 Configuration

The Vitessce configuration specifies which views to render, how to layout views in a grid, how views are coordinated, where data files are located, and how data files are organized. At the lowest level, this configuration is passed as a JSON object to the Vitessce React component in JavaScript. While JSON syntax is human-readable, it can be tedious to write and may not be familiar to Python or R programmers. We provide object-oriented APIs in JavaScript, Python, and R which compile JSON configurations internally to simplify the process. Upon attempted initialization of the React component based on a provided configuration, the JSON object is validated against a JSON schema to catch syntactic errors. The configuration schema is versioned, and each successive schema version must be paired with a function to automatically upgrade configurations defined against the preceding schema version. By recursively upgrading configurations from previous schema versions, the Vitessce React component remains backwards compatible with previously-defined configurations while allowing the schema to evolve.

To support coordinated multiple views, not only is the state of Vitessce stored in the coordination model, but also the coordination model is integrated into the configuration. A top-level coordination property stores, for every coordination type, a dictionary mapping coordination scopes to their current values. The top-level layout property defines a list of views which each have a dictionary mapping coordination types to coordination scopes.

The configuration also enables visualization of multimodal datasets by specification of which axes of data align based on shared observation and feature types.

Datasets, observation types, and feature types are all implemented using the coordination model. While coordination is typically described for visualization and interaction properties that may be anticipated to change frequently, it is used here for data-related properties to support multimodal visualization. Views use the coordination model to "look up" their data. On each render, views get their current configuration values from the coordination space, including the value for the dataset coordination type. Views then request data according to dataset, data type, and any additional coordination values they specify. In Python-like pseudocode, a data file matches a view's request for data of a particular type if:

```
def partial_comparison(A, B):
    for key_b, val_b in B.items():
        if A[key_b] != val_b:
            return False
    return True

def is_match(view, data_type, file):
    return (
        view.dataset == get_containing_dataset(file)
        && data_type == get_data_type(file.file_type)
        && partial_comparison(
            file.coordination_values,
            view.coordination_values
        )
    )
```

Listing 10: Definition of observation and feature types enables multimodal data visualization.

Through the aforementioned method of matching views to files, multimodal data visualization is enabled by specification of observation and feature types for files and views. Consider a case in which A represents a cell-by-gene matrix, B a cell-by-peak matrix, and C a nuclei-by-gene matrix (Figure G.10). When configuring Vitessce, the three observation-by-feature matrices can be matched to three different heatmap

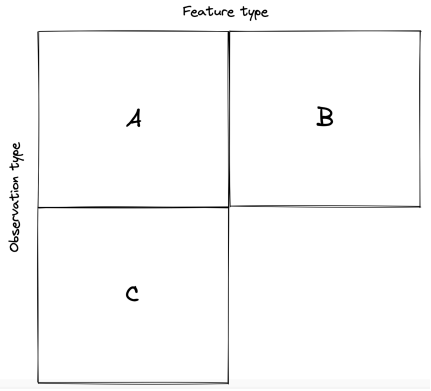

Figure G.10: Definition of observation and feature types enables multimodal data visualization.

views:

```
Heatmap A:
    Observation type: cell
    Feature type: gene
Heatmap B:
    Observation type: cell
    Feature type: peak
Heatmap C:
    Observation type: nuclei
    Feature type: gene
```

Listing 11: Multiple heatmap views example.

Meanwhile, a common control view to select genes for heatmaps A and C only needs to specify a feature type as it is not concerned with observations. The presence of shared observations and/or features is determined by both shared types and shared identifiers in the indices of data frames for observation and feature annotations.

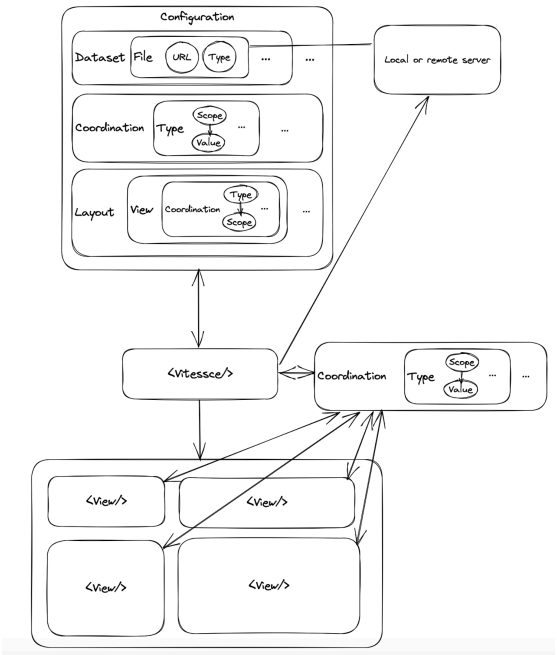

Figure G.11: React component architecture for Vitessce.

G.9.2 JavaScript APIs

React component API

Visualizations are rendered using JavaScript, web technologies, and React[5]. React provides a component-based API, allowing individual visualization components (termed views) to be implemented as independent JavaScript functions (Figure G.11). Vitessce relies on the reactivity paradigms of React to efficiently render views and manage state. In addition to the functions for state management included with React, Vitessce uses the Zustand JavaScript package to store the initial configuration in a centralized location and manage updates to the layout and coordination values. The React component emits configuration (and therefore coordination) updates via an `onConfigChange` handler function that can optionally be provided to the component. This enables a

parent React application to update in response to user interactions that occur within the Vitessce component.

View implementation

Views in Vitessce are implemented using several existing JavaScript packages for data visualization including DeckGL[433], Vega-Lite[378], D3[87], Viv[288], and HiGlass[231]. DeckGL is used for the scatterplot, spatial, and heatmap visualization components. DeckGL provides an abstraction over the WebGL APIs exposed by the browser, with a particular focus on support for geospatial visualization types. Visualizations implemented using DeckGL and WebGL are scalable because rendering operations can occur in parallel on the GPU. In the spatial component, we use the previously-described Viv library to load and render multiplexed multi-scale imaging data without the need for a specialized server[288]. To visualize genome browser tracks and Hi-C heatmaps, we have integrated the previously-described HiGlass library into Vitessce[231].

Notable view implementation details

Scatterplot force simulation-based text label positioning. Optionally, users can render the text labels corresponding to the names of the currently-selected cell sets in the scatterplot, providing an alternative to using color or tooltips to encode cell set information. The text label positioning begins by computing a centroid coordinate for each cell set. As usage of these centroid positions directly can result in multiple overlapping rendered labels, we use a force simulation approach to identify improved non-overlapping positions. To do so, we obtain the position and dimensions of the rectangle corresponding to each text label, based on the text length, font size,

and centroid coordinate. Then, we use the force simulation function from the D3 library[87], with a collision function adapted for rectangular (as opposed to circular) elements[23]. We then run the simulation for 100 ticks, resulting in improved (though in some cases not perfect) label positions.

Scatterplot quadtree-based lasso selection. To efficiently determine which scatterplot points are contained in a lasso selection, we store all scatterplot points using a quadtree data structure[87]. Then, upon user lasso selection, we obtain the polygon formed by the lasso. We visit each node of the quadtree and determine whether the lasso polygon contains, is contained within, or overlaps with the rectangle polygon corresponding to the visited node. If so, and the node is a leaf node, then we check each of its contained points to determine whether it falls within the lasso polygon. On the other hand, if none of the node-level criteria are met, then we ignore the node and move to visit the next node.

View coordination

Views in Vitessce are rendered in a grid layout, enabling the same data to be displayed using different visual encodings in juxtaposed views. Coordinated multiple views is achieved through implementation of the abstract coordination model proposed by Boukhelifa *et al.*[88]. Vitessce supports a set of coordination types on which multiple views can be linked. A coordination type in Vitessce is defined by a semantic property name such as "spatial rotation" or "heatmap zoom" and a programming language-like type such as "boolean" or a more complex array or object type. Named coordination type instances, termed coordination scopes, allow subsets of views to be linked to particular coordination types. The term "coordination space" refers to the

container for all coordination objects in a visualization system. Vitessce stores the coordination space in the JSON-serializable configuration object, through which the coordination state can be saved and shared. The JavaScript implementation uses the Zustand package for state management internally, while views use custom React hook functions that abstract away coordination details by exposing simple getter and setter functions corresponding to the configured coordination scopes for each view.

Data loading

Views load data based on a set of abstract data types which may be compatible with more than one file type. Data loading classes are implemented using standard JavaScript class syntax and must conform to a common interface that defines one load function at a minimum. To support caching of data that may overlap between file definitions, a data source instance is shared between data loader instances for multiple files that point to the same URL. For example, this enables data loaders for AnnData objects to cache index columns via the data source. Internally, views use custom React hook functions to load data corresponding to particular data types, abstracting away file format details.

These abstract data types correspond to elements of data structures such as SpatialData[291]. The SpatialData project addresses data storage and coordinate transformation problems for spatially-resolved omics datasets. Vitessce currently supports visualization of SpatialData's SpatialElements including tables, images, labels (segmentations), and shapes (circles), with support for point and non-circular shape elements planned.

Plugin API

The Vitessce JavaScript package exposes a plugin API for developers to extend Vitessce with new view types, file types, and coordination types. This API is implemented in the form of functions which register new functionality by assigning JavaScript implementations to strings that can be referenced from the JSON configuration. Plugin view types can be defined as React components in plain JavaScript functions using any internal or external dependencies to implement visualizations. Plugin file types enable developers to add support for custom file formats and can be defined as plain JavaScript classes that implement the aforementioned data loading interface (*i.e.*, a class with a "load" function). Plugin coordination types can be defined by specifying a string name and a default value. More information about plugin development can be found on the main documentation website at http://vitessce.io/docs/.

G.9.3 Coordinated multiple views

We implement coordinated multiple views by drawing upon the abstract model proposed by Boukhelifa *et al.*[88] and described in further detail in Appendix A. The following differences exist in terms of each rudiment of coordination:

- Coordination entities: The description of the original model mentions entities related to the window, view, data, record, tuple, attribute, parameter, process, event, function, graphic, and time. The views currently implemented in Vitessce support coordination related to data, visualization parameters, data transformation functions, and behavioral modifiers. These built-in coordination types can be expanded through the plugin API.

- Coordination types: Each coordination type is defined by a name, a JSON schema specifying its programming language-like data type (*e.g.*, string, number, or array of boolean values), and a default value. Directionality of coordination is not addressed explicitly, but views that use both getter and setter functions for a given coordination type are bidirectional while those that use either function exclusively are effectively unidirectional (*i.e.*, read-only or write-only).
- Chronology, lifetime, and scheduling: The implementation does not currently support configuring lifetime or chronology of coordination. However coordination is not necessarily permanent as mappings from views to coordination scopes could be dynamically updated by views internally or by the parent React component.
- Scope, render groups, and granularity of links: Each view is mapped to one scope per coordination type. In future versions of Vitessce, this mapping may be extended to support multiple scopes per coordination scope and/or finer levels of granularity.
- Initialization: Coordination is initialized through the configuration. An automatic initialization strategy can be specified to fill omitted values with defaults.
- Updating: Views in Vitessce do not represent a visual history so issues of synchronization are not applicable.
- Realization: Coordination is not explicitly conveyed to the user but the issue of realization is a priority for future research.
- Translation functions: While views do not need to define translation functions explicitly, view implementations may translate values internally before using

them for rendering or passing them to children React components.

G.9.4 Object-oriented configuration APIs

The Python, R, and JavaScript packages for Vitessce each contain object-oriented configuration APIs written using their respective syntax and idioms (Figure G.9). These APIs simplify the process of specifying datasets, views, and view coordinations in configurations. Users can render the objects generated through these APIs as visualizations in interactive environments or exported to the JSON-based format that is compatible with the Vitessce web application and React component. The object-oriented APIs are designed to enable chained function calls, which enables both instantiation of the configuration object and definition of datasets, views, and coordination in a single language statement. In all three languages, a configuration object can be instantiated not only via the class constructor, but also via a static method that takes a JSON-based configuration as an input. In Python, the configuration object is also able to output an imperative Python code string that can be used to generate an identical configuration, enabling automatic generation of Jupyter notebooks that use the object-oriented Python API from declarative JSON inputs for improved readability and educational purposes.

G.9.5 Online Configuration Editor

The online configuration editor embeds both Vitessce and a text editor for configuring Vitessce using either JSON (declarative) or JavaScript (imperative and object-oriented) syntax. The JavaScript object-oriented code is executed and exported to JSON directly in the web browser using react-live[32].

G.9.6 Python and R packages

The Python and R packages for Vitessce are designed to be used in interactive computational environments during data analysis. These packages contain not only the object-oriented configuration APIs described above, but also helper functions for data preparation and interactive widgets for visualization.

Local data server modules of the R and Python packages automatically serve local data for visualization, but are not required (*e.g.*, all data files may be located on remote servers or the user may opt out to serve local files independently). Local data can be served when the user passes a local file path (rather than a URL) to a class representing a data object in Python or R.

The Python package provides helper functions for exporting local data specified in configurations to a local directory or a remote object store such as AWS S3. These functions are designed to help users transfer data for long-term hosting, enabling them to share visualizations configured using remote file URLs. We provide documentation of this export process as well as the documentation about configuring different object storage platforms to host data that is intended to be accessed by Vitessce (https://python-docs.vitessce.io/).

Interactive widgets

Interactive widgets in Python and R are implemented as wrappers around the Vitessce React component and JavaScript code, with additional functionality for communication between the Python/R programming environments and the JavaScript code running in the web browser. In Python, the widget implementation is based on Anywidget[289]. The R widget is implemented using Htmlwidgets[41], with a planned future

implementation using Anyhtmlwidget[15].

G.10 Planned designs

There are a number of not only visualizations but also user interface elements which we have designed but have not yet implemented.

G.10.1 Sidebar

We have designed a sidebar element which we plan to implement for Vitessce (Figure G.12). This sidebar contains controls for action recovery (undo/redo), adding views, generating shareable URLs, changing the theme, and editing the JSON configuration. This sidebar will enable users to make certain changes to the visualization configuration directly from the graphical interface which are currently accessible from programmatic APIs.

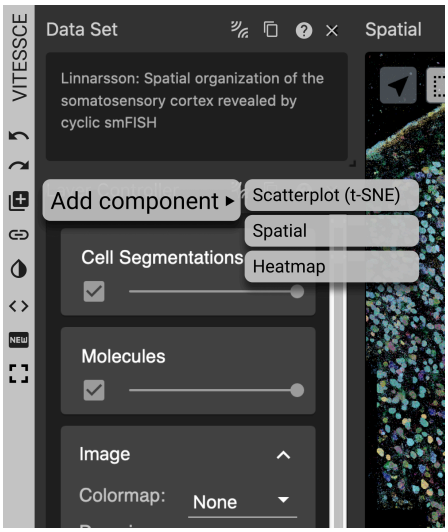

Figure G.12: Design of sidebar for Vitessce.

H

Supplementary Information for Chapter 4

H.1 Formative Interview Questions

Each interview participant was asked the following questions:

- Q1. Which KPMP persona best describes you?

Participants were asked to choose from this set of seven personas that had been previously developed by KPMP to guide consortium-wide software design and development efforts.

  - Patient: “If KPMP could come up with clear answers about my disease, it would be great.”
  - Clinician: “I‘m hoping KPMP gives me the ability to link my individual patients to the best treatments.”
  - Pathologist: “I expect KPMP to help me link biopsies to outcomes and mechanism.”
  - Student: “As a future clinician/scientist, I look for resources to help me understand kidney function and failure at a molecular level”
  - Researcher, Summary Data Consumer: “I‘m not a bioinformatician. I just need the highlights for a quick validation of my gene of interest in AKI and CKD”
  - Researcher, Data Analyst: “I work closely with my wet lab and want to use my analytical skills to answer their biological questions using KPMP.”
  - Reseacher, Data Miner: “I develop tools and analytic methods for kidney researchers to use in KPMP”

- Q2. What are your goals during comparative analysis of multiple datasets? The term ‘multiple datasets’ may refer to multiple single-cell experiments, multiple [microscopy or pathology] images, or a dataset comprising multiple tissue donors (within or across groups defined by clinical metadata).
- Q3. What data is required to answer your questions?

  - Q3.1. Which data types are required? (*e.g.*, spatially-resolved)
  - Q3.2. Which file formats are required?
  - Q3.3. Which external resources and databases are required?

- Q4. What is your current workflow to conduct comparative analysis?
  - Q4.1. Where does your workflow start? (*e.g.*, hypothesis-driven vs. hypothesis-generating / exploratory)
  - Q4.2. Which software packages are used?
  - Q4.3. What is the resolution of comparison? (*e.g.*, coarse vs. fine cell types, genes vs. pathways)

- Q5. How do you currently determine statistical significance in this workflow?
  - Q5.1. Statistical tests?
  - Q5.2. Software packages?

- Q6. How do you currently approach data visualization in this workflow?
  - Q6.1. What types of plots do you create in different parts of the workflow? (*e.g.*, during quality control, data analysis, communication with collaboration, communication to journal audience)

- Q7. What are the limitations of your current workflow?
  - Q7.1. What are limitations of current tools?
  - Q7.2. Are there parts of the comparative analysis workflow that are relatively more difficult than others?

H.2 Formative Interview Results

We group the analysis goals into the following four scenarios:

- **T2.3**: Biomarkers of interest (confirmatory), no stratification of samples
- **T2.1**: Biomarkers of interest (confirmatory), stratification of samples (case vs. control)
- **T2.4**: Lack of biomarkers of interest (exploratory), no stratification of samples
- **T2.2**: Lack of biomarkers of interest (exploratory), stratification of samples (case vs. control)

Within these scenarios, users have analysis goals that involve comparisons among the following types of items:

- **CT**: Cell type
- **N**: Neighborhood
- **SI**: Segmented instance
- **I**: Image
- **P**: Participant

Organized by these scenarios and comparison levels, the following questions arose from users:

- T2.3, CT: Does the quantification of the biomarkers of interest differ among cell types (*e.g.*, is the biomarker level elevated in all cell types, a subset, or none)?

- T2.3, CT: If multiple biomarkers: Do the patterns of individual biomarkers within the set differ from the aggregate of all selected biomarkers?
- T2.3, N: What does the spatial transcriptomics data look like for the selected biomarker in spatial regions containing the selected neighborhood type, across all datasets of the selected experimental technology?
- T2.3, SI: What does the spatial transcriptomics data look like for the selected biomarker in spatial regions containing the selected segmentation type, across all datasets of the selected experimental technology?
- T2.3, I: What does the spatial transcriptomics data look like for the selected biomarker in representative images at different points along the distribution of expression (contained within the selected segmentation type), across all datasets of the selected experimental technology?
- T2.3, P: How many participant datasets contain expression of each gene in the signature?
- T2.3, P: For which participants are the selected genes significantly differentially expressed within a cell type (compared to all other cell types for that participant)?
- T2.3, P: What is the distribution of mean expression of the gene signature across participants within the selected cell type?
- T2.3, P: For the selected participant (from the distribution of mean scores above), how does their distribution of cell types and expression values compare to the rest of the transcriptomics data in the atlas?

- T2.1, CT: Does the quantification of the biomarkers of interest for each cell type differ between case and control groups?
- T2.1, CT: If multiple biomarkers: Do the patterns of individual biomarkers within the set differ from the aggregate of all selected biomarkers?
- T2.1, N: What does the spatial transcriptomics data look like for the selected biomarker in spatial regions containing the selected neighborhood type, across all datasets of the selected experimental technology? How does this differ between case and control groups?
- T2.1, SI: What does the spatial transcriptomics data look like for the selected biomarker in spatial regions containing the selected segmentation type, across all datasets of the selected experimental technology? How does this differ between case and control groups?
- T2.1, I: What does the spatial transcriptomics data look like for the selected biomarker in representative images at different points along the distribution of expression (contained within the selected segmentation type), across all datasets of the selected experimental technology? How does this differ between case and control groups?
- T2.1, P: How many participant datasets contain expression of each gene in the signature? How does this differ between case and control groups?
- T2.1, P: For which participants are the selected genes significantly differentially expressed within a cell type (compared to all other cell types for that participant), with participants annotated by their group membership?

- T2.1, P: For the selected participant (from the distribution of mean scores above), how does their distribution of cell types and expression values compare to the rest of the transcriptomics data in the atlas? How does this differ between case and control groups?
- T2.4, CT: Which genes are significantly differentially expressed within a cell type (compared to all other cell types)?
- T2.4, CT: Of the genes are significantly differentially expressed within a cell type, what is the distribution of gene expression across all cell types?
- T2.4, CT: Using the significantly differentially expressed genes within each cell type, what pathway terms are enriched?
- T2.4, SI: What does the imaging data look like in spatial regions containing the selected segmentation type, across all datasets of the selected experimental technology?
- T2.4, SI: Which genes are significantly differentially expressed in cells within a segmentation type (compared to cells in all other segmentation types)?
- T2.4, SI: Of the genes are significantly differentially expressed within a segmentation type, what is the distribution of gene expression across all segmentation types?
- T2.4, SI: Using the significantly differentially expressed genes within each segmentation type, what pathway terms are enriched?
- T2.4, P: What are the sizes of participant groups defined by clinical metadata?

- T2.4, P: What are the top differentially-expressed genes per participant within each cell type?
- T2.2, CT: Which genes are significantly differentially expressed between the case and control groups within each cell type?
- T2.2, CT, Q: Of the genes are significantly differentially expressed within each cell type, what is the distribution of gene expression across all cell types? How does this differ between case and control groups?
- T2.2, CT: Using the significantly differentially expressed genes within each cell type, what pathway terms are enriched?
- T2.2, SI: What does the imaging data look like in spatial regions containing the selected segmentation type, across all datasets of the selected experimental technology? How does this differ between case and control groups?
- T2.2, SI: Which genes are significantly differentially expressed between groups in cells within a segmentation type?
- T2.2, SI: Of the genes are significantly differentially expressed within a segmentation type, what is the distribution of gene expression across all segmentation types? How does this differ between case and control groups?
- T2.2, SI: Using the significantly differentially expressed genes within each cell type, what pathway terms are enriched?
- T2.2, P: What are the sizes of participant subgroups defined by clinical metadata, within the case and control groups?

- T2.2, P: What are the top differentially-expressed genes per participant within each cell type, with participants annotated by their group membership?

H.3 Comparative Analysis Survey

Survey respondents were provided the following background information: "Please help us to determine which case-vs.-control style comparisons would be most interesting. Groups defined based on age would mean comparison of young vs. adult, for instance. Groups defined based on segmented functional tissue units (FTUs) could mean comparison of within-glomerulus vs. outside-glomerulus regions or glomerulus vs. tubule, for instance. Groups defined based on organ could mean comparison of kidney vs. lung, for instance."

Then, they were asked the following questions:

- Most interesting: Groups defined based on...
- 2nd-most interesting: Groups defined based on...
- 3rd-most interesting: Groups defined based on...
- 4th-most interesting: Groups defined based on...
- 5th-most interesting: Groups defined based on...
- Least interesting: Groups defined based on...
- Your role: Professor or Principal investigator, Staff, Postdoctoral fellow, Graduate student
- Your expertise: Anatomy, Biology, Bioinformatics, Chemistry, Computer Science, Expert in particular organ / tissue type

- Other comments or suggestions (optional)

For the first six questions, respondents were asked to select one of the following options:

- Sex (donor-level, 100% complete, 233 donors)
- Age (donor-level, 100%, 233)
- Race (donor-level, 100%, 233)
- BMI (donor-level, 86%, 202)
- Ethnicity (donor-level, 22%, 51)
- Blood type (donor-level, 13%, 31)
- Cause of death (donor-level, 27%, 61)
- History of cancer (donor-level, 35%, 82)
- History of hypertension (donor-level, 35%, 82)
- History of hyperlipidimia (donor-level, 35%, 82)
- History of diabetes (donor-level, 35%, 82)
- History of cardiac arrest (donor-level, 35%, 82)
- Smoking history (donor-level, 21%, 49)
- Sun-exposed (sample-level, 25%, 542)
- Organ (spatial)

- Block or section (spatial)
- Segmentations of functional tissue units (spatial)
- Experimental metadata (*e.g.*, instrument model)

H.4 Comparative Visualization Implementation Details

H.4.1 Dimensionality reduction contour plot

Circle-based text label positioning approach. Rather than using the centroid coordinate of many data points, potentially obscuring the points, in this method we position the labels around the outside of all data points, yet aiming to minimize the distance between each text label and the region containing its corresponding data points. This method works by first determining the extent of all data points, along the X and Y axes of the plot. Then, we fit a circle to this region and identify some fixed number of vertices (currently, 96) along this circle. Then, we iterate over each contour (Figure 4.10) and identify the contour polygon having the largest area among the polygons in the same opacity-threshold level (from largest to smallest threshold). If this largest-area polygon for the threshold level is above some minimum size, then we move on without considering polygons at the smaller threshold levels. Next, we iterate over a fixed number (currently, 36) of vertices (at evenly spaced indices) along this polygon and identify the vertex with the minimum distance to any of the vertices on the circle. Finally, we position the contour's text label at this vertex on the circle (or the subsequent free vertex if this vertex has already been assigned to another contour) and render a line between the contour vertex and the circle vertex. We render the line and text according to the color and opacity of the contour polygon used for

positioning.

H.5 Comparative Analysis Metadata Format

We have defined JSON and Zod schemas to store comparative analysis metadata within an AnnData object. We serialize the metadata dictionary and store it in a subkey of the unstructured part of the object. We have also written a Python class to help with constructing such a metadata dictionary within data processing pipeline code.

At the top-level the dictionary contains the following properties:

- `schema_version` (string): The version of the comparison metadata schema.
- `cell_type_cols` (string[]): Array of one or more columns in adata.obs storing cell type annotations.
- `sample_id_col` (string): The column of adata.obs which maps cells to sample IDs.
- `sample_group_pairs` ([string, [string, string]][]): Array of sample group pairs in the form of [colname, [ctrl_val, case_val]].
- `comparisons` ({ [string]: { `comparison`: ComparisonPath, `results`: ComparisonResult[] } }): Dictionary mapping "comparison key"s to objects with the properties comparison and results. The value of comparison should be the "comparison path", a string or (nested) array of strings which represent the pre-slug-ified comparison key. See below for the expected format of ComparisonResult dictionaries. Note that results stores an array, since the same comparison key/path may correspond to more than one comparison result (*e.g.*, for

the comparison key compare_celltype.val_macrophage.___rest___, we might perform both a differential expression test and a gene set enrichment test, producing two separate comparison result dataframes).

Within each of the `results` entries, the dictionary contains the following properties:

- `path` (string): The path to the dataframe, relative to the AnnData object root.
- `coordination_values` ({ [string]: string }): Key-value pairs in the form of Vitessce coordination values (*i.e.*, coordinationType: coordinationValue mappings).
- `analysis_type` (string): The type of comparative analysis method which was applied to generate this result (*e.g.*, rank_genes_groups or enrich).
- `analysis_params` (null|object): Parameters that are relevant to the analysis. This may take a different form depending on the analysis type.

H.6 Planned designs

There are a number of comparative visualizations which we have designed but have not yet implemented.

H.6.1 Spatial representative samples

We have designed comparative visualizations which display a limited number of images (or image regions) corresponding to particular representative samples along a distribution of quantitative values (Figure H.1). For example, the visualization could

display images corresponding to the 25th, 50th, and 75th percentiles of the distribution of a biomarker summary. In scenarios in which a user is interested in a pair of cohorts, the view could display two quantitative distributions side-by-side to enable comparison of two sets of representative samples (Figure H.2).

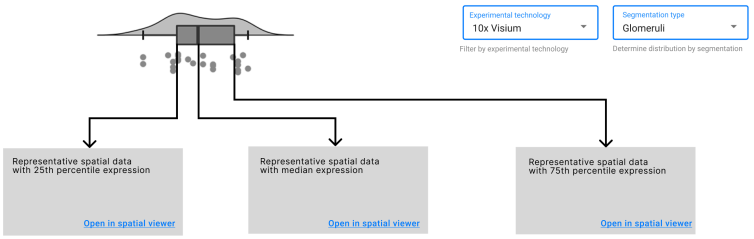

Figure H.1: Design of spatial representative samples view for Vitessce.

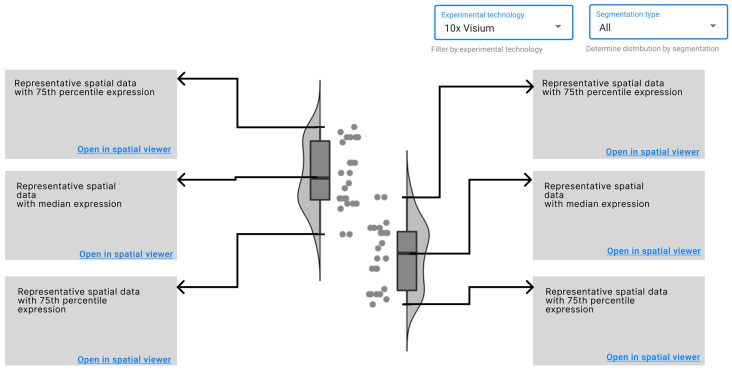

Figure H.2: Design of comparative spatial representative samples view for Vitessce.

H.6.2 Spatial small multiples

Users may be interested to view many spatially-resolved datasets simultaneously, either entire image fields of view or particular zoomed-in regions (*e.g.*, surrounding a particular cell neighborhood or segmented functional tissue unit). In this case, we can use a small multiples approach. Figure H.3 presents a basic design in which this

approach is used. Additional features could be integrated, for instance, through adoption of "piling" techniques[260].

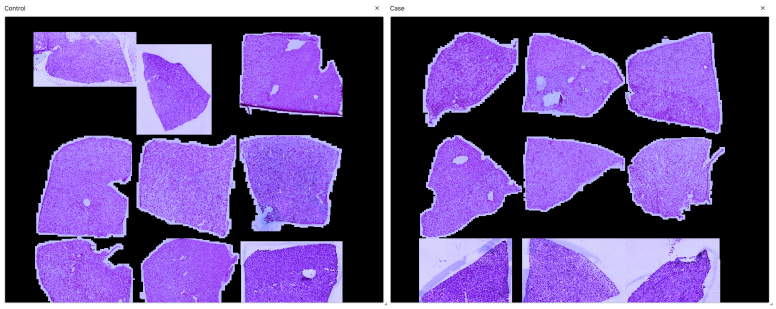

Figure H.3: Design of spatial small multiples view for Vitessce.

I

Supplementary Information for Chapter 5

I.1 Preface

This chapter contains supplementary information for Chapter 5.

I.2 Methods for Section 5.5

I.2.1 Extension of Vitessce

Vitessce (Chapter 3) supports spatially-resolved visualizations (*e.g.*, segmentations) comprising a single type of observation (*e.g.*, cells). We extended the spatial view in Vitessce, generalizing it to support multiple observation types (*e.g.*, multiple types of functional tissue units) simultaneously.

Visualization rendering

We implemented a WebGL fragment shader (compatible with the DeckGL framework used by Vitessce) that renders a multi-channel bitmask image. This shader supports dynamic color mapping on a per-channel level, for both categorical and quantitative colormaps. For example, a user may want to color the Glomeruli bitmask in image channel 0 using a categorical colormap (sclerotic and non-sclerotic categories) while coloring the Tubules bitmask in channel 1 using a quantitative colormap (continuum of luminal fraction values). The shader supports specification of the following visual properties on a per-channel basis: "isFilled", "opacity", "strokeWidth", "colormapRangeStart", "colormapRangeEnd", "isStaticColorMode", "isSetColorMode", "isVisible". In staticColorMode, a channel can be mapped to its own RGB color value which will be used to render all non-zero (*i.e.*, non-background) pixels. In setColorMode, a channel can be passed a mapping between sets of observations and RGB color values (*i.e.*, one color per set ID). In featureColorMode, the shader applies a quantitative colormap function (*e.g.*, magma or viridis) to each observation using a feature value vector (*i.e.*, one feature value per observation) for the channel.

Each channel can be assigned to a different color mode; the color mode is specified per channel. Provided RGB values, set membership values, and feature values are channel-specific. In order to pass these values to the GPU, they are stored efficiently using WebGL textures.

Algorithm for on-the-fly stroked vs. filled bitmask rendering

Bitmask images associate pixels with observation IDs using a "filled" approach (*i.e.*, all pixels corresponding to a cell are assigned the cell ID value). However, it is often desired to render these using a "stroked" approach in which only the observation outline is colored while the interior remains transparent.

A naive approach is to pre-determine which pixels lie along each cell's outline, and construct a separate bitmask image in which the interior pixels of each cell are represented using the background pixel value. While this approach will result in the interior pixels being mapped to a transparent color and therefore only rendering the cell outlines, it has several drawbacks. The application must load two separate bitmask images, one for filled and one for stroked visualization, for each bitmask. These two images must be precomputed and stored, requiring double the disk space and complicating visualization configuration. Finally, the width of the cell outline must be fixed at the time of bitmask construction. For these reasons, we devised an on-the-fly stroked rendering method for filled bitmask images.

This algorithm is implemented as a function within a WebGL fragment shader and is therefore executed for every pixel via the GPU. The function takes as input a single channel bitmask stored as a WebGL texture, the current pixel coordinate, and the user-defined stroke width. Then lookups must be performed for the current pixel value ("currentValue") as well as the values of the pixels located strokeWidth-pixels

away in eight different directions (north, south, west, east, north-west, north-east, south-west, south-east). The function then compares currentValue to the eight other pixel values. If one or more of the eight values differ from currentValue, then we conclude that the current pixel is located less than strokeWidth-pixels away from the edge of the segmentation boundary and therefore should be rendered opaque as opposed to transparent (during stroked rendering mode). Within the same fragment shader, we execute this function for each channel of multi-channel segmentation bitmasks and combine the results with user-defined per-channel properties such as colormaps and opacity values.

Configuration of spatial and layer controller views

We implemented spatial and layer controller views for Vitessce which support a hierarchy of coordinated properties. For example, each segmentation bitmask layer has properties such as "isVisible," with each layer having multiple children segmentation channels (corresponding to functional tissue units) that each have their own visual properties such as "color" and "isFilled". This hierarchical approach to visualization configuration and coordination follows the approach described in Appendix A.

Data loading

To facilitate configuration of multiple segmentation bitmasks contained in the same image file (*e.g.*, multiple channels of an OME-TIFF), we use channel name metadata fields to specify observation types (*i.e.*, the type of functional tissue unit). For example, a two-channel segmentation bitmask image may have the channel names "glomerulus" and "tubule." Then, Vitessce uses the channel name metadata from the

image to determine which channels correspond to each observation type and identify correspondences between channels and other files (*e.g.*, tables containing extracted feature values for glomeruli and tubules).

I.2.2 Development of models for segmentation

Three previously validated deep learning-based segmentation models were used in the segmentation of renal structures and pathology-associated regions in the kidney. Each model was trained on manual and bootstrapped annotations validated by a renal pathologist. The first model is outlined in Lucarelli *et al.*[277], and supports segmentation of both the cortical and medullary interstitia, non-globally-sclerotic glomeruli, globally-sclerotic glomeruli, tubules, and arteries/arterioles. This model is built on the Detectron2 panoptic segmentation architecture. Kidney samples from various institutions, various stain types, and various disease states, including diabetic nephropathy, lupus nephritis, transplant surveillance, and reference tissues were included in the training of the network. This model was previously validated in a PAS-stained holdout set.

The second model is outlined in Jayapandian *et al.*[215], and supports segmentation of the peritubular capillaries. This network is built on the standard U-Net architecture, and was trained using a large sample of the NEPTUNE cohort that contained four different histological stains from patients with Minimal Change Disease (MCD). This model was previously validated in a holdout set with each of the same four stains.

The last model is outlined in Ginley *et al.*[169], and supports segmentation of the pathology-associated regions of Interstitial Fibrosis and Tubular Atrophy (IFTA). This model is built on the DeeplabV2 semantic segmentation architecture, and was

trained using a large, multi-institutional cohort of PAS-stained renal biopsies from patients with Diabetic Nephropathy, transplant kidneys, deceased-donor kidneys, and reference kidneys. The model was previously tested in a PAS-stained holdout set, and compared to multiple pathologist annotations for agreement.

I.2.3 Extraction of pathomics features

Hand-engineered pathomic features were designed with the help of renal pathologists for clinical relevance and explainability. These features can be classified as morphological features, texture features, color features, and distance transform features. Sizes, shapes, color means and standard deviations, and thicknesses of sub-segmentations make up the majority of the feature set.

I.2.4 Specification for on-disk representation

Label images may be stored using OME-TIFF or OME-NGFF (including OME-NGFF images contained within the Labels element of a SpatialData object). Each channel represents a different type of observation. Channel names indicate the type of observation contained in the channel. The pixel value zero is used to indicate background. A segmentation for the ith observation of a given type is indicated by pixels with value i. Each observation may be associated with feature values (*i.e.*, an observation-by-feature matrix for each channel, where each row corresponds to an observation belonging to the channel's observation type). These observation-by-feature matrices may be stored in an AnnData object, MuData object, or CSV file. The observation index values for each observation-by-feature matrix should match the non-zero pixel values within the corresponding label image (*i.e.*, should consist of 1-indexed integers,

although not necessarily be ordered sequentially or gapless).

I.2.5 Data availability

The data described in this manuscript is available from the KPMP Atlas data portal[43].

I.2.6 Code availability

The JavaScript and WebGL code which implements the interactive visualization features described here are contained in the Vitessce GitHub repository. Segmentation and feature extraction code are available on GitHub[26,20].

I.3 Proposal for using OME-NGFF image format to store point-based data

One idea for increasing the scalability of point-based data storage (*e.g.*, for MERFISH transcript points), loading, and visualization would be to reuse the OME-NGFF[311] image format (Figure I.1).

In this proposal, we would use the "channel" dimension of an OME-NGFF image to store X, Y, and gene index values (*i.e.*, channel 0 would store X-coordinates, channel 1 would store Y-coordinates, and channel 2 would store gene/transcript indices). With an additional image channel, the image could store Z-coordinates of points, to support 3D datasets.

For example, to store transcripts originating from a MERFISH, experiment, we could construct a point-image file with horizontal and vertical dimensions at least as large as the underlying microscope image that the points originated from (*i.e.*, the

image prior to point detection). As there cannot be more than one point per pixel, this would facilitate storage of all transcript points. This would also enable point X/Y coordinates to be stored as integers without loss of information (although this does not necessarily need to be the case). It would likely be the case that most image tiles would lack a full set of points, in which case the tiles could be padded with null values (and we would defer to the built-in compression features of Zarr to negate the on-disk impact of null-value padding).

While we hypothesize that this proposal would enable visualization applications to efficiently query and render large point-based datasets (and leverages OME-NGFF to avoid reinventing the format wheel), this proposal remains to be benchmarked against conceptually-simpler formats to determine whether there is a performance benefit to justify it. Tools in the data analysis ecosystem would also need to buy-in to such a proposal to reduce data conversion overhead.

I.3.1 Relative coordinates optimization

This proposal allows for optimization of the storage of point X/Y coordinates by only storing the offset of each point relative to the edge of the image tile (Zarr chunk), as opposed to "absolute" coordinates (*i.e.*, relative to the edge of the full field-of-view of the image). This optimization could reduce the size of the resulting OME-NGFF image significantly by allowing for usage of smaller NumPy integer data types to store X/Y coordinate values (*e.g.*, `uint8` if 256×256 chunk shapes). Further, such an optimization would enable efficient loading of data over a network.

However, this optimization would mean that array values would be dependent on the chunk shape of the array. Usage of tools such as rechunker[33] to modify the chunk shape of an array would therefore need to be accompanied by changes to all X/Y co-

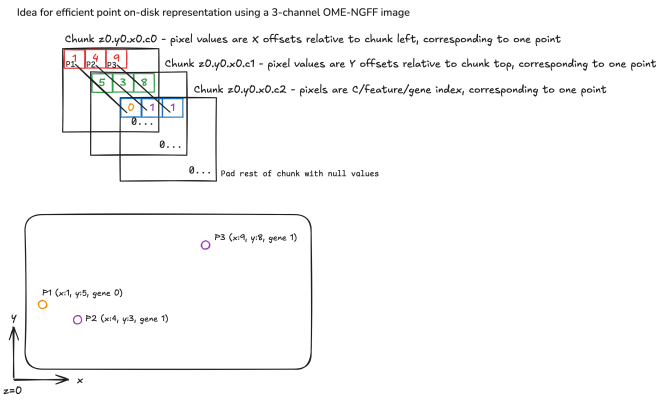

Figure I.1: Proposal for on-disk representation of point data using OME-NGFF. Diagram corresponding to a single chunk in space 0.0.0 (z,y,x).

ordinates to ensure that their values remain chunk-relative under the modified chunk shape. Yet for large-scale image data, one might assume that Zarr array chunk shapes will not be changed frequently (or at all).

Note that it would also be possible to discard the relative-coordinates part of this proposal yet still use the other aspects (to alleviate concerns about array values being dependent on chunk shape).

References

[1] Relevance of 3D Imaging. https://www.alpenglowbiosciences.com/why-3d. Last accessed 2025-01-08.

[2] A 'Wikipedia for cells': Researchers get an updated look at the Human Cell Atlas, and it's remarkable. *Nature*, 635(8039):523–524, November 2024. doi: 10.1038/d41586-024-03754-y.

[3] Niivue. https://github.com/niivue/niivue, 2024. Last accessed 2024-07-25.

[4] react-plotly.js. https://github.com/plotly/react-plotly.js, 2024. Last accessed 2024-07-25.

[5] React. https://github.com/facebook/react, 2024. Last accessed 2024-07-25.

[6] ReactFlow. https://github.com/xyflow/xyflow, 2024. Last accessed 2024-07-25.

[7] SolidJS. https://github.com/solidjs/solid, 2024. Last accessed 2024-07-25.

[8] Svelte. https://github.com/sveltejs/svelte, 2024. Last accessed 2024-07-25.

[9] Visx. https://github.com/airbnb/visx, 2024. Last accessed 2024-07-25.

[10] Vue. https://github.com/vuejs/core, 2024. Last accessed 2024-07-25.

[11] Zod. https://github.com/colinhacks/zod, 2024. Last accessed 2024-07-25.

[12] Visualization for spatial single-cell atlases: Identifying challenges and opportunities in complex biomedical data at multiple scales and modalities. https://ieeevis.org/year/2023/info/spotlights, 2025. Last accessed 2025-01-22.

[13] BioMedVis challenge at IEEE VIS 2024. http://biovis.net/2024/biovisChallenges_vis/, 2025. Last accessed 2025-01-22.

[14] Figma. https://www.figma.com/, 2025. Last accessed 2025-04-14.

[15] anyhtmlwidget. https://github.com/keller-mark/anyhtmlwidget, 2025. Last accessed 2025-04-14.

[16] celldega. https://github.com/broadinstitute/celldega, 2025. Last accessed 2025-04-14.

[17] cistrome-explorer. https://github.com/hms-dbmi/cistrome-explorer, 2025. Last accessed 2025-04-14.

[18] cistrome-explorer-higlass-server. https://github.com/hms-dbmi/cistrome-explorer-higlass-server, 2025. Last accessed 2025-04-14.

[19] cistrome-to-multivec pipeline. https://github.com/hms-dbmi/cistrome-explorer/tree/master/pipelines/cistrome-to-multivec, 2025. Last accessed 2025-04-14.

[20] CombinedFeatureExtraction. https://github.com/SarderLab/CombinedFeatureExtraction, 2025. Last accessed 2025-04-14.

[21] esbuild-py. https://github.com/keller-mark/esbuild-py, 2025. Last accessed 2025-04-14.

[22] geotiff.js. https://github.com/geotiffjs/geotiff.js, 2025. Last accessed 2025-04-14.

[23] rectCollide. https://gist.github.com/cmgiven/547658968d365bcc324f3e62e175709b, 2025. Last accessed 2025-04-14.

[24] higlass-bigwig-datafetcher. https://github.com/higlass/higlass-bigwig-datafetcher, 2025. Last accessed 2025-04-14.

[25] higlass-zarr-datafetchers. https://github.com/higlass/higlass-zarr-datafetchers, 2025. Last accessed 2025-04-14.

[26] Multi-Compartment-Segmentation. https://github.com/SarderLab/Multi-Compartment-Segmentation, 2025. Last accessed 2025-04-14.

[27] Kidney chromatin landscape browser. https://kpmp.github.io/gisch-et-al-2023/, 2025. Last accessed 2025-04-14.

[28] pizzarr. https://github.com/keller-mark/pizzarr, 2025. Last accessed 2025-04-14.

[29] higlass-zarr-datafetchers. https://github.com/hubmapconsortium/portal-containers/tree/master/containers/mudata-to-ui, 2025. Last accessed 2025-04-14.

[30] qdpx-to-json. https://github.com/keller-mark/qdpx-to-json, 2025. Last accessed 2025-04-14.

[31] Rarr. https://github.com/grimbough/Rarr, 2025. Last accessed 2025-04-14.

[32] react-live. https://github.com/FormidableLabs/react-live, 2025. Last accessed 2025-04-14.

[33] rechunker. https://github.com/pangeo-data/rechunker, 2025. Last accessed 2025-04-14.

[34] sc-star-site. https://github.com/keller-mark/sc-star-site, 2025. Last accessed 2025-04-14.

[35] SpatialQuery. https://github.com/ShaokunAn/Spatial-Query, 2025. Last accessed 2025-04-14.

[36] two.js. https://github.com/jonobr1/two.js, 2025. Last accessed 2025-04-14.

[37] use-coordination-solid. https://github.com/keller-mark/use-coordination-solid, 2025. Last accessed 2025-04-14.

[38] viv. https://github.com/hms-dbmi/viv, 2025. Last accessed 2025-04-14.

[39] with-coordination. https://github.com/manzt/with-coordination, 2025. Last accessed 2025-04-14.

[40] zarr.js. https://github.com/gzuidhof/zarr.js, 2025. Last accessed 2025-04-14.

[41] htmlwidgets. https://www.htmlwidgets.org/, 2025. Last accessed 2025-04-14.

[42] KPMP atlas. https://atlas.kpmp.org/explorer, 2025. Last accessed 2025-04-14.

[43] Segmentation mask visualizations in KPMP Atlas. https://atlas.kpmp.org/spatial-viewer/?filters[0][field]=imagetype&filters[0][values][0]=PAS(SegmentationMasks)&filters[0][type]=any, 2025. Last accessed 2025-04-14.

[44] Course 6.859: Interactive data visualization. https://vis.csail.mit.edu/classes/6.859/, 2025. Last accessed 2025-04-14.

[45] Vitessce documentation: usage showcase. https://vitessce.io/docs/showcase/, 2025. Last accessed 2025-04-14.

[46] Zarr implementations. https://zarr.dev/implementations/, 2025. Last accessed 2025-04-14.

[47] zarrita js. https://github.com/manzt/zarrita.js, 2025. Last accessed 2025-04-14.

[48] Miri Adler, Arun R. Chavan, and Ruslan Medzhitov. Tissue Biology: In Search of a New Paradigm. *Annual Review of Cell and Developmental Biology*, 39(1):67–89, October 2023. ISSN 1081-0706, 1530-8995. doi: 10.1146/annurev-cellbio-120420-113830.

[49] D. Agrafiotis. Stochastic proximity embedding. *Journal of Computational Chemistry*, 24, 2003. Referenced in Table C.1.

[50] Constantin Ahlmann-Eltze and Wolfgang Huber. Comparison of transformations for single-cell RNA-seq data. *Nature Methods*, 20(5):665–672, May 2023. ISSN 1548-7105. doi: 10.1038/s41592-023-01814-1.

[51] Constantin Ahlmann-Eltze and Wolfgang Huber. Analysis of multi-condition single-cell data with latent embedding multivariate regression. *Nature Genetics*, pages 1–9, January 2025. ISSN 1546-1718. doi: 10.1038/s41588-024-01996-0.

[52] Derya Akbaba and Miriah Meyer. "Two heads are better than one": Pair-interviews for visualization. In *2023 IEEE Visualization and Visual Analytics (VIS)*, pages 206–210, 2023. doi: 10.1109/VIS54172.2023.00050.

[53] Mayar Allam and Ahmet F. Coskun. Combining spatial metabolomics and proteomics profiling of single cells. *Nature Reviews Immunology*, pages 1–1, August 2024. ISSN 1474-1741. doi: 10.1038/s41577-024-01084-8.

[54] Chris Allan, Jean-Marie Burel, Josh Moore, Colin Blackburn, Melissa Linkert, Scott Loynton, Donald MacDonald, William J. Moore, Carlos Neves, Andrew Patterson, Michael Porter, Aleksandra Tarkowska, Brian Loranger, Jerome Avondo, Ingvar Lagerstedt, Luca Lianas, Simone Leo, Katherine Hands, Ron T. Hay, Ardan Patwardhan, Christoph Best, Gerard J. Kleywegt, Gianluigi Zanetti, and Jason R. Swedlow. OMERO: Flexible, model-driven data management for experimental biology. *Nature Methods*, 9(3):245–253, March 2012. ISSN 1548-7105. doi: 10.1038/nmeth.1896.

[55] Susan J. Allison. HE4—a biomarker and target in renal fibrosis. *Nature Reviews Nephrology*, 9(3):124–124, March 2013. ISSN 1759-507X. doi: 10.1038/nrneph.2013.2.

[56] Seth A. Ament, Marcia Cortes-Gutierrez, Brian R. Herb, Evelina Mocci, Carlo Colantuoni, and Margaret M. McCarthy. A single-cell genomic atlas for maturation of the human cerebellum during early childhood. *Science Translational Medicine*, 15(721):eade1283, October 2023. doi: 10.1126/scitranslmed.ade1283.

[57] Robert A. Amezquita, Aaron T. L. Lun, Etienne Becht, Vince J. Carey, Lindsay N. Carpp, Ludwig Geistlinger, Federico Marini, Kevin Rue-Albrecht, Davide Risso, Charlotte Soneson, Levi Waldron, Hervé Pagès, Mike L. Smith, Wolfgang Huber, Martin Morgan, Raphael Gottardo, and Stephanie C. Hicks. Orchestrating single-cell analysis with Bioconductor. *Nature Methods*, 17(2):137–145, February 2020. ISSN 1548-7105. doi: 10.1038/s41592-019-0654-x.

[58] Ido Amit, Kristin Ardlie, Fabiana Arzuaga, Gordon Awandare, Gary Bader, Alexander Bernier, Piero Carninci, Stacey Donnelly, Roland Eils, Alistair R. R. Forrest, Henry T. Greely, Roderic Guigo, Nir Hacohen, Muzlifah Haniffa, Emily Sarah Kirby, Bartha Maria Knoppers, Arnold Kriegstein, Ed S. Lein, Sten Linnarsson, Partha P. Majumder, Miriam Merad, Kerstin Meyer, Musa M. Mhlanga, Garry Nolan, Ntobeko A. B. Ntusi, Dana Pe'er, Shyam Prabhakar, Maili Raven-Adams, Aviv Regev, Orit Rozenblatt-Rosen, Senjuti Saha, Andrea Saltzman, Alex K. Shalek, Jay W. Shin, Henk Stunnenberg, Sarah A. Teichmann, Timothy Tickle, Alexandra-Chloe Villani, Christine Wells, Barbara Wold, Huanming Yang, and Xiaowei Zhuang. The commitment of the human cell atlas to humanity. *Nature Communications*, 15(1):10019, November 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-54306-x.

[59] Elisa Amorim, E. V. Brazil, L. G. Nonato, F. Samavati, and M. Sousa. Multidimensional projection with radial basis function and control points selection. *2014 IEEE Pacific Visualization Symposium*, pages 209–216, 2014. Referenced in Table C.1.

[60] Maryanne E. Ardini-Poleske, Robert F. Clark, Charles Ansong, James P. Carson, Richard A. Corley, Gail H. Deutsch, James S. Hagood, Naftali Kaminski, Thomas J. Mariani, Steven S. Potter, Gloria S. Pryhuber, David Warburton, Jeffrey A. Whitsett, Scott M. Palmer, Namasivayam Ambalavanan, and LungMAP Consortium. LungMAP: The Molecular Atlas of Lung Development Program. *American Journal of Physiology. Lung Cellular and Molecular Physiology*, 313(5):L733–L740, November 2017. ISSN 1522-1504. doi: 10.1152/ajplung.00139.2017.

[61] Ricard Argelaguet, Anna S. E. Cuomo, Oliver Stegle, and John C. Marioni. Computational principles and challenges in single-cell data integration. *Nature Biotechnology*, 39(10):1202–1215. ISSN 1546-1696. doi: 10.1038/s41587-021-00895-7.

[62] Amjad Askary, Wei Chen, Junhong Choi, Lucia Y. Du, Michael B. Elowitz, James A. Gagnon, Alexander F. Schier, Sophie Seidel, Jay Shendure, Tanja Stadler, and Martin Tran. The lives of cells, recorded. *Nature Reviews Genetics*, pages 1–20, November 2024. ISSN 1471-0064. doi: 10.1038/s41576-024-00788-w.

[63] Rengasamy Asokan, Jing Hua, Kendra A. Young, Hannah J. Gould, Jonathan P. Hannan, Damian M. Kraus, Gerda Szakonyi, Gabrielle J. Grundy, Xiaojiang S. Chen, Mary K. Crow, and V. Michael Holers. Characterization of Human Complement Receptor Type 2 (CR2/CD21) as a Receptor for IFN-α: A Potential Role in Systemic Lupus Erythematosus1. *The Journal of Immunology*, 177(1):383–394, July 2006. ISSN 0022-1767. doi: 10.4049/jimmunol.177.1.383.

[64] Ethan A. G. Baker, Denis Schapiro, Bianca Dumitrascu, Sanja Vickovic, and Aviv Regev. In silico tissue generation and power analysis for spatial omics. *Nature Methods*, 20(3):424–431, March 2023. ISSN 1548-7105. doi: 10.1038/s41592-023-01766-6.

[65] Gregory J. Baker, Edward Novikov, Ziyuan Zhao, Tuulia Vallius, Janae A. Davis, Jia-Ren Lin, Jeremy L. Muhlich, Elizabeth A. Mittendorf, Sandro Santagata, Jennifer L. Guerriero, and Peter K. Sorger. Quality control for single-cell analysis of high-plex tissue profiles using CyLinter. *Nature Methods*, pages 1–12, October 2024. ISSN 1548-7105. doi: 10.1038/s41592-024-02328-0.

[66] Peter Bankhead, Maurice B. Loughrey, José A. Fernández, Yvonne Dombrowski, Darragh G. McArt, Philip D. Dunne, Stephen McQuaid, Ronan T. Gray, Liam J. Murray, Helen G. Coleman, Jacqueline A. James, Manuel Salto-Tellez, and Peter W. Hamilton. QuPath: Open source software for digital pathology image analysis. *Scientific Reports*, 7(1):16878, December 2017. ISSN 2045-2322. doi: 10.1038/s41598-017-17204-5.

[67] Z. Bar-Joseph, D. K. Gifford, and T. S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17(Suppl 1):S22–S29, June 2001. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/17.suppl_1.S22.

[68] A. Barbosa, F. Paulovich, Afonso Paiva, S. Goldenstein, Fabiano Petronetto, and L. G. Nonato. Visualizing and interacting with kernelized data. *IEEE Transactions on Visualization and Computer Graphics*, 22:1314–1325, 2016. Referenced in Table C.1.

[69] Laura Barisoni, Kyle J. Lafata, Stephen M. Hewitt, Anant Madabhushi, and Ulysses G. J. Balis. Digital pathology and computational image analysis in nephropathology. *Nature Reviews Nephrology*, 16(11):669–685, November 2020. ISSN 1759-507X. doi: 10.1038/s41581-020-0321-6.

[70] Nikolas Barkas, Viktor Petukhov, Daria Nikolaeva, Yaroslav Lozinsky, Samuel Demharter, Konstantin Khodosevich, and Peter V. Kharchenko. Joint analysis of heterogeneous single-cell RNA-seq dataset collections. *Nature Methods*, 16(8): 695–698, August 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0466-z.

[71] Soumyadeep Basu, Jeroen Eggermont, Thomas Kroes, Nikolas Jorstad, Trygve Bakken, Ed Lein, Boudewijn Lelieveldt, and Thomas Höllt. Cytosplore Simian Viewer: Visual Exploration for Multi-Species Single-Cell RNA Sequencing Data. 2023.

[72] Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel W. H. Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W. Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 37(1):38–44, January 2019. ISSN 1546-1696. doi: 10.1038/nbt.4314.

[73] Richard A. Becker, William S. Cleveland, and Allan R. Wilks. Dynamic Graphics for Data Analysis. *Statistical Science*, 2(4):355–383, November 1987. ISSN 0883-4237, 2168-8745. doi: 10.1214/ss/1177013104.

[74] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. pages 585–591, 2001. Referenced in Table C.1.

[75] J. P. Benzecri. L'analyse des données. *Population*, 30(6):221–241, 1975. Referenced in Table C.1.

[76] Sébastien Besson, Roger Leigh, Melissa Linkert, Chris Allan, Jean-Marie Burel, Mark Carroll, David Gault, Riad Gozim, Simon Li, Dominik Lindner, Josh Moore, Will Moore, Petr Walczysko, Frances Wong, and Jason R Swedlow. Bringing open data to whole slide imaging. *Digital pathology : 15th European Congress, ECDP 2019, Warwick, UK, April 10-13, 2019, Proceedings. European Congress on Digital Pathology (15th : 2019 : Warwick, United Kingdom)*, 2019: 3–10, April 2019.

[77] Johanna Beyer, Jakob Troidl, Saeed Boorboor, Markus Hadwiger, Arie Kaufman, and Hanspeter Pfister. A Survey of Visualization and Analysis in High-Resolution Connectomics. *Computer Graphics Forum*, 41(3):573–607. ISSN 1467-8659. doi: 10.1111/cgf.14574.

[78] Sebastian Birk, Irene Bonafonte-Pardàs, Adib Miraki Feriz, Adam Boxall, Eneritz Agirre, Fani Memi, Anna Maguza, Anamika Yadav, Erick Armingol, Rong Fan, Gonçalo Castelo-Branco, Fabian J. Theis, Omer Ali Bayraktar, Carlos Talavera-López, and Mohammad Lotfollahi. Quantitative characterization of

cell niches in spatially resolved omics data. *Nature Genetics*, pages 1–13, March 2025. ISSN 1546-1718. doi: 10.1038/s41588-025-02120-6.

[79] Charles M. Bishop, M. Svensén, and Christopher K. I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10:215–234, 1998. Referenced in Table C.1.

[80] Sarah Black, Darci Phillips, John W Hickey, Julia Kennedy-Darling, Vishal G Venkataraaman, Nikolay Samusik, Yury Goltsev, Christian M Schürch, and Garry P Nolan. CODEX multiplexed tissue imaging with DNA-conjugated antibodies. *Nature Protocols*, 16(8):3802–3835, August 2021.

[81] Quentin Blampey, Hakim Benkirane, Nadège Bercovici, Fabrice André, and Paul-Henry Cournède. Novae: A graph-based foundation model for spatial transcriptomics data, September 2024.

[82] Vittoria Dickinson Bocchi, Paola Conforti, Elena Vezzoli, Dario Besusso, Claudio Cappadona, Tiziana Lischetti, Maura Galimberti, Valeria Ranzani, Raoul J. P. Bonnal, Marco De Simone, Grazisa Rossetti, Xiaoling He, Kenji Kamimoto, Ira Espuny-Camacho, Andrea Faedo, Federica Gervasoni, Romina Vuono, Samantha A. Morris, Jian Chen, Dan Felsenfeld, Giulio Pavesi, Roger A. Barker, Massimiliano Pagani, and Elena Cattaneo. The coding and long noncoding single-cell atlas of the developing human fetal striatum. *Science*, 372 (eabf5759):591+, May 2021. ISSN 0036-8075. doi: 10.1126/science.abf5759.

[83] Boyan Bonev, Castelo-Branco Gonçalo, Fei Chen, Simone Codeluppi, M. Ryan Corces, Jean Fan, Myriam Heiman, Kenneth Harris, Fumitaka Inoue, Manolis Kellis, Ariel Levine, Mo Lotfollahi, Chongyuan Luo, Kristen R. Maynard, Mor Nitzan, Vijay Ramani, Rahul Satijia, Lucas Schirmer, Yin Shen, Na Sun, Gilad S. Green, Fabian Theis, Xiao Wang, Joshua D. Welch, Ozgun Gokce, Genevieve Konopka, Shane Liddelow, Evan Macosko, Omer Bayraktar, Naomi Habib, and Tomasz J. Nowakowski. Opportunities and challenges of single-cell and spatially resolved genomics methods for neuroscience discovery. *Nature Neuroscience*, 27(12):2292–2309, December 2024. ISSN 1546-1726. doi: 10.1038/s41593-024-01806-0.

[84] Nicholas Borcherding and Nicholas L. Bormann. scRepertoire: An R-based toolkit for single-cell immune receptor analysis. *F1000Research*, 9:47, January 2020. ISSN 2046-1402. doi: 10.12688/f1000research.22139.1.

[85] Katy Börner, Sarah A. Teichmann, Ellen M. Quardokus, James C. Gee, Kristen Browne, David Osumi-Sutherland, Bruce W. Herr, Andreas Bueckle, Hrishikesh Paul, and Muzlifah Haniffa. Anatomical structures, cell types and

biomarkers of the Human Reference Atlas. *Nature Cell Biology*, 2021. doi: 10.1038/s41556-021-00788-6.

[86] Lutz Bornmann and Richard Williams. An evaluation of percentile measures of citation impact, and a proposal for making them better. *Scientometrics*, 124(2): 1457–1478, August 2020. ISSN 1588-2861. doi: 10.1007/s11192-020-03512-7.

[87] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12): 2301–2309, October 2011. doi: 10.1109/TVCG.2011.185.

[88] N Boukhelifa, J C Roberts, and P J Rodgers. A coordination model for exploratory multiview visualization. In *Proc. CMV*, pages 76–85, London, England, 2003. IEEE. doi: 10.1109/CMV.2003.1215005.

[89] Nadia Boukhelifa and Peter J Rodgers. A model and software system for coordinated and multiple views in exploratory visualization. *Information Visualization*, 2(4):258–269, December 2003. doi: 10.1057/palgrave.ivs.9500057.

[90] Alan P. Boyle, Sean Davis, Hennady P. Shulha, Paul Meltzer, Elliott H. Margulies, Zhiping Weng, Terrence S. Furey, and Gregory E. Crawford. High-resolution mapping and characterization of open chromatin across the genome. *Cell*, 132(2):311–322, January 2008. ISSN 1097-4172. doi: 10.1016/j.cell.2007.12.014.

[91] M. Brand. Charting a manifold. pages 961–968, 2002. Referenced in Table C.1.

[92] Emelie Braun, Miri Danan-Gotthold, Lars E. Borm, Ka Wai Lee, Elin Vinsland, Peter Lonnerberg, Lijuan Hu, Xiaofei Li, Xiaoling He, Zaneta Andrusivova, Joakim Lundeberg, Roger A. Barker, Ernest Arenas, Erik Sundstrom, and Sten Linnarsson. Comprehensive cell atlas of the first-trimester developing human brain. *Science*, 382(eadf1226):172+, October 2023. ISSN 0036-8075. doi: 10.1126/science.adf1226.

[93] Maria Brbić, Kaidi Cao, John W. Hickey, Yuqi Tan, Michael P. Snyder, Garry P. Nolan, and Jure Leskovec. Annotation of spatially resolved single-cell data with STELLAR. *Nature Methods*, pages 1–8, October 2022. ISSN 1548-7105. doi: 10.1038/s41592-022-01651-8.

[94] Danila Bredikhin, Ilia Kats, and Oliver Stegle. MUON: Multimodal omics analysis framework. *Genome Biology*, 23(1):42. ISSN 1474-760X. doi: 10.1186/s13059-021-02577-8.

[95] Matthew Brehmer and Tamara Munzner. A Multi-Level Typology of Abstract Visualization Tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, December 2013. ISSN 1077-2626. doi: 10.1109/TVCG.2013.124.

[96] M. Bronstein, A. Bronstein, Ron Kimmel, and I. Yavneh. Multigrid multidimensional scaling. *Numerical Linear Algebra with Applications*, 13, 2006. Referenced in Table C.1.

[97] Jason D. Buenrostro, Paul G. Giresi, Lisa C. Zaba, Howard Y. Chang, and William J. Greenleaf. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. *Nature Methods*, 10(12):1213–1218, December 2013. ISSN 1548-7105. doi: 10.1038/nmeth.2688.

[98] Jason D Buenrostro, Beijing Wu, Ulrike M Litzenburger, Dave Ruff, Michael L Gonzales, Michael P Snyder, Howard Y Chang, and William J Greenleaf. Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature*, 523(7561):486–490, July 2015.

[99] Roman D. Bülow, David L. Hölscher, Ivan G. Costa, and Peter Boor. Extending the landscape of omics technologies by pathomics. *npj Systems Biology and Applications*, 9(1):1–3, August 2023. ISSN 2056-7189. doi: 10.1038/s41540-023-00301-9.

[100] Charlotte Bunne, Yusuf Roohani, Yanay Rosen, Ankit Gupta, Xikun Zhang, Marcel Roed, Theo Alexandrov, Mohammed AlQuraishi, Patricia Brennan, Daniel B. Burkhardt, Andrea Califano, Jonah Cool, Abby F. Dernburg, Kirsty Ewing, Emily B. Fox, Matthias Haury, Amy E. Herr, Eric Horvitz, Patrick D. Hsu, Viren Jain, Gregory R. Johnson, Thomas Kalil, David R. Kelley, Shana O. Kelley, Anna Kreshuk, Tim Mitchison, Stephani Otte, Jay Shendure, Nicholas J. Sofroniew, Fabian Theis, Christina V. Theodoris, Srigokul Upadhyayula, Marc Valer, Bo Wang, Eric Xing, Serena Yeung-Levy, Marinka Zitnik, Theofanis Karaletsos, Aviv Regev, Emma Lundberg, Jure Leskovec, and Stephen R. Quake. How to build the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 187(25):7045–7063, December 2024. ISSN 0092-8674, 1097-4172. doi: 10.1016/j.cell.2024.11.015.

[101] M. Büttner, J. Ostner, C. L. Müller, F. J. Theis, and B. Schubert. scCODA is a Bayesian model for compositional single-cell data analysis. *Nature Communications*, 12(1):6876, November 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-27150-6.

[102] Batuhan Cakir, Martin Prete, Ni Huang, Stijn van Dongen, Pinar Pir, and Vladimir Yu Kiselev. Comparison of visualization tools for single-cell RNAseq data. *NAR Genomics and Bioinformatics*, 2(3). doi: 10.1093/nargab/lqaa052.

[103] 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(eaba7721):808+, November 2020. ISSN 0036-8075. doi: 10.1126/science.aba7721.

[104] Dylan Cashman, Mark Keller, Hyeon Jeon, Bum Chul Kwon, and Qianwen Wang. A Critical Analysis of the Usage of Dimensionality Reduction in Four Domains, March 2025.

[105] Howard Cedar and Yehudit Bergman. Linking DNA methylation and histone modification: Patterns and paradigms. *Nature Reviews Genetics*, 10(5):295–304, May 2009. ISSN 1471-0064. doi: 10.1038/nrg2540.

[106] M. Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. *Proceedings of Seventh Annual IEEE Visualization '96*, pages 127–131, 1996. Referenced in Table C.1.

[107] Joseph Chee Chang, Nathan Hahn, Yongsung Kim, Julina Coupland, Bradley Breneisen, Hannah S Kim, John Hwong, and Aniket Kittur. When the Tab Comes Due:Challenges in the Cost Structure of Browser Tab Usage. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, Yokohama Japan, May 2021. ACM. ISBN 978-1-4503-8096-6. doi: 10.1145/3411764.3445585.

[108] Joseph Chee Chang, Yongsung Kim, Victor Miller, Michael Xieyang Liu, Brad A Myers, and Aniket Kittur. Tabs.do: Task-Centric Browser Tab Management. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 663–676, Virtual Event USA, October 2021. ACM. ISBN 978-1-4503-8635-7. doi: 10.1145/3472749.3474777.

[109] Tara Chari and Lior Pachter. The specious art of single-cell genomics. *PLOS Computational Biology*, 19(8):e1011288, August 2023. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1011288.

[110] Florin Chelaru, Llewellyn Smith, Naomi Goldstein, and Héctor Corrada Bravo. Epiviz: Interactive visual analytics for functional genomics data. *Nature Methods*, 11(9):938–940, September 2014. ISSN 1548-7105. doi: 10.1038/nmeth.3038.

[111] Duo Chen, Mingwei Shao, Yi Song, Gaofei Ren, Feng Guo, Xunjie Fan, Yanyan Wang, Wei Zhang, and Guijun Qin. Single-cell RNA-seq with spatial transcriptomics to create an atlas of human diabetic kidney disease. *FASEB Journal*, 37 (e22938), June 2023. ISSN 0892-6638. doi: 10.1096/fj.202202013RR.

[112] Kok Hao Chen, Alistair N. Boettiger, Jeffrey R. Moffitt, Siyuan Wang, and Xiaowei Zhuang. Spatially resolved, highly multiplexed RNA profiling in single cells. *Science*, 348(6233), April 2015. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aaa6090.

[113] Ran Chen, Xinhuan Shu, Jiahui Chen, Di Weng, Junxiu Tang, Siwei Fu, and Yingcai Wu. Nebula: A coordinating grammar of graphics. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4127–4140, April 2022. doi: 10.1109/TVCG.2021.3076222.

[114] Song Chen, Blue B. Lake, and Kun Zhang. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nature Biotechnology*, 37(12):1452–1457, December 2019. ISSN 1546-1696. doi: 10.1038/s41587-019-0290-0.

[115] Tao Chen, Yiliang Xu, Xiaocui Xu, Jianzhang Wang, Zhiruo Qiu, Yayuan Yu, Xiaohong Jiang, Wanqi Shao, Dandan Bai, Mingzhu Wang, Shuyan Mei, Tao Cheng, Li Wu, Shaorong Gao, and Xuan Che. Comprehensive transcriptional atlas of human adenomyosis deciphered by the integration of single-cell RNA-sequencing and spatial transcriptomics. *Protein & Cell*, 15(7):530–546, May 2024. ISSN 1674-800X. doi: 10.1093/procel/pwae012.

[116] Xinyue Chen, Yin Huang, Liangfeng Huang, Ziliang Huang, Zhao-Zhe Hao, Lahong Xu, Nana Xu, Zhi Li, Yonggao Mou, Mingli Ye, Renke You, Xuegong Zhang, Sheng Liu, and Zhichao Miao. A brain cell atlas integrating single-cell transcriptomes across human brain regions. *Nature Medicine*, 2024 AUG 2 2024. ISSN 1078-8956. doi: 10.1038/s41591-024-03150-z.

[117] Yangchi Chen, M. Crawford, and Joydeep Ghosh. Improved nonlinear manifold learning for land cover classification via intelligent landmark selection. *2006 IEEE International Symposium on Geoscience and Remote Sensing*, pages 545–548, 2006. Referenced in Table C.1.

[118] Yiqun T Chen and Daniela M Witten. Selective inference for k-means clustering.

[119] Furui Cheng, Mark S Keller, Huamin Qu, Nils Gehlenborg, and Qianwen Wang. Polyphony: an interactive transfer learning framework for single-cell data analy-

sis. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):591–601, January 2023. doi: 10.1109/TVCG.2022.3209408.

[120] Benjamin Chidester, Tianming Zhou, Shahul Alam, and Jian Ma. SPICEMIX enables integrative single-cell spatial modeling of cell identity. *Nature Genetics*, 55(1):78–88, January 2023. ISSN 1546-1718. doi: 10.1038/s41588-022-01256-z.

[121] Uthsav Chitra, Brian J. Arnold, Hirak Sarkar, Cong Ma, Sereno Lopez-Darwin, Kohei Sanno, and Benjamin J. Raphael. Mapping the topography of spatial gene expression with interpretable deep learning. Preprint, Bioinformatics, October 2023.

[122] Simone Codeluppi, Lars E Borm, Amit Zeisel, Gioele La Manno, Josina A van Lunteren, Camilla I Svensson, and Sten Linnarsson. Spatial organization of the somatosensory cortex revealed by osmFISH. *Nature Methods*, 15(11):932–935, November 2018.

[123] Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. Structural Scaffolds for Citation Intent Classification in Scientific Publications. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3586–3596, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1361.

[124] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006. Referenced in Table C.1.

[125] Charles Comiter, Eeshit Dhaval Vaishnav, Metamia Ciampricotti, Bo Li, Yiming Yang, Scott J. Rodig, Madison Turner, Kathleen L. Pfaff, Judit Jané-Valbuena, Michal Slyper, Julia Waldman, Sebastian Vigneau, Jingyi Wu, Timothy R. Blosser, Åsa Segerstolpe, Daniel Abravanel, Nikil Wagle, Xiaowei Zhuang, Charles M. Rudin, Johanna Klughammer, Orit Rozenblatt-Rosen, Koseki J. Kobayash-Kirschvink, Jian Shu, and Aviv Regev. Inference of single cell profiles from histology stains with the Single-Cell omics from Histology Analysis Framework (SCHAF), March 2023.

[126] T. Cox and M. Cox. Metric multidimensional scaling. 2000. Referenced in Table C.1.

[127] Leticia Cuarental, Marta Ribagorda, Maria I. Ceballos, Aranzazu Pintor-Chocano, Sol M. Carriazo, Ana Dopazo, Enrique Vazquez, Beatriz Suarez-Alvarez, Pablo Cannata-Ortiz, Ana B. Sanz, Alberto Ortiz, and Maria D. Sanchez-Niño. The transcription factor Fosl1 preserves Klotho expression and

protects from acute kidney injury. *Kidney International*, 103(4):686–701, April 2023. ISSN 0085-2538. doi: 10.1016/j.kint.2022.11.023.

[128] Zach Cutler, Kiran Gadhave, and Alexander Lex. Trrack: A library for provenance-tracking in web-based visualizations. In *Proc. VIS*, pages 116–120, October 2020. doi: 10.1109/VIS47514.2020.00030.

[129] CZI Cell Science Program, Shibla Abdulla, Brian Aevermann, Pedro Assis, Seve Badajoz, Sidney M Bell, Emanuele Bezzi, Batuhan Cakir, Jim Chaffer, Signe Chambers, J Michael Cherry, Tiffany Chi, Jennifer Chien, Leah Dorman, Pablo Garcia-Nieto, Nayib Gloria, Mim Hastie, Daniel Hegeman, Jason Hilton, Timmy Huang, Amanda Infeld, Ana-Maria Istrate, Ivana Jelic, Kuni Katsuya, Yang Joon Kim, Karen Liang, Mike Lin, Maximilian Lombardo, Bailey Marshall, Bruce Martin, Fran McDade, Colin Megill, Nikhil Patel, Alexander Predeus, Brian Raymor, Behnam Robatmili, Dave Rogers, Erica Rutherford, Dana Sadgat, Andrew Shin, Corinn Small, Trent Smith, Prathap Sridharan, Alexander Tarashansky, Norbert Tavares, Harley Thomas, Andrew Tolopko, Meghan Urisko, Joyce Yan, Garabet Yeretssian, Jennifer Zamanian, Arathi Mani, Jonah Cool, and Ambrose Carr. CZ CELLxGENE Discover: A single-cell data platform for scalable exploration, analysis and modeling of aggregated data. *Nucleic Acids Research*, page gkae1142, November 2024. ISSN 0305-1048. doi: 10.1093/nar/gkae1142.

[130] Emma Dann, Neil C. Henderson, Sarah A. Teichmann, Michael D. Morgan, and John C. Marioni. Differential abundance testing on single-cell data using k-nearest neighbor graphs. *Nature Biotechnology*, 40(2):245–253, February 2022. ISSN 1546-1696. doi: 10.1038/s41587-021-01033-z.

[131] S. Dasgupta. Experiments with random projection. *ArXiv*, abs/1301.3849, 2000. Referenced in Table C.1.

[132] Ian H de Boer, Charles E Alpers, Evren U Azeloglu, Ulysses G J Balis, Jonathan M Barasch, Laura Barisoni, Kristina N Blank, Andrew S Bomback, Keith Brown, Pierre C Dagher, Ashveena L Dighe, Michael T Eadon, Tarek M El-Achkar, Joseph P Gaut, Nir Hacohen, Yongqun He, Jeffrey B Hodgin, Sanjay Jain, John A Kellum, Krzysztof Kiryluk, Richard Knight, Zoltan G Laszik, Chrysta Lienczewski, Laura H Mariani, Robyn L McClelland, Steven Menez, Dennis G Moledina, Sean D Mooney, John F O'Toole, Paul M Palevsky, Chirag R Parikh, Emilio D Poggio, Sylvia E Rosas, Matthew R Rosengart, Minnie M Sarwal, Jennifer A Schaub, John R Sedor, Kumar Sharma, Becky Steck, Robert D Toto, Olga G Troyanskaya, Katherine R Tuttle, Miguel A Vazquez, Sushrut S Waikar, Kayleen Williams, Francis Perry Wilson, Kun Zhang,

Ravi Iyengar, Matthias Kretzler, Jonathan Himmelfarb, and Kidney Precision Medicine Project. Rationale and design of the kidney precision medicine project. *Kidney International*, 99(3):498–510, March 2021.

[133] P. Demartines and J. Hérault. Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *IEEE transactions on neural networks*, 8 1:148–54, 1997. Referenced in Table C.1.

[134] Marie Deprez, Laure-Emmanuelle Zaragosi, Marin Truchi, Christophe Becavin, Sandra Ruiz Garcia, Marie-Jeanne Arguel, Magali Plaisant, Virginie Magnone, Kevin Lebrigand, Sophie Abelanet, Frederic Brau, Agnes Paquet, Dana Pe'er, Charles-Hugo Marquette, Sylvie Leroy, and Pascal Barbry. A single-cell atlas of the human healthy airways. *American Journal of Respiratory And Critical Care Medicine*, 202(12):1636–1645, December 2020. ISSN 1073-449X. doi: 10.1164/rccm.201911-2199OC.

[135] Anna Devor, Peter A. Bandettini, David A. Boas, James M. Bower, Richard B. Buxton, Lawrence B. Cohen, Anders M. Dale, Gaute T. Einevoll, Peter T. Fox, Maria Angela Franceschini, Karl J. Friston, James G. Fujimoto, Mark A. Geyer, Joel H. Greenberg, Eric Halgren, Matti S. Hämäläinen, Fritjof Helmchen, Bradley T. Hyman, Alan Jasanoff, Terry L. Jernigan, Lewis L. Judd, Seong-Gi Kim, David Kleinfeld, Nancy J. Kopell, Marta Kutas, Kenneth K. Kwong, Matthew E. Larkum, Eng H. Lo, Pierre J. Magistretti, Joseph B. Mandeville, Eliezer Masliah, Partha P. Mitra, William C. Mobley, Michael A. Moskowitz, Axel Nimmerjahn, John H. Reynolds, Bruce R. Rosen, Brian M. Salzberg, Chris B. Schaffer, Gabriel A. Silva, Peter T. C. So, Nicholas C. Spitzer, Roger B. Tootell, David C. Van Essen, Wim Vanduffel, Sergei A. Vinogradov, Lawrence L. Wald, Lihong V. Wang, Bruno Weber, and Arjun G. Yodh. The Challenge of Connecting the Dots in the B.R.A.I.N. *Neuron*, 80(2):270–274, October 2013. ISSN 0896-6273. doi: 10.1016/j.neuron.2013.09.008.

[136] Felipe Segato Dezem, Wani Arjumand, Hannah DuBose, Natalia Silva Morosini, and Jasmine Plummer. Spatially Resolved Single-Cell Omics: Methods, Challenges, and Future Perspectives. *Annual Review of Biomedical Data Science*, 7(1):131–153, August 2024. ISSN 2574-3414. doi: 10.1146/annurev-biodatasci-102523-103640.

[137] Song-Lin Ding, Joshua J. Royall, Susan M. Sunkin, Lydia Ng, Benjamin A. C. Facer, Phil Lesnar, Angie Guillozet-Bongaarts, Bergen McMurray, Aaron Szafer, Tim A. Dolbeare, Allison Stevens, Lee Tirrell, Thomas Benner, Shiella Caldejon, Rachel A. Dalley, Nick Dee, Christopher Lau, Julie Nyhus, Melissa Reding, Zackery L. Riley, David Sandman, Elaine Shen, Andre van der Kouwe, Ani Varjabedian, Michelle Wright, Lilla Zöllei, Chinh Dang, James A. Knowles,

Christof Koch, John W. Phillips, Nenad Sestan, Paul Wohnoutka, H. Ronald Zielke, John G. Hohmann, Allan R. Jones, Amy Bernard, Michael J. Hawrylycz, Patrick R. Hof, Bruce Fischl, and Ed S. Lein. Comprehensive cellular-resolution atlas of the adult human brain. *The Journal of Comparative Neurology*, 524(16): 3127–3481, November 2016. ISSN 1096-9861. doi: 10.1002/cne.24080.

[138] Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P. Fulco, Livnat Jerby-Arnon, Nemanja D. Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, Britt Adamson, Thomas M. Norman, Eric S. Lander, Jonathan S. Weissman, Nir Friedman, and Aviv Regev. Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens. *Cell*, 167(7):1853–1866.e17, December 2016. ISSN 0092-8674, 1097-4172. doi: 10.1016/j.cell.2016.11.038.

[139] Silvia Domcke and Jay Shendure. A reference cell tree will serve science better than a reference cell atlas. *Cell*, 186(6):1103–1114, March 2023. ISSN 0092-8674, 1097-4172. doi: 10.1016/j.cell.2023.02.016.

[140] Silvia Domcke, Andrew J. Hill, Riza M. Daza, Junyue Cao, Diana R. O'Day, Hannah A. Pliner, Kimberly A. Aldinger, Dmitry Pokholok, Fan Zhang, Jennifer H. Milbank, Michael A. Zager, Ian A. Glass, Frank J. Steemers, Dan Doherty, Cole Trapnell, Darren A. Cusanovich, and Jay Shendure. A human cell atlas of fetal chromatin accessibility. *Science*, 370(eaba7612):809+, November 2020. ISSN 0036-8075. doi: 10.1126/science.aba7612.

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

[142] D. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences of the United States of America*, 100:5591 – 5596, 2003. Referenced in Table C.1.

[143] Yvette Drabsch, Honor Hugo, Rui Zhang, Dennis H. Dowhan, Yu Rebecca Miao, Alan M. Gewirtz, Simon C. Barry, Robert G. Ramsay, and Thomas J. Gonda. Mechanism of and requirement for estrogen-regulated MYB expression in estrogen-receptor-positive breast cancer cells. *Proceedings of the National*

Academy of Sciences of the United States of America, 104(34):13762–13767, August 2007. ISSN 0027-8424. doi: 10.1073/pnas.0700104104.

[144] Caifan Du, Johanna Cohoon, Patrice Lopez, and James Howison. Softcite dataset: A dataset of software mentions in biomedical and economic research publications. *Journal of the Association for Information Science and Technology*, 72(7):870–884, 2021.

[145] Mirjana Efremova, Miquel Vento-Tormo, Sarah A. Teichmann, and Roser Vento-Tormo. CellPhoneDB: Inferring cell–cell communication from combined expression of multi-subunit ligand–receptor complexes. *Nature Protocols*, 15(4): 1484–1506, April 2020. ISSN 1750-2799. doi: 10.1038/s41596-020-0292-x.

[146] Markus Ekvall, Ludvig Bergenstråhle, Alma Andersson, Paulo Czarnewski, Johannes Olegård, Lukas Käll, and Joakim Lundeberg. Spatial landmark detection and tissue registration with deep learning. *Nature Methods*, 21(4):673–679, April 2024. ISSN 1548-7105. doi: 10.1038/s41592-024-02199-5.

[147] John Ellson, Emden R. Gansner, Eleftherios Koutsofios, Stephen C. North, and Gordon Woodhull. *Graphviz and Dynagraph — Static and Dynamic Graph Drawing Tools*, pages 127–148. Springer Berlin Heidelberg, 2004. ISBN 978-3-642-18638-7. doi: 10.1007/978-3-642-18638-7_6.

[148] Rasa Elmentaite, Cecilia Domínguez Conde, Lu Yang, and Sarah A. Teichmann. Single-cell atlases: Shared and tissue-specific cell types across human organs. *Nature Reviews Genetics*, 23(7):395–410. ISSN 1471-0064. doi: 10.1038/s41576-022-00449-w.

[149] Rasa Elmentaite, Natsuhiko Kumasaka, Kenny Roberts, Aaron Fleming, Emma Dann, Hamish W. King, Vitalii Kleshchevnikov, Monika Dabrowska, Sophie Pritchard, Liam Bolt, Sara F. Vieira, Lira Mamanova, Ni Huang, Francesca Perrone, Issac Goh Kai'En, Steven N. Lisgo, Matilda Katan, Steven Leonard, Thomas R. W. Oliver, C. Elizabeth Hook, Komal Nayak, Lia S. Campos, Cecilia Domínguez Conde, Emily Stephenson, Justin Engelbert, Rachel A. Botting, Krzysztof Polanski, Stijn van Dongen, Minal Patel, Michael D. Morgan, John C. Marioni, Omer Ali Bayraktar, Kerstin B. Meyer, Xiaoling He, Roger A. Barker, Holm H. Uhlig, Krishnaa T. Mahbubani, Kourosh Saeb-Parsy, Matthias Zilbauer, Menna R. Clatworthy, Muzlifah Haniffa, Kylie R. James, and Sarah A. Teichmann. Cells of the human intestinal tract mapped across space and time. *Nature*, 597(7875):250–255, September 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03852-1.

[150] Martin Emons, Samuel Gunz, Helena L. Crowell, Izaskun Mallona, Reinhard Furrer, and Mark D. Robinson. Pasta: Pattern Analysis for Spatial Omics Data, December 2024.

[151] Mateus Espadoto, Rafael M. Martins, Andreas Kerren, Nina S. T. Hirata, and Alexandru C. Telea. Toward a Quantitative Survey of Dimension Reduction Techniques. *IEEE Transactions on Visualization and Computer Graphics*, 27(3): 2153–2173, March 2021. ISSN 1077-2626, 1941-0506, 2160-9306. doi: 10.1109/TVCG.2019.2944182.

[152] Allison B. Esselman, Felipe A. Moser, Léonore E. M. Tideman, Lukasz G. Migas, Katerina V. Djambazova, Madeline E. Colley, Ellie L. Pingry, Nathan Heath Patterson, Melissa A. Farrow, Haichun Yang, Agnes B. Fogo, Mark de Caestecker, Raf Van de Plas, and Jeffrey M. Spraggins. *In Situ* molecular profiles of glomerular cells by integrated imaging mass spectrometry and multiplexed immunofluorescence microscopy. *Kidney International*, 107(2):332–337, February 2025. ISSN 0085-2538. doi: 10.1016/j.kint.2024.11.008.

[153] John Erol Evangelista, Zhuorui Xie, Giacomo B Marino, Nhi Nguyen, Daniel J B Clarke, and Avi Ma'ayan. Enrichr-KG: Bridging enrichment analysis across multiple libraries. *Nucleic Acids Research*, 51(W1):W168–W179, July 2023. ISSN 0305-1048. doi: 10.1093/nar/gkad393.

[154] Ugomma C. Eze, Aparna Bhaduri, Maximilian Haeussler, Tomasz J. Nowakowski, and Arnold R. Kriegstein. Single-cell atlas of early human brain development highlights heterogeneity of human neuroepithelial cells and early radial glia. *Nature Neuroscience*, 24(4):584–594, April 2021. ISSN 1097-6256. doi: 10.1038/s41593-020-00794-1.

[155] Samuel G. Fadel, Francisco M. Fatore, Felipe S. L. G. Duarte, and F. Paulovich. LoCH: A neighborhood-based multidimensional projection technique for high-dimensional sparse spaces. *Neurocomputing*, 150:546–556, 2015. Referenced in Table C.1.

[156] Christos Faloutsos and King-Ip Lin. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pages 163–174, 1995. Referenced in Table C.1.

[157] Rongxin Fang, Sebastian Preissl, Yang Li, Xiaomeng Hou, Jacinta Lucero, Xinxin Wang, Amir Motamedi, Andrew K Shiau, Xinzhu Zhou, Fangming Xie, Eran A Mukamel, Kai Zhang, Yanxiao Zhang, M Margarita Behrens, Joseph R Ecker, and Bing Ren. Comprehensive analysis of single cell ATAC-seq data with SnapATAC. *Nature Communications*, 12(1):1337, February 2021.

[158] David S. Fischer, Martin A. Villanueva, Peter S. Winter, and Alex K. Shalek. Adapting systems biology to address the complexity of human disease in the single-cell era. *Nature Reviews Genetics*, pages 1–18, March 2025. ISSN 1471-0064. doi: 10.1038/s41576-025-00821-6.

[159] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, 7(2):179–188, 1936. Referenced in Table C.1.

[160] K. Fukunaga and D. R. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, C-20:176–183, 1971. Referenced in Table C.1.

[161] Anja Füllgrabe, Nancy George, Matthew Green, Parisa Nejad, Bruce Aronow, Silvie Korena Fexova, Clay Fischer, Mallory Ann Freeberg, Laura Huerta, Norman Morrison, Richard H Scheuermann, Deanne Taylor, Nicole Vasilevsky, Laura Clarke, Nils Gehlenborg, Jim Kent, John Marioni, Sarah Teichmann, Alvis Brazma, and Irene Papatheodorou. Guidelines for reporting single-cell RNA-seq experiments. *Nature Biotechnology*, 38(12):1384–1386, December 2020.

[162] Baudat G and Anouar F. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12:2385–2404, 2000. Referenced in Table C.1.

[163] Lucy L. Gao, Jacob Bien, and Daniela Witten. Selective Inference for Hierarchical Clustering. *Journal of the American Statistical Association*, pages 1–11. ISSN 0162-1459, 1537-274X. doi: 10.1080/01621459.2022.2116331.

[164] Teng Gao, Ruslan Soldatov, Hirak Sarkar, Adam Kurkiewicz, Evan Biederstedt, Po-Ru Loh, and Peter V. Kharchenko. Haplotype-aware analysis of somatic copy number variations from single-cell transcriptomes. *Nature Biotechnology*, 41(3):417–426, March 2023. ISSN 1546-1696. doi: 10.1038/s41587-022-01468-y.

[165] Laura A Garrison, Ivan Kolesar, Ivan Viola, Helwig Hauser, and Stefan Bruckner. Trends & Opportunities in Visualization for Physiology: A Multiscale Overview. *Computer Graphics Forum*, page 35.

[166] Adam Gayoso, Zoë Steier, Romain Lopez, Jeffrey Regier, Kristopher L. Nazor, Aaron Streets, and Nir Yosef. Joint probabilistic modeling of single-cell multiomic data with totalVI. *Nature Methods*, 18(3):272–282, March 2021. ISSN 1548-7105. doi: 10.1038/s41592-020-01050-x.

[167] Nancy George, Silvie Fexova, Alfonso Munoz Fuentes, Pedro Madrigal, Yalan Bi, Haider Iqbal, Upendra Kumbham, Nadja Francesca Nolte, Lingyun Zhao, Anil S Thanki, Iris D Yu, Jose C Marugan Calles, Karoly Erdos, Liora Vilmovsky, Sandeep R Kurri, Anna Vathrakokoili-Pournara, David Osumi-Sutherland, Ananth Prakash, Shengbo Wang, Marcela K Tello-Ruiz, Sunita

Kumari, Doreen Ware, Damien Goutte-Gattat, Yanhui Hu, Nick Brown, Norbert Perrimon, Juan Antonio Vizcaíno, Tony Burdett, Sarah Teichmann, Alvis Brazma, and Irene Papatheodorou. Expression Atlas update: Insights from sequencing data at both bulk and single cell level. *Nucleic Acids Research*, 52 (D1):D107–D114, January 2024. ISSN 0305-1048. doi: 10.1093/nar/gkad1021.

[168] Graham Gibbs. *Analyzing Qualitative Data*. SAGE Publications, Ltd, 1 Oliver's Yard, 55 City Road, London England EC1Y 1SP United Kingdom, 2007. ISBN 978-0-7619-4980-0 978-1-84920-857-4. doi: 10.4135/9781849208574.

[169] Brandon Ginley, Kuang-Yu Jen, Seung Seok Han, Luís Rodrigues, Sanjay Jain, Agnes B. Fogo, Jonathan Zuckerman, Vighnesh Walavalkar, Jeffrey C. Miecznikowski, Yumeng Wen, Felicia Yen, Donghwan Yun, Kyung Chul Moon, Avi Rosenberg, Chirag Parikh, and Pinaki Sarder. Automated Computational Detection of Interstitial Fibrosis, Tubular Atrophy, and Glomerulosclerosis. *Journal of the American Society of Nephrology*, 32(4):837–850, April 2021. ISSN 1046-6673, 1533-3450. doi: 10.1681/ASN.2020050652.

[170] Debora L. Gisch, Michelle Brennan, Blue B. Lake, Jeannine Basta, Mark S. Keller, Ricardo Melo Ferreira, Shreeram Akilesh, Reetika Ghag, Charles Lu, Ying-Hua Cheng, Kimberly S. Collins, Samir V. Parikh, Brad H. Rovin, Lynn Robbins, Lisa Stout, Kimberly Y. Conklin, Dinh Diep, Bo Zhang, Amanda Knoten, Daria Barwinska, Mahla Asghari, Angela R. Sabo, Michael J. Ferkowicz, Timothy A. Sutton, Katherine J. Kelly, Ian H. De Boer, Sylvia E. Rosas, Krzysztof Kiryluk, Jeffrey B. Hodgin, Fadhl Alakwaa, Seth Winfree, Nichole Jefferson, Aydın Türkmen, Joseph P. Gaut, Nils Gehlenborg, Carrie L. Phillips, Tarek M. El-Achkar, Pierre C. Dagher, Takashi Hato, Kun Zhang, Jonathan Himmelfarb, Matthias Kretzler, Shamim Mollah, Sanjay Jain, Michael Rauchman, and Michael T. Eadon. The chromatin landscape of healthy and injured cell types in the human kidney. *Nature Communications*, 15(1):433, January 2024. ISSN 2041-1723. doi: 10.1038/s41467-023-44467-6.

[171] Michael Gleicher. Considerations for Visualizing Comparison. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):413–423, January 2018. ISSN 1941-0506. doi: 10.1109/TVCG.2017.2744199.

[172] A. Globerson and S. Roweis. Metric learning by collapsing classes. pages 451–458, 2005. Referenced in Table C.1.

[173] Issac Goh, Rachel A. Botting, Antony Rose, Simone Webb, Justin Engelbert, Yorick Gitton, Emily Stephenson, Mariana Quiroga Londono, Michael Mather, Nicole Mende, Ivan Imaz-Rosshandler, Lu Yang, Dave Horsfall, Daniela Basurto-Lozada, Nana-Jane Chipampe, Victoria Rook, Jimmy Tsz Hang Lee,

Mai-Linh Ton, Daniel Keitley, Pavel Mazin, M. S. Vijayabaskar, Rebecca Hannah, Laure Gambardella, Kile Green, Stephane Ballereau, Megumi Inoue, Elizabeth Tuck, Valentina Lorenzi, Kwasi Kwakwa, Clara Alsinet, Bayanne Olabi, Mohi Miah, Chloe Admane, Dorin-Mirel Popescu, Meghan Acres, David Dixon, Thomas Ness, Rowen Coulthard, Steven Lisgo, Deborah J. Henderson, Emma Dann, Chenqu Suo, Sarah J. Kinston, Jong-eun Park, Krzysztof Polanski, John Marioni, Stijn van Dongen, Kerstin B. Meyer, Marella de Bruijn, James Palis, Sam Behjati, Elisa Laurenti, Nicola K. Wilson, Roser Vento-Tormo, Alain Chedotal, Omer Bayraktar, Irene Roberts, Laura Jardine, Berthold Goettgens, Sarah A. Teichmann, and Muzlifah Haniffa. Yolk sac cell atlas reveals multiorgan functions during human early development. *Science*, 381(eadd7564):747+, August 2023. ISSN 0036-8075. doi: 10.1126/science.add7564.

[174] Mary J. Goldman, Brian Craft, Mim Hastie, Kristupas Repečka, Fran McDade, Akhil Kamath, Ayan Banerjee, Yunhai Luo, Dave Rogers, Angela N. Brooks, Jingchun Zhu, and David Haussler. Visualizing and interpreting cancer genomics data via the Xena platform. *Nature Biotechnology*, 38(6):675–678, June 2020. ISSN 1546-1696. doi: 10.1038/s41587-020-0546-8.

[175] Alexander N Gorban, I. Karlin, and A. Zinovyev. Constructive methods of invariant manifolds for kinetic problems. *Physics Reports*, 396:197–403, 2003. Referenced in Table C.1.

[176] Jeffrey M. Granja, M. Ryan Corces, Sarah E. Pierce, S. Tansu Bagdatli, Hani Choudhry, Howard Y. Chang, and William J. Greenleaf. ArchR is a scalable software package for integrative single-cell chromatin accessibility analysis. *Nature Genetics*, 53(3):403–411, March 2021. ISSN 1546-1718. doi: 10.1038/s41588-021-00790-6.

[177] G. Kenneth Gray, Carman Man-Chung Li, Jennifer M. Rosenbluth, Laura M. Selfors, Nomeda Girnius, Jia-Ren Lin, Ron C. J. Schackmann, Walter L. Goh, Kaitlin Moore, Hana K. Shapiro, Shaolin Mei, Kurt D'Andrea, Katherine L. Nathanson, Peter K. Sorger, Sandro Santagata, Aviv Regev, Judy E. Garber, Deborah A. Dillon, and Joan S. Brugge. A human breast atlas integrating single-cell proteomics and transcriptomics. *Developmental Cell*, 57(11):1400+, June 2022. ISSN 1534-5807. doi: 10.1016/j.devcel.2022.05.003.

[178] Saul Greenberg and Bill Buxton. Usability evaluation considered harmful (some of the time). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 111–120, New York, NY, USA, April 2008. Association for Computing Machinery. ISBN 978-1-60558-011-1. doi: 10.1145/1357054.1357074.

[179] Noah F. Greenwald, Geneva Miller, Erick Moen, Alex Kong, Adam Kagel, Thomas Dougherty, Christine Camacho Fullaway, Brianna J. McIntosh, Ke Xuan Leow, Morgan Sarah Schwartz, Cole Pavelchek, Sunny Cui, Isabella Camplisson, Omer Bar-Tal, Jaiveer Singh, Mara Fong, Gautam Chaudhry, Zion Abraham, Jackson Moseley, Shiri Warshawsky, Erin Soon, Shirley Greenbaum, Tyler Risom, Travis Hollmann, Sean C. Bendall, Leeat Keren, William Graf, Michael Angelo, and David Van Valen. Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nature Biotechnology*, 40(4):555–565, April 2022. ISSN 1546-1696. doi: 10.1038/s41587-021-01094-0.

[180] David A. Gutman, Mohammed Khalilia, Sanghoon Lee, Michael Nalisnik, Zach Mullen, Jonathan Beezley, Deepak R. Chittajallu, David Manthey, and Lee A.D. Cooper. The Digital Slide Archive: A Software Platform for Management, Integration, and Analysis of Histology for Cancer Research. *Cancer Research*, 77(21):e75–e78, November 2017. ISSN 0008-5472, 1538-7445. doi: 10.1158/0008-5472.CAN-17-0629.

[181] Christoph Hafemeister and Rahul Satija. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome Biology*, 20(1):296, December 2019. ISSN 1474-760X. doi: 10.1186/s13059-019-1874-1.

[182] Laleh Haghverdi, Aaron T. L. Lun, Michael D. Morgan, and John C. Marioni. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nature Biotechnology*, 36(5):421–427, May 2018. ISSN 1546-1696. doi: 10.1038/nbt.4091.

[183] J. B. S. Haldane. Organisers and genes. *Nature*, (3700):413, September 1940.

[184] N. Halko, P. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. 2009. Referenced in Table C.1.

[185] Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, January 2004. ISSN 1573-756X. doi: 10.1023/B:DAMI.0000005258.31418.83.

[186] Ya Han, Yuting Wang, Xin Dong, Dongqing Sun, Zhaoyang Liu, Jiali Yue, Haiyun Wang, Taiwen Li, and Chenfei Wang. TISCH2: Expanded datasets and new tools for single-cell transcriptome analyses of the tumor microenvironment. *Nucleic Acids Research*, 51(D1):D1425–D1431, January 2023. ISSN 0305-1048. doi: 10.1093/nar/gkac959.

[187] Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M. Mauck, Shiwei Zheng, Andrew Butler, Maddie J. Lee, Aaron J. Wilk, Charlotte Darby, Michael Zager, Paul Hoffman, Marlon Stoeckius, Efthymia Papalexi, Eleni P. Mimitou, Jaison Jain, Avi Srivastava, Tim Stuart, Lamar M. Fleming, Bertrand Yeung, Angela J. Rogers, Juliana M. McElrath, Catherine A. Blish, Raphael Gottardo, Peter Smibert, and Rahul Satija. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587.e29, June 2021. ISSN 00928674. doi: 10.1016/j.cell.2021.04.048.

[188] Yuhan Hao, Tim Stuart, Madeline H. Kowalski, Saket Choudhary, Paul Hoffman, Austin Hartman, Avi Srivastava, Gesmira Molla, Shaista Madad, Carlos Fernandez-Granda, and Rahul Satija. Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nature Biotechnology*, 42(2):293–304, February 2024. ISSN 1546-1696. doi: 10.1038/s41587-023-01767-y.

[189] R. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-model factor analysis. volume 16, pages 1–84, 1970. Referenced in Table C.1.

[190] Philipp Harth, Arco Bast, Jakob Troidl, Bjorge Meulemeester, Hanspeter Pfister, Johanna Beyer, Marcel Oberlaender, Hans-Christian Hege, and Daniel Baum. Rapid prototyping for coordinated views of multi-scale spatial and abstract data: A grammar-based approach. In *Proc. VCBM*, 2023. doi: 10.2312/vcbm.20231218.

[191] Austin Hartman and Rahul Satija. Comparative analysis of multiplexed in situ gene expression profiling technologies. *eLife*, 13, June 2024. doi: 10.7554/eLife.96949.1.

[192] Trevor Hastie and Werner Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989. Referenced in Table C.1.

[193] Peng He, Kyungtae Lim, Dawei Sun, Jan Patrick Pett, Quitz Jeng, Krzysztof Polanski, Ziqi Dong, Liam Bolt, Laura Richardson, Lira Mamanova, Monika Dabrowska, Anna Wilbrey-Clark, Elo Madissoon, Zewen Kelvin Tuong, Emma Dann, Chenqu Suo, Isaac Goh, Masahiro Yoshida, Marko Z. Nikolic, Sam M. Janes, Xiaoling He, Roger A. Barker, Sarah A. Teichmann, John C. Marioni, Kerstin B. Meyer, and Emma L. Rawlins. A human fetal lung cell atlas uncovers proximal-distal gradients of differentiation and key regulators of epithelial fates. *Cell*, 185(25):4841+, December 2022. ISSN 0092-8674. doi: 10.1016/j.cell.2022.11.005.

[194] Shuai He, Lin-He Wang, Yang Liu, Yi-Qi Li, Hai-Tian Chen, Jing-Hong Xu, Wan Peng, Guo-Wang Lin, Pan-Pan Wei, Bo Li, Xiaojun Xia, Dan Wang, Jin-Xin Bei, Xiaoshun He, and Zhiyong Guo. Single-cell transcriptome profiling of an adult human cell atlas of 15 major organs. *Genome Biology*, 21(294), December 2020. ISSN 1474-760X. doi: 10.1186/s13059-020-02210-0.

[195] Xiaofei He and P. Niyogi. Locality preserving projections. pages 153–160, 2003. Referenced in Table C.1.

[196] Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, and Weinan E. PaSa: An LLM Agent for Comprehensive Academic Paper Search, January 2025.

[197] Marti A. Hearst. Show It or Tell It? Text, Visualization, and Their Combination. *Communications of the ACM*, 66(10):68–75, October 2023. ISSN 0001-0782, 1557-7317. doi: 10.1145/3593580.

[198] Jeffrey Heer and Dominik Moritz. Mosaic: An Architecture for Scalable & Interoperable Data Views. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2023. ISSN 1077-2626, 1941-0506, 2160-9306. doi: 10.1109/TVCG.2023.3327189.

[199] Martin Hemberg, Federico Marini, Shila Ghazanfar, Ahmad Al Ajami, Najla Abassi, Benedict Anchang, Bérénice A Benayoun, Yue Cao, Ken Chen, Yesid Cuesta-Astroz, Zach DeBruine, Calliope A Dendrou, Iwijn De Vlaminck, Katharina Imkeller, Ilya Korsunsky, Alex R Lederer, Pieter Meysman, Clint Miller, Kerry Mullan, Uwe Ohler, Nikolaos Patikas, Jonas Schuck, Jacqueline HY Siu, Timothy J Triche, Alex Tsankov, Masanao Yajima, Jean Yang, Fabio Zanini, and Ivana Jelic. Insights, opportunities and challenges provided by large cell atlases.

[200] Lukas Herzberger, Markus Hadwiger, Robert Krueger, Peter Sorger, Hanspeter Pfister, Eduard Groeller, and Johanna Beyer. Residency octree: A hybrid approach for scalable web-based multi-volume rendering. *IEEE Transactions on Visualization and Computer Graphics (Proceedings IEEE Visualization 2023)*, Oct 2023.

[201] Lukas Heumos, Anna C. Schaar, Christopher Lance, Anastasia Litinetskaya, Felix Drost, Luke Zappia, Malte D. Lücken, Daniel C. Strobl, Juan Henao, Fabiola Curion, Herbert B. Schiller, and Fabian J. Theis. Best practices for single-cell analysis across modalities. *Nature Reviews Genetics*, 24(8):550–572, August 2023. ISSN 1471-0064. doi: 10.1038/s41576-023-00586-w.

[202] Lukas Heumos, Philipp Ehmele, Tim Treis, Julius Upmeier zu Belzen, Eljas Roellin, Lilly May, Altana Namsaraeva, Nastassya Horlava, Vladimir A. Shitov, Xinyue Zhang, Luke Zappia, Rainer Knoll, Niklas J. Lang, Leon Hetzel, Isaac Virshup, Lisa Sikkema, Fabiola Curion, Roland Eils, Herbert B. Schiller, Anne Hilgendorff, and Fabian J. Theis. An open-source framework for end-to-end analysis of electronic health record data. *Nature Medicine*, pages 1–12, September 2024. ISSN 1546-170X. doi: 10.1038/s41591-024-03214-0.

[203] Lukas Heumos, Yuge Ji, Lilly May, Tessa Green, Xinyue Zhang, Xichen Wu, Johannes Ostner, Stefan Peidli, Antonia Schumacher, Karin Hrovatin, Michaela Müller, Faye Chong, Gregor Sturm, Alejandro Tejada, Emma Dann, Mingze Dong, Mojtaba Bahrami, Ilan Gold, Sergei Rybakov, Altana Namsaraeva, Amir Moinfar, Zihe Zheng, Eljas Roellin, Isra Mekki, Chris Sander, Mohammad Lotfollahi, Herbert B. Schiller, and Fabian J. Theis. Pertpy: An end-to-end framework for perturbation analysis, August 2024.

[204] Geoffrey E. Hinton and S. Roweis. Stochastic neighbor embedding. pages 833–840, 2002. Referenced in Table C.1.

[205] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. Referenced in Table C.1.

[206] Ariel A. Hippen, Matias M. Falco, Lukas M. Weber, Erdogan Pekcan Erkan, Kaiyang Zhang, Jennifer Anne Doherty, Anna Vähärautio, Casey S. Greene, and Stephanie C. Hicks. miQC: An adaptive probabilistic framework for quality control of single-cell RNA-sequencing data. *PLOS Computational Biology*, 17 (8):e1009290, August 2021. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1009290.

[207] Karin Hrovatin, Lisa Sikkema, Vladimir A. Shitov, Graham Heimberg, Maiia Shulman, Amanda J. Oliver, Michaela F. Mueller, Ignacio L. Ibarra, Hanchen Wang, Ciro Ramírez-Suástegui, Peng He, Anna C. Schaar, Sarah A. Teichmann, Fabian J. Theis, and Malte D. Luecken. Considerations for building and using integrated single-cell atlases. *Nature Methods*, pages 1–17, December 2024. ISSN 1548-7105. doi: 10.1038/s41592-024-02532-y.

[208] Zhan Hu, Wendao Liu, Xiumeng Hua, Xiao Chen, Yuan Chang, Yiqing Hu, Zhenyu Xu, and Jiangping Song. Single-cell transcriptomic atlas of different human cardiac arteries identifies cell types associated with vascular physiology. *Arteriosclerosis Thrombosis and Vascular Biology*, 41(4):1408–1427, April 2021. ISSN 1079-5642. doi: 10.1161/ATVBAHA.120.315373.

[209] Peter J Huber. Projection pursuit. *The annals of Statistics*, pages 435–475, 1985. Referenced in Table C.1.

[210] HuBMAP Consortium, Writing Group, Michael P. Snyder, Shin Lin, Amanda Posgai, Mark Atkinson, Aviv Regev, Jennifer Rood, Orit Rozenblatt-Rosen, Leslie Gaffney, Anna Hupalowska, Rahul Satija, Nils Gehlenborg, Jay Shendure, Julia Laskin, Pehr Harbury, Nicholas A. Nystrom, Jonathan C. Silverstein, Ziv Bar-Joseph, Kun Zhang, Katy Börner, Yiing Lin, Richard Conroy, Dena Procaccini, Ananda L. Roy, Ajay Pillai, Marishka Brown, Zorina S. Galis, Caltech-UW TMC, Long Cai, Jay Shendure, Cole Trapnell, Shin Lin, Dana Jackson, Stanford-WashU TMC, Michael P. Snyder, Garry Nolan, William James Greenleaf, Yiing Lin, Sylvia Plevritis, Sara Ahadi, Stephanie A. Nevins, Hayan Lee, Christian Martijn Schuerch, Sarah Black, Vishal Gautham Venkataraaman, Ed Esplin, Aaron Horning, Amir Bahmani, UCSD TMC, Kun Zhang, Xin Sun, Sanjay Jain, James Hagood, Gloria Pryhuber, Peter Kharchenko, University of Florida TMC, Mark Atkinson, Bernd Bodenmiller, Todd Brusko, Michael Clare-Salzler, Harry Nick, Kevin Otto, Amanda Posgai, Clive Wasserfall, Marda Jorgensen, Maigan Brusko, Sergio Maffioletti, Vanderbilt University TMC, Richard M. Caprioli, Jeffrey M. Spraggins, Danielle Gutierrez, Nathan Heath Patterson, Elizabeth K. Neumann, Raymond Harris, Mark deCaestecker, Agnes B. Fogo, Raf Van De Plas, Ken Lau, California Institute of Technology TTD, Long Cai, Guo-Cheng Yuan, Qian Zhu, Ruben Dries, Harvard TTD, Peng Yin, Sinem K. Saka, Jocelyn Y. Kishi, Yu Wang, Isabel Goldaracena, Purdue TTD, Julia Laskin, DongHye Ye, Kristin E. Burnum-Johnson, Paul D. Piehowski, Charles Ansong, Ying Zhu, Stanford TTD, Pehr Harbury, Tushar Desai, Jay Mulye, Peter Chou, Monica Nagendran, HuBMAP Integration, Visualization, and Engagement (HIVE) Collaboratory: Carnegie Mellon, Tools Component, Ziv Bar-Joseph, Sarah A. Teichmann, Benedict Paten, Robert F. Murphy, Jian Ma, Vladimir Yu. Kiselev, Carl Kingsford, Allyson Ricarte, Maria Keays, Sushma A. Akoju, Matthew Ruffalo, Harvard Medical School, Tools Component, Nils Gehlenborg, Peter Kharchenko, Margaret Vella, Chuck McCallum, Indiana University Bloomington, Mapping Component, Katy Börner, Leonard E. Cross, Samuel H. Friedman, Randy Heiland, Bruce Herr, Paul Macklin, Ellen M. Quardokus, Lisel Record, James P. Sluka, Griffin M. Weber, Pittsburgh Supercomputing Center and University of Pittsburgh, Infrastructure and Engagement Component, Nicholas A. Nystrom, Jonathan C. Silverstein, Philip D. Blood, Alexander J. Ropelewski, William E. Shirey, Robin M. Scibek, University of South Dakota, Collaboration Core, Paula Mabee, W. Christopher Lenhardt, Kimberly Robasky, Stavros Michailidis, New York Genome Center, Mapping Component, Rahul Satija, John Marioni, Aviv Regev, Andrew Butler, Tim Stuart, Eyal Fisher, Shila Ghazanfar, Jennifer Rood, Leslie Gaffney, Gokcen Eraslan, Tommaso Biancalani, Eeshit D. Vaishnav, NIH HuBMAP Working Group, Richard Conroy, Dena Procaccini, Ananda Roy, Ajay Pillai, Marishka Brown, Zorina Galis, Pothur Srinivas, Aaron Pawlyk, Salvatore Sechi, Elizabeth

Wilder, and James Anderson. The human body at cellular resolution: The NIH Human Biomolecular Atlas Program. *Nature*, 574(7777):187–192, October 2019. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-019-1629-x.

[211] A. Hyvarinen. Fast ICA for noisy data using gaussian moments. *ISCAS'99. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems VLSI (Cat. No.99CH36349)*, 5:57–61 vol.5, 1999. Referenced in Table C.1.

[212] Thomas Höllt, Nicola Pezzotti, Vincent van Unen, Frits Koning, Boudewijn P.F. Lelieveldt, and Anna Vilanova. CyteGuide: Visual Guidance for Hierarchical Single-Cell Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):739–748. ISSN 1941-0506. doi: 10.1109/TVCG.2017.2744318.

[213] Ana-Maria Istrate, Donghui Li, Dario Taraborelli, Michaela Torkar, Boris Veytsman, and Ivana Williams. A large dataset of software mentions in the biomedical literature, September 2022.

[214] Hartland W. Jackson, Jana R. Fischer, Vito R. T. Zanotelli, H. Raza Ali, Robert Mechera, Savas D. Soysal, Holger Moch, Simone Muenst, Zsuzsanna Varga, Walter P. Weber, and Bernd Bodenmiller. The single-cell pathology landscape of breast cancer. *Nature*, 578(7796):615–620, February 2020. ISSN 1476-4687. doi: 10.1038/s41586-019-1876-x.

[215] Catherine P. Jayapandian, Yijiang Chen, Andrew R. Janowczyk, Matthew B. Palmer, Clarissa A. Cassol, Miroslav Sekulic, Jeffrey B. Hodgin, Jarcy Zee, Stephen M. Hewitt, John O'Toole, Paula Toro, John R. Sedor, Laura Barisoni, Anant Madabhushi, J. Sedor, K. Dell, M. Schachere, J. Negrey, K. Lemley, E. Lim, T. Srivastava, A. Garrett, C. Sethna, K. Laurent, G. Appel, M. Toledo, L. Barisoni, L. Greenbaum, C. Wang, C. Kang, S. Adler, C. Nast, J. LaPage, John H. Stroger, A. Athavale, M. Itteera, A. Neu, S. Boynton, F. Fervenza, M. Hogan, J. Lieske, V. Chernitskiy, F. Kaskel, N. Kumar, P. Flynn, J. Kopp, J. Blake, H. Trachtman, O. Zhdanova, F. Modersitzki, S. Vento, R. Lafayette, K. Mehta, C. Gadegbeku, D. Johnstone, S. Quinn-Boyle, D. Cattran, M. Hladunewich, H. Reich, P. Ling, M. Romano, A. Fornoni, C. Bidot, M. Kretzler, D. Gipson, A. Williams, J. LaVigne, V. Derebail, K. Gibson, A. Froment, S. Grubbs, L. Holzman, K. Meyers, K. Kallem, J. Lalli, K. Sambandam, Z. Wang, M. Rogers, A. Jefferson, S. Hingorani, K. Tuttle, M. Bray, M. Kelton, A. Cooper, B. Freedman, and J.J. Lin. Development and evaluation of deep learning–based segmentation of histologic structures in the kidney cortex with multiple histologic stains. *Kidney International*, 99(1):86–101, January 2021. ISSN 00852538. doi: 10.1016/j.kint.2020.07.044.

[216] R. Jenssen. Kernel entropy component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:847–860, 2010. Referenced in Table C.1.

[217] Livnat Jerby-Arnon, Parin Shah, Michael S. Cuoco, Christopher Rodman, Mei-Ju Su, Johannes C. Melms, Rachel Leeson, Abhay Kanodia, Shaolin Mei, Jia-Ren Lin, Shu Wang, Bokang Rabasha, David Liu, Gao Zhang, Claire Margolais, Orr Ashenberg, Patrick A. Ott, Elizabeth I. Buchbinder, Rizwan Haq, F. Stephen Hodi, Genevieve M. Boland, Ryan J. Sullivan, Dennie T. Frederick, Benchun Miao, Tabea Moll, Keith T. Flaherty, Meenhard Herlyn, Russell W. Jenkins, Rohit Thummalapalli, Monika S. Kowalczyk, Israel Cañadas, Bastian Schilling, Adam N. R. Cartwright, Adrienne M. Luoma, Shruti Malu, Patrick Hwu, Chantale Bernatchez, Marie-Andrée Forget, David A. Barbie, Alex K. Shalek, Itay Tirosh, Peter K. Sorger, Kai Wucherpfennig, Eliezer M. Van Allen, Dirk Schadendorf, Bruce E. Johnson, Asaf Rotem, Orit Rozenblatt-Rosen, Levi A. Garraway, Charles H. Yoon, Benjamin Izar, and Aviv Regev. A Cancer Cell Program Promotes T Cell Exclusion and Resistance to Checkpoint Blockade. *Cell*, 175(4):984–997.e24, November 2018. ISSN 0092-8674. doi: 10.1016/j.cell.2018.09.006.

[218] Paul Joia, D. Coimbra, J. Cuminato, F. Paulovich, and L. G. Nonato. Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics*, 17:2563–2571, 2011. Referenced in Table C.1.

[219] Andrew Jones, F. William Townes, Didong Li, and Barbara E. Engelhardt. Alignment of spatial genomics data using deep Gaussian processes. *Nature Methods*, 20(9):1379–1387, September 2023. ISSN 1548-7105. doi: 10.1038/s41592-023-01972-2.

[220] Robert C. Jones, Jim Karkanias, Mark A. Krasnow, Angela Oliveira Pisco, Stephen R. Quake, Julia Salzman, Nir Yosef, Bryan Bulthaup, Phillip Brown, William Harper, Marisa Hemenez, Ravikumar Ponnusamy, Ahmad Salehi, Bhavani A. Sanagavarapu, Eileen Spallino, Waldo Concepcion, Ksenia A. Aaron, James M. Gardner, Burnett Kelly, Nikole Neidlinger, Zifa Wang, Sheela Crasta, Saroja Kolluru, Maurizio Morri, Serena Y. Tan, Kyle J. Travaglini, Chenling Xu, Marcela Alcantara-Hernandez, Nicole Almanzar, Jane Antony, Benjamin Beyersdorf, Deviana Burhan, Kruti Calcuttawala, Matthew M. Carter, Charles K. F. Chan, Charles A. Chang, Stephen Chang, Alex Colville, Rebecca N. Culver, Gaetano D'Amato, Camille Ezran, Francisco X. Galdos, Astrid Gillich, William R. Goodyer, Yan Hang, Alyssa Hayashi, Sahar Houshdaran, Xianxi Huang, Juan C. Irwin, SoRi Jang, Julia Vallve Juanico, Aaron M. Kershner, Soochi Kim, Bernhard Kiss, William Kong, Maya E. Kumar, Angera H. Kuo, Rebecca Leylek, Baoxiang Li, Gabriel B. Loeb, Wan-Jin Lu, Sruthi Mantri,

Maxim Markovic, Patrick L. McAlpine, Antoine de Morree, Karim Mrouj, Shravani Mukherjee, Tyler Muser, Patrick Neuhofer, Thi D. Nguyen, Kimberly Perez, Ragini Phansalkar, Nazan Puluca, Zhen Qi, Poorvi Rao, Hayley Raquer-McKay, Nicholas Schaum, Bronwyn Scott, Bobak Seddighzadeh, Joe Segal, Sushmita Sen, Shaheen Sikandar, Sean P. Spencer, Lea C. Steffes, Varun R. Subramaniam, Aditi Swarup, Michael Swift, Will Van Treuren, Emily Trimm, Stefan Veizades, Sivakamasundari Vijayakumar, Kim Chi Vo, Sevahn K. Vorperian, Wanxin Wang, Hannah N. W. Weinstein, Juliane Winkler, Timothy T. H. Wu, Jamie Xie, Andrea R. Yung, Yue Zhang, Angela M. Detweiler, Honey Mekonen, Norma F. Neff, Rene Sit, V, Michelle Tan, Jia Yan, Gregory R. Bean, Vivek Charu, Erna Forgo, Brock A. Martin, Michael G. Ozawa, Oscar Silva, Angus Toland, Venkata N. P. Vemuri, Shaked Afik, Kyle Awayan, Olga Borisovna Botvinnik, Ashley Byrne, Michelle Chen, Roozbeh Dehghannasiri, Adam Gayoso, Alejandro A. Granados, Qiqing Li, Gita Mahmoudabadi, Aaron McGeever, Julia Eve Olivieri, Madeline Park, Neha Ravikumar, Geoff Stanley, Weilun Tan, Alexander J. Tarashansky, Rohan Vanheusden, Peter Wang, Sheng Wang, Galen Xing, Rebecca Culver, Les Dethlefsen, Po-Yi Ho, Shixuan Liu, Jonathan S. Maltzman, Ross J. Metzger, Koki Sasagawa, Rahul Sinha, Hanbing Song, Bruce Wang, Steven E. Artandi, Philip A. Beachy, Michael F. Clarke, Linda C. Giudice, Franklin W. Huang, Kerwyn Casey Huang, Juliana Idoyaga, Seung K. Kim, Mark Krasnow, Christin S. Kuo, Patricia Nguyen, Thomas A. Rando, Kristy Red-Horse, Jeremy Reiter, David A. Relman, Justin L. Sonnenburg, Albert Wu, Sean M. Wu, Tony Wyss-Coray, and Tabula Sapiens Consortium. The Tabula Sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. *Science*, 376(eabl4896):711+, May 2022. ISSN 0036-8075. doi: 10.1126/science.abl4896.

[221] C. Jutten and J. Hérault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Process.*, 24:1–10, 1991. Referenced in Table C.1.

[222] N. Kambhatla and T. Leen. Fast nonlinear dimension reduction. *IEEE International Conference on Neural Networks*, pages 1213–1218 vol.3, 1993. Referenced in Table C.1.

[223] Joyce B. Kang, Aparna Nathan, Kathryn Weinand, Fan Zhang, Nghia Millard, Laurie Rumker, D. Branch Moody, Ilya Korsunsky, and Soumya Raychaudhuri. Efficient and precise single-cell reference atlas mapping with Symphony. *Nature Communications*, 12(1):5890, October 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-25957-x.

[224] J. Karhunen. Independent component analysis: Nonlinear ICA. 2001. Referenced in Table C.1.

[225] Qifa Ke and T. Kanade. Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1: 739–746 vol. 1, 2005. Referenced in Table C.1.

[226] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. Visual Analytics: Definition, Process, and Challenges. In Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North, editors, *Information Visualization: Human-Centered Issues and Perspectives*, pages 154–175. Springer, Berlin, Heidelberg, 2008. ISBN 978-3-540-70956-5. doi: 10.1007/978-3-540-70956-5_7.

[227] Mark S. Keller, Trevor Manz, and Nils Gehlenborg. Use-Coordination: Model, Grammar, and Library for Implementation of Coordinated Multiple Views. In *2024 IEEE Visualization and Visual Analytics (VIS)*, pages 166–170, October 2024. doi: 10.1109/VIS55277.2024.00041.

[228] Mark S. Keller, Ilan Gold, Chuck McCallum, Trevor Manz, Peter V. Kharchenko, and Nils Gehlenborg. Vitessce: Integrative visualization of multimodal and spatially resolved single-cell data. *Nature Methods*, 2024 SEP 27 2024. ISSN 1548-7091. doi: 10.1038/s41592-024-02436-x.

[229] Mark S. Keller, Eric Moerth, Thomas C. Smits, Simon Warchol, Grace Guo, Qianwen Wang, Robert Krueger, Hanspeter Pfister, and Nils Gehlenborg. The State of Single-Cell Atlas Data Visualization in the Biological Literature, April 2025.

[230] W. James Kent, Charles W. Sugnet, Terrence S. Furey, Krishna M. Roskin, Tom H. Pringle, Alan M. Zahler, Haussler, and David. The Human Genome Browser at UCSC. *Genome Research*, 12(6):996–1006, June 2002. ISSN 1088-9051, 1549-5469. doi: 10.1101/gr.229102.

[231] Peter Kerpedjiev, Nezar Abdennur, Fritz Lekschas, Chuck McCallum, Kasper Dinkla, Hendrik Strobelt, Jacob M. Luber, Scott B. Ouellette, Alaleh Azhir, Nikhil Kumar, Jeewon Hwang, Soohyun Lee, Burak H. Alver, Hanspeter Pfister, Leonid A. Mirny, Peter J. Park, and Nils Gehlenborg. HiGlass: Web-based visual exploration and analysis of genome interaction maps. *Genome Biology*, 19 (1):125. ISSN 1474-760X. doi: 10.1186/s13059-018-1486-1.

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

[233] Rodney Michael Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, Miles Crawford, Doug Downey, Jason Dunkelberger, Oren Etzioni, Rob Evans, Sergey Feldman, Joseph Gorney, David W. Graham, F.Q. Hu, Regan Huff, Daniel King, Sebastian Kohlmeier, Bailey Kuehl, Michael Langan, Daniel Lin, Haokun Liu, Kyle Lo, Jaron Lochner, Kelsey MacMillan, Tyler Murray, Christopher Newell, Smita R Rao, Shaurya Rohatgi, Paul L Sayre, Zejiang Shen, Amanpreet Singh, Luca Soldaini, Shivashankar Subramanian, A. Tanaka, Alex D Wade, Linda M. Wagner, Lucy Lu Wang, Christopher Wilhelm, Caroline Wu, Jiangjiang Yang, Angele Zamarron, Madeleine van Zuylen, and Daniel S. Weld. The semantic scholar open data platform. *ArXiv*, abs/2301.10140, 2023.

[234] T. R. Knapp. Canonical correlation analysis: A general parametric significance-testing system. *Psychological Bulletin*, 85:410–416, 1978. Referenced in Table C.1.

[235] Amy J. Ko, Thomas D. LaToza, and Margaret M. Burnett. A practical guide to controlled experiments of software engineering tools with human participants. *Empirical Software Engineering*, 20(1):110–141, February 2015. ISSN 1573-7616. doi: 10.1007/s10664-013-9279-3.

[236] Jina Ko, Kilean Lucas, Rainer Kohler, Elias A. Halabi, Martin Wilkovitsch, Jonathan C. T. Carlson, and Ralph Weissleder. In Vivo Click Chemistry Enables Multiplexed Intravital Microscopy. *Advanced Science*, 9(24):2200064, August 2022. ISSN 2198-3844. doi: 10.1002/advs.202200064.

[237] Dmitry Kobak and Philipp Berens. The art of using t-SNE for single-cell transcriptomics. *Nature Communications*, 10(1):5416, November 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-13056-x.

[238] Andrew L. Koenig, Irina Shchukina, Junedh Amrute, Prabhakar S. Andhey, Konstantin Zaitsev, Lulu Lai, Geetika Bajpai, Andrea Bredemeyer, Gabriella Smith, Cameran Jones, Emily Terrebonne, Stacey L. Rentschler, Maxim N. Artyomov, and Kory J. Lavine. Single-cell transcriptomics reveals cell-type-specific diversification in human heart failure. *Nature Cardiovascular Research*, 1(3):263–280, March 2022. ISSN 2731-0590. doi: 10.1038/s44161-022-00028-6.

[239] Kofi, P., Adragni, R., and Dennis Cook. Sufficient dimension reduction and prediction in regression. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367:4385 – 4405, 2009. Referenced in Table C.1.

[240] André Kohn, Dominik Moritz, Mark Raasveldt, Hannes Mühleisen, and Thomas Neumann. DuckDB-wasm: Fast analytical processing for the web. *Proceedings of the VLDB Endowment*, 15(12):3574–3577, August 2022. ISSN 2150-8097. doi: 10.14778/3554821.3554847.

[241] 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, December 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0619-0.

[242] B. Kozlíková, M. Krone, M. Falk, N. Lindow, M. Baaden, D. Baum, I. Viola, J. Parulek, and H.-C. Hege. Visualization of Biomolecular Structures: State of the Art Revisited. *Computer Graphics Forum*, 36(8):178–204. ISSN 1467-8659. doi: 10.1111/cgf.13072.

[243] M. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *Aiche Journal*, 37:233–243, 1991. Referenced in Table C.1.

[244] J. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964. Referenced in Table C.1.

[245] Tapsi Kumar, Kevin Nee, Runmin Wei, Siyuan He, Quy H. Nguyen, Shanshan Bai, Kerrigan Blake, Maren Pein, Yanwen Gong, Emi Sei, Min Hu, Anna K. Casasent, Aatish Thennavan, Jianzhuo Li, Tuan Tran, Ken Chen, Benedikt Nilges, Nachiket Kashikar, Oliver Braubach, Bassem Ben Cheikh, Nadya Nikulina, Hui Chen, Mediget Teshome, Brian Menegaz, Huma Javaid, Chandandeep Nagi, Jessica Montalvan, Tatyana Lev, Sharmila Mallya, Delia F. Tifrea, Robert Edwards, Erin Lin, Ritesh Parajuli, Summer Hanson, Sebastian Winocour, Alastair Thompson, Bora Lim, Devon A. Lawson, Kai Kessenbrock, and Nicholas Navin. A spatially resolved single-cell genomic atlas of the adult human breast. *Nature*, 620(7972):181–191, August 2023. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-023-06252-9.

[246] Gioele La Manno, Ruslan Soldatov, Amit Zeisel, Emelie Braun, Hannah Hochgerner, Viktor Petukhov, Katja Lidschreiber, Maria E. Kastriti, Peter Lönnerberg, Alessandro Furlan, Jean Fan, Lars E. Borm, Zehua Liu, David van Bruggen, Jimin Guo, Xiaoling He, Roger Barker, Erik Sundström, Gonçalo

Castelo-Branco, Patrick Cramer, Igor Adameyko, Sten Linnarsson, and Peter V. Kharchenko. RNA velocity of single cells. *Nature*, 560(7719):494–498, August 2018. ISSN 1476-4687. doi: 10.1038/s41586-018-0414-6.

[247] Alexander Lachmann, Zhuorui Xie, and Avi Ma'ayan. blitzGSEA: Efficient computation of gene set enrichment analysis through gamma distribution approximation. *Bioinformatics*, 38(8):2356–2357, April 2022. ISSN 1367-4803. doi: 10.1093/bioinformatics/btac076.

[248] David Lähnemann, Johannes Köster, Ewa Szczurek, Davis J. McCarthy, Stephanie C. Hicks, Mark D. Robinson, Catalina A. Vallejos, Kieran R. Campbell, Niko Beerenwinkel, Ahmed Mahfouz, Luca Pinello, Pavel Skums, Alexandros Stamatakis, Camille Stephan-Otto Attolini, Samuel Aparicio, Jasmijn Baaijens, Marleen Balvert, Buys de Barbanson, Antonio Cappuccio, Giacomo Corleone, Bas E. Dutilh, Maria Florescu, Victor Guryev, Rens Holmer, Katharina Jahn, Thamar Jessurun Lobo, Emma M. Keizer, Indu Khatri, Szymon M. Kielbasa, Jan O. Korbel, Alexey M. Kozlov, Tzu-Hao Kuo, Boudewijn P.F. Lelieveldt, Ion I. Mandoiu, John C. Marioni, Tobias Marschall, Felix Mölder, Amir Niknejad, Alicja Rączkowska, Marcel Reinders, Jeroen de Ridder, Antoine-Emmanuel Saliba, Antonios Somarakis, Oliver Stegle, Fabian J. Theis, Huan Yang, Alex Zelikovsky, Alice C. McHardy, Benjamin J. Raphael, Sohrab P. Shah, and Alexander Schönhuth. Eleven grand challenges in single-cell data science. *Genome Biology*, 21(1):31, February 2020. ISSN 1474-760X. doi: 10.1186/s13059-020-1926-6.

[249] Yiwei Lai, Ignacio Ramirez-Pardo, Joan Isern, Juan An, Eusebio Perdiguero, Antonio L. Serrano, Jinxiu Li, Esther Garcia-Dominguez, Jessica Segales, Pengcheng Guo, Vera Lukesova, Eva Andres, Jing Zuo, Yue Yuan, Chuanyu Liu, Jose Vina, Julio Domenech-Fernandez, Mari Carmen Gomez-Cabrera, Yancheng Song, Longqi Liu, Xun Xu, Pura Munoz-Canoves, and Miguel A. Esteban. Multimodal cell atlas of the ageing human skeletal muscle. *Nature*, 2024 APR 22 2024. ISSN 0028-0836. doi: 10.1038/s41586-024-07348-6.

[250] Blue B. Lake, Rajasree Menon, Seth Winfree, Qiwen Hu, Ricardo Melo Ferreira, Kian Kalhor, Daria Barwinska, Edgar A. Otto, Michael Ferkowicz, Dinh Diep, Nongluk Plongthongkum, Amanda Knoten, Sarah Urata, Laura H. Mariani, Abhijit S. Naik, Sean Eddy, Bo Zhang, Yan Wu, Diane Salamon, James C. Williams, Xin Wang, Karol S. Balderrama, Paul J. Hoover, Evan Murray, Jamie L. Marshall, Teia Noel, Anitha Vijayan, Austin Hartman, Fei Chen, Sushrut S. Waikar, Sylvia E. Rosas, Francis P. Wilson, Paul M. Palevsky, Krzysztof Kiryluk, John R. Sedor, Robert D. Toto, Chirag R. Parikh, Eric H. Kim, Rahul Satija, Anna Greka, Evan Z. Macosko, Peter Kharchenko, V,

Joseph P. Gaut, Jeffrey B. Hodgin, Michael T. Eadon, Pierre C. Dagher, Tarek M. El-Achkar, Kun Zhang, Matthias Kretzler, Sanjay Jain, Richard Knight, Stewart H. Lecker, Isaac Stillman, Afolarin A. Amodu, Titlayo Ilori, Shana Maikhor, Insa Schmidt, Gearoid M. McMahon, Astrid Weins, Nir Hacohen, Lakeshia Bush, Agustin Gonzalez-Vicente, Jonathan Taliercio, John O'toole, Emilio Poggio, Leslie Cooperman, Stacey Jolly, Leal Herlitz, Jane Nguyen, Ellen Palmer, Dianna Sendrey, Kassandra Spates-Harden, Paul Appelbaum, Jonathan M. Barasch, Andrew S. Bomback, Vivette D. D'Agati, Karla Mehl, Pietro A. Canetta, Ning Shang, Olivia Balderes, Satoru Kudose, Laura Barisoni, Theodore Alexandrov, Yinghua Cheng, Kenneth W. Dunn, Katherine J. Kelly, Timothy A. Sutton, Yumeng M. Wen, Celia P. Corona-Villalobos, Steven Menez, Avi Rosenberg, Mohammed Atta, Camille Johansen, Jennifer Sun, Neil Roy, Mark Williams, Evren U. Azeloglu, Cijang He, Ravi Iyengar, Jens Hansen, Yuguang Xiong, Brad Rovin, Samir Parikh, Sethu M. Madhavan, Christopher R. Anderton, Ljiljana Pasa-Tolic, Dusan Velickovic, Olga Troyanskaya, Rachel Sealfon, Katherine R. Tuttle, Zoltan G. Laszik, Garry Nolan, Minnie Sarwal, Kavya Anjani, Tara Sigdel, Heather Ascani, Ulysses G. J. Balis, Chrysta Lienczewski, Becky Steck, Yougqun He, Jennifer Schaub, Victoria M. Blanc, Raghavan Murugan, Parmjeet Randhawa, Matthew Rosengart, Mitchell Tublin, Tina Vita, John A. Kellum, Daniel E. Hall, Michele M. Elder, James Winters, Matthew Gilliam, Charles E. Alpers, Kristina N. Blank, Jonas Carson, Ian H. De Boer, Ashveena L. Dighe, Jonathan Himmelfarb, Sean D. Mooney, Stuart Shankland, Kayleen Williams, Christopher Park, Frederick Dowd, Robyn L. McClelland, Stephen Daniel, Andrew N. Hoofnagle, Adam Wilcox, Shweta Bansal, Kumar Sharma, Manjeri Venkatachalam, Guanshi Zhang, Annapurna Pamreddy, Vijaykumar R. Kakade, Dennis Moledina, Melissa M. Shaw, Ugochukwu Ugwuowo, Tanima Arora, Joseph Ardayfio, Jack Bebiak, Keith Brown, Catherine E. Campbell, John Saul, Anna Shpigel, Christy Stutzke, Robert Koewler, Taneisha Campbell, Lynda Hayashi, Nichole Jefferson, Roy Pinkeney, Glenda Roberts, V, and KPMP Consortium. An atlas of healthy and injured cell states and niches in the human kidney. *Nature*, 619(7970):585+, July 2023. ISSN 0028-0836. doi: 10.1038/s41586-023-05769-3.

[251] Heidi Lam, Melanie Tory, and Tamara Munzner. Bridging from Goals to Tasks with Design Study Analysis Reports. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):435–445, January 2018. ISSN 1941-0506. doi: 10.1109/TVCG.2017.2744319.

[252] Michael R. Lamprecht, David M. Sabatini, and Anne E. Carpenter. CellProfiler: Free, Versatile Software for Automated Biological Image Analysis. *BioTechniques*, 42(1):71–75, January 2007. ISSN 0736-6205, 1940-9818. doi: 10.2144/000112257.

[253] Jan Lause, Philipp Berens, and Dmitry Kobak. Analytic Pearson residuals for normalization of single-cell RNA-seq UMI data. *Genome Biology*, 22(1):258, September 2021. ISSN 1474-760X. doi: 10.1186/s13059-021-02451-7.

[254] Jan Lause, Philipp Berens, and Dmitry Kobak. The art of seeing the elephant in the room: 2D embeddings of single-cell data do make sense. *PLOS Computational Biology*, 20(10):e1012403, October 2024. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1012403.

[255] Neil D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. pages 329–336, 2003. Referenced in Table C.1.

[256] Daniel D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999. Referenced in Table C.1.

[257] Nicole Yee Shin Lee, Mengwei Li, Kok Siong Ang, and Jinmiao Chen. Establishing a human bone marrow single cell reference atlas to study ageing and diseases. *Frontiers in Immunology*, 14(1127879), March 2023. ISSN 1664-3224. doi: 10.3389/fimmu.2023.1127879.

[258] Woochan Lee, Seyoon Lee, Jung-Ki Yoon, Dakyung Lee, Yuri Kim, Yeon Bi Han, Rokhyun Kim, Sungji Moon, Young Jun Park, Kyunghyuk Park, Bukyoung Cha, Jaeyong Choi, Juhyun Kim, Na-young Ha, Kwhanmien Kim, Sukki Cho, Nam-Hyuk Cho, Tushar J. Desai, Jin-Haeng Chung, Joo-Hyeon Lee, and Jong-Il Kim. A single-cell atlas of in vitro multiculture systems uncovers the in vivo lineage trajectory and cell state in the human lung. *Experimental and Molecular Medicine*, 55(8):1831–1842, August 2023. ISSN 1226-3613. doi: 10.1038/s12276-023-01076-z.

[259] J. Leeuw. Applications of convex analysis to multidimensional scaling. 2000. Referenced in Table C.1.

[260] Fritz Lekschas, Xinyi Zhou, Wei Chen, Nils Gehlenborg, Benjamin Bach, and Hanspeter Pfister. A generic framework and library for exploration of small multiples through interactive piling. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):358–368, 2 2021. doi: 10.1109/TVCG.2020.3028948.

[261] Ernst Lengyel, Yan Li, Melanie Weigert, Lisha Zhu, Heather Eckart, Melissa Javellana, Sarah Ackroyd, Jason Xiao, Susan Olalekan, Dianne Glass, Shilpa Iyer, Rahul Krishnan, Agnes Julia Bilecz, Ricardo Lastra, Mengjie Chen, and Anindita Basu. A molecular atlas of the human postmenopausal fallopian tube and ovary from single-cell RNA and ATAC sequencing. *Cell Reports*, 41 (111838), December 2022. ISSN 2211-1247. doi: 10.1016/j.celrep.2022.111838.

[262] S. Lespinats, Michaël Aupetit, and A. Meyer-Bäse. ClassiMap: A new dimension reduction technique for exploratory data analysis of labeled data. *Int. J. Pattern Recognit. Artif. Intell.*, 29:1551008:1–1551008:45, 2015. Referenced in Table C.1.

[263] Bo Li, Joshua Gould, Yiming Yang, Siranush Sarkizova, Marcin Tabaka, Orr Ashenberg, Yanay Rosen, Michal Slyper, Monika S Kowalczyk, Alexandra-Chloé Villani, Timothy Tickle, Nir Hacohen, Orit Rozenblatt-Rosen, and Aviv Regev. Cumulus provides cloud-based data analysis for large-scale single-cell and single-nucleus RNA-seq. *Nature Methods*, 17(8):793–798, July 2020.

[264] Haoyang Li, Juexiao Zhou, Zhongxiao Li, Siyuan Chen, Xingyu Liao, Bin Zhang, Ruochi Zhang, Yu Wang, Shiwei Sun, and Xin Gao. A comprehensive benchmarking with practical guidelines for cellular deconvolution of spatial transcriptomics. *Nature Communications*, 14(1):1548, March 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-37168-7.

[265] Mengwei Li, Kok Siong Ang, Brian Teo, Uddamvathanak Rom, Minh N Nguyen, Sebastian Maurer-Stroh, and Jinmiao Chen. Rediscovering publicly available single-cell data with the DISCO platform. *Nucleic Acids Research*, page gkae1108, November 2024. ISSN 0305-1048. doi: 10.1093/nar/gkae1108.

[266] Yang Eric Li, Sebastian Preissl, Michael Miller, Nicholas D. Johnson, Zihan Wang, Henry Jiao, Chenxu Zhu, Zhaoning Wang, Yang Xie, Olivier Poirion, Colin Kern, Antonio Pinto-Duarte, Wei Tian, Kimberly Siletti, Nora Emerson, Julia Osteen, Jacinta Lucero, Lin Lin, Qian Yang, Quan Zhu, Nathan Zemke, Sarah Espinoza, Anna Marie Yanny, Julie Nyhus, Nick Dee, Tamara Casper, Nadiya Shapovalova, Daniel Hirschstein, Rebecca D. Hodge, Sten Linnarsson, Trygve Bakken, Boaz Levi, C. Dirk Keene, Jingbo Shang, Ed Lein, Allen Wang, M. Margarita Behrens, Joseph R. Ecker, and Bing Ren. A comparative atlas of single-cell chromatin accessibility in the human brain. *Science*, 382(6667): eadf7044, October 2023. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.adf7044.

[267] Zhijian Li, Zain M.Patel, Dongyuan Song, Guanao Yan, Jingyi Jessica Li, and Luca Pinello. Benchmarking computational methods to identify spatially variable genes and peaks, December 2023.

[268] Qingnan Liang, Xuesen Cheng, Jun Wang, Leah Owen, Akbar Shakoor, John L. Lillvis, Charles Zhang, Michael Farkas, Ivana K. Kim, Yumei Li, Margaret DeAngelis, and Rui Chen. A multi-omics atlas of the human retina at single-cell resolution. *Cell Genomics*, 3(100298), June 2023. ISSN 2666-979X. doi: 10.1016/j.xgen.2023.100298.

[269] Mingfeng Liao, Yang Liu, Jing Yuan, Yanling Wen, Gang Xu, Juanjuan Zhao, Lin Cheng, Jinxiu Li, Xin Wang, Fuxiang Wang, Lei Liu, Ido Amit, Shuye Zhang, and Zheng Zhang. Single-cell landscape of bronchoalveolar immune cells in patients with COVID-19. *Nature Medicine*, 26(6):842–844, June 2020. ISSN 1546-170X. doi: 10.1038/s41591-020-0901-9.

[270] Melissa Linkert, Curtis T Rueden, Chris Allan, Jean-Marie Burel, Will Moore, Andrew Patterson, Brian Loranger, Josh Moore, Carlos Neves, Donald Macdonald, Aleksandra Tarkowska, Caitlin Sticco, Emma Hill, Mike Rossner, Kevin W Eliceiri, and Jason R Swedlow. Metadata matters: Access to image data in the real world. *Journal of Cell Biology*, 189(5):777–782, May 2010.

[271] Tao Liu, Jorge A. Ortiz, Len Taing, Clifford A. Meyer, Bernett Lee, Yong Zhang, Hyunjin Shin, Swee S. Wong, Jian Ma, Ying Lei, Utz J. Pape, Michael Poidinger, Yiwen Chen, Kevin Yeung, Myles Brown, Yaron Turpaz, and X. Shirley Liu. Cistrome: An integrative platform for transcriptional regulation studies. *Genome Biology*, 12(8):R83, August 2011. ISSN 1474-760X. doi: 10.1186/gb-2011-12-8-r83.

[272] Michael A. Lodato, Rachel E. Rodin, Craig L. Bohrson, Michael E. Coulter, Alison R. Barton, Minseok Kwon, Maxwell A. Sherman, Carl M. Vitzthum, Lovelace J. Luquette, Chandri N. Yandava, Pengwei Yang, Thomas W. Chittenden, Nicole E. Hatem, Steven C. Ryu, Mollie B. Woodworth, Peter J. Park, and Christopher A. Walsh. Aging and neurodegeneration are associated with increased mutations in single human neurons. *Science*, 359(6375):555–559, February 2018. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aao4426.

[273] John Lonsdale, Jeffrey Thomas, Mike Salvatore, Rebecca Phillips, Edmund Lo, Saboor Shad, Richard Hasz, Gary Walters, Fernando Garcia, Nancy Young, Barbara Foster, Mike Moser, Ellen Karasik, Bryan Gillard, Kimberley Ramsey, Susan Sullivan, Jason Bridge, Harold Magazine, John Syron, Johnelle Fleming, Laura Siminoff, Heather Traino, Maghboeba Mosavel, Laura Barker, Scott Jewell, Dan Rohrer, Dan Maxim, Dana Filkins, Philip Harbach, Eddie Cortadillo, Bree Berghuis, Lisa Turner, Eric Hudson, Kristin Feenstra, Leslie Sobin, James Robb, Phillip Branton, Greg Korzeniewski, Charles Shive, David Tabor, Liqun Qi, Kevin Groch, Sreenath Nampally, Steve Buia, Angela Zimmerman, Anna Smith, Robin Burges, Karna Robinson, Kim Valentino, Deborah Bradbury, Mark Cosentino, Norma Diaz-Mayoral, Mary Kennedy, Theresa Engel, Penelope Williams, Kenyon Erickson, Kristin Ardlie, Wendy Winckler, Gad Getz, David DeLuca, Daniel MacArthur, Manolis Kellis, Alexander Thomson, Taylor Young, Ellen Gelfand, Molly Donovan, Yan Meng, George Grant, Deborah Mash, Yvonne Marcus, Margaret Basile, Jun Liu, Jun Zhu, Zhidong

Tu, Nancy J. Cox, Dan L. Nicolae, Eric R. Gamazon, Hae Kyung Im, Anuar Konkashbaev, Jonathan Pritchard, Matthew Stevens, Timothèe Flutre, Xiaoquan Wen, Emmanouil T. Dermitzakis, Tuuli Lappalainen, Roderic Guigo, Jean Monlong, Michael Sammeth, Daphne Koller, Alexis Battle, Sara Mostafavi, Mark McCarthy, Manual Rivas, Julian Maller, Ivan Rusyn, Andrew Nobel, Fred Wright, Andrey Shabalin, Mike Feolo, Nataliya Sharopova, Anne Sturcke, Justin Paschal, James M. Anderson, Elizabeth L. Wilder, Leslie K. Derr, Eric D. Green, Jeffery P. Struewing, Gary Temple, Simona Volpi, Joy T. Boyer, Elizabeth J. Thomson, Mark S. Guyer, Cathy Ng, Assya Abdallah, Deborah Colantuoni, Thomas R. Insel, Susan E. Koester, A. Roger Little, Patrick K. Bender, Thomas Lehner, Yin Yao, Carolyn C. Compton, Jimmie B. Vaught, Sherilyn Sawyer, Nicole C. Lockhart, Joanne Demchok, and Helen F. Moore. The Genotype-Tissue Expression (GTEx) project. *Nature Genetics*, 45(6):580–585, June 2013. ISSN 1546-1718. doi: 10.1038/ng.2653.

[274] Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, December 2018. ISSN 1548-7105. doi: 10.1038/s41592-018-0229-2.

[275] Mohammad Lotfollahi, Mohsen Naghipourfar, Malte D. Luecken, Matin Khajavi, Maren Büttner, Marco Wagenstetter, Žiga Avsec, Adam Gayoso, Nir Yosef, Marta Interlandi, Sergei Rybakov, Alexander V. Misharin, and Fabian J. Theis. Mapping single-cell data to reference atlases by transfer learning. *Nature Biotechnology*, 40(1):121–130, January 2022. ISSN 1546-1696. doi: 10.1038/s41587-021-01001-7.

[276] Ming Y. Lu, Bowen Chen, Drew F. K. Williamson, Richard J. Chen, Melissa Zhao, Aaron K. Chow, Kenji Ikemura, Ahrong Kim, Dimitra Pouli, Ankush Patel, Amr Soliman, Chengkuan Chen, Tong Ding, Judy J. Wang, Georg Gerber, Ivy Liang, Long Phi Le, Anil V. Parwani, Luca L. Weishaupt, and Faisal Mahmood. A multimodal generative AI copilot for human pathology. *Nature*, 634(8033):466–473, October 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07618-3.

[277] Nicholas Lucarelli, Brandon Ginley, Jarcy Zee, Sayat Mimar, Anindya S. Paul, Sanjay Jain, Seung Seok Han, Luis Rodrigues, Tezcan Ozrazgat-Baslanti, Michelle L. Wong, Girish Nadkarni, William L. Clapp, Kuang-Yu Jen, and Pinaki Sarder. Correlating Deep Learning-Based Automated Reference Kidney Histomorphometry with Patient Demographics and Creatinine. *Kidney360*, 4(12):1726, December 2023. ISSN 2641-7650. doi: 10.34067/KID.0000000000000299.

[278] Malte D Luecken and Fabian J Theis. Current best practices in single-cell RNA-seq analysis: A tutorial. *Molecular Systems Biology*, 15(6):e8746, June 2019. ISSN 1744-4292. doi: 10.15252/msb.20188746.

[279] Malte D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, M. Interlandi, M. F. Mueller, D. C. Strobl, L. Zappia, M. Dugas, M. Colomé-Tatché, and Fabian J. Theis. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, 19(1):41–50. ISSN 1548-7105. doi: 10.1038/s41592-021-01336-8.

[280] Brendon Lutnick, David Manthey, Jan U. Becker, Brandon Ginley, Katharina Moos, Jonathan E. Zuckerman, Luis Rodrigues, Alexander J. Gallan, Laura Barisoni, Charles E. Alpers, Xiaoxin X. Wang, Komuraiah Myakala, Bryce A. Jones, Moshe Levi, Jeffrey B. Kopp, Teruhiko Yoshida, Jarcy Zee, Seung Seok Han, Sanjay Jain, Avi Z. Rosenberg, Kuang Yu Jen, and Pinaki Sarder. A user-friendly tool for cloud-based whole slide image segmentation with examples from renal histopathology. *Communications Medicine*, 2(1):1–15, August 2022. ISSN 2730-664X. doi: 10.1038/s43856-022-00138-z.

[281] Sehi L'Yi, Qianwen Wang, Fritz Lekschas, and Nils Gehlenborg. Gosling: A Grammar-based Toolkit for Scalable and Interactive Genomics Data Visualization. *IEEE transactions on visualization and computer graphics*, 28(1):140–150, January 2022. ISSN 1077-2626. doi: 10.1109/TVCG.2021.3114876.

[282] Sehi L'Yi, Mark S. Keller, Ariaki Dandawate, Len Taing, Chen-Hao Chen, Myles Brown, Clifford A. Meyer, and Nils Gehlenborg. Cistrome Explorer: An interactive visual analysis tool for large-scale epigenomic data. *Bioinformatics*, 39(2), February 2023. ISSN 1367-4803. doi: 10.1093/bioinformatics/btad018.

[283] Sehi L'Yi, Astrid van den Brandt, Etowah Adams, Huyen N. Nguyen, and Nils Gehlenborg. Learnable and Expressive Visualization Authoring Through Blended Interfaces. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):459–469, January 2025. ISSN 1941-0506. doi: 10.1109/TVCG.2024.3456598.

[284] Jun Ma, Ronald Xie, Shamini Ayyadhury, Cheng Ge, Anubha Gupta, Ritu Gupta, Song Gu, Yao Zhang, Gihun Lee, Joonkee Kim, Wei Lou, Haofeng Li, Eric Upschulte, Timo Dickscheid, José Guilherme de Almeida, Yixin Wang, Lin Han, Xin Yang, Marco Labagnara, Vojislav Gligorovski, Maxime Scheder, Sahand Jamal Rahi, Carly Kempster, Alice Pollitt, Leon Espinosa, Tâm Mignot, Jan Moritz Middeke, Jan-Niklas Eckardt, Wangkai Li, Zhaoyang Li, Xiaochen Cai, Bizhe Bai, Noah F. Greenwald, David Van Valen, Erin Weisbart, Beth A. Cimini, Trevor Cheung, Oscar Brück, Gary D. Bader, and

Bo Wang. The multimodality cell segmentation challenge: Toward universal solutions. *Nature Methods*, 21(6):1103–1113, June 2024. ISSN 1548-7105. doi: 10.1038/s41592-024-02233-6.

[285] L. Maaten. An introduction to dimensionality reduction using matlab. 2007. Referenced in Table C.1.

[286] Evan Z. Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R. Bialas, Nolan Kamitaki, Emily M. Martersteck, John J. Trombetta, David A. Weitz, Joshua R. Sanes, Alex K. Shalek, Aviv Regev, and Steven A. McCarroll. Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell*, 161 (5):1202–1214, May 2015. ISSN 0092-8674. doi: 10.1016/j.cell.2015.05.002.

[287] Clarence K. Mah, Noorsher Ahmed, Nicole A. Lopez, Dylan C. Lam, Avery Pong, Alexander Monell, Colin Kern, Yuanyuan Han, Gino Prasad, Anthony J. Cesnik, Emma Lundberg, Quan Zhu, Hannah Carter, and Gene W. Yeo. Bento: A toolkit for subcellular analysis of spatial transcriptomics data. *Genome Biology*, 25(1):82, April 2024. ISSN 1474-760X. doi: 10.1186/s13059-024-03217-7.

[288] Trevor Manz, Ilan Gold, Nathan Heath Patterson, Chuck McCallum, Mark S. Keller, Bruce W. Herr, Katy Börner, Jeffrey M. Spraggins, and Nils Gehlenborg. Viv: Multiscale visualization of high-resolution multiplexed bioimaging data on the web. *Nature Methods*, pages 1–2. ISSN 1548-7105. doi: 10.1038/s41592-022-01482-7.

[289] Trevor Manz, Nezar Abdennur, and Nils Gehlenborg. Anywidget: Reusable widgets for interactive analysis and visualization in computational notebooks. *Journal of Open Source Software*, 9(102):6939, October 2024. ISSN 2475-9066. doi: 10.21105/joss.06939.

[290] Trevor Manz, Fritz Lekschas, Evan Greene, Greg Finak, and Nils Gehlenborg. A General Framework for Comparing Embedding Visualizations Across Class-Label Hierarchies. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):283–293, January 2025. ISSN 1941-0506. doi: 10.1109/TVCG.2024.3456370.

[291] Luca Marconato, Giovanni Palla, Kevin A. Yamauchi, Isaac Virshup, Elyas Heidari, Tim Treis, Wouter-Michiel Vierdag, Marcella Toth, Sonja Stockhaus, Rahul B. Shrestha, Benjamin Rombaut, Lotte Pollaris, Laurens Lehner, Harald Vöhringer, Ilia Kats, Yvan Saeys, Sinem K. Saka, Wolfgang Huber, Moritz Gerstung, Josh Moore, Fabian J. Theis, and Oliver Stegle. SpatialData: An open and universal data framework for spatial omics. *Nature Methods*, pages 1–5. ISSN 1548-7105. doi: 10.1038/s41592-024-02212-x.

[292] Luca Marconato, Wouter-Michiel Vierdag, Kevin Yamauchi, Clarence Mah, Helena Crowell, Quentin Blampey, Benjamin Rombaut, Mark S. Keller, Lotte Pollaris, Estella Yixing Dong, Sonja Stockhaus, Laurens Lehner, Chiara Schiller, Grzegorz Bokota, Tong Li, Miguel Ibarra, Artur Manukyan, Dario Righelli, Louise Deconick, and Vincent J. Carey. 1st SpatialData Developer Workshop, December 2024.

[293] Laura D. Martens, David S. Fischer, Vicente A. Yépez, Fabian J. Theis, and Julien Gagneur. Modeling fragment counts improves single-cell ATAC-seq analysis. *Nature Methods*, 21(1):28–31, January 2024. ISSN 1548-7105. doi: 10.1038/s41592-023-02112-6.

[294] Allen R. Martin and Matthew O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proc. VIS*, pages 271–278, October 1995. doi: 10.1109/VISUAL.1995.485139.

[295] Vivien Marx. Method of the Year: Spatially resolved transcriptomics. *Nature Methods*, 18(1):9–14, January 2021. ISSN 1548-7105. doi: 10.1038/s41592-020-01033-y.

[296] Ramon Massoni-Badosa, Sergio Aguilar-Fernandez, Juan C. Nieto, Paula Soler-Vila, Marc Elosua-Bayes, Domenica Marchese, Marta Kulis, Amaia Vilas-Zornoza, Marco Matteo Buhler, Sonal Rashmi, Clara Alsinet, Ginevra Caratu, Catia Moutinho, Sara Ruiz, Patricia Lorden, Giulia Lunazzi, Dolors Colomer, Gerard Frigola, Will Blevins, Lucia Romero-Rivero, Victor Jimenez-Martinez, Anna Vidal, Judith Mateos-Jaimez, Alba Maiques-Diaz, Sara Ovejero, Jerome Moreaux, Sara Palomino, David Gomez-Cabrero, Xabier Agirre, Marc A. Weniger, Hamish W. King, Lucy C. Garner, Federico Marini, Francisco Javier Cervera-Paz, Peter M. Baptista, Isabel Vilaseca, Cecilia Rosales, Silvia Ruiz-Gaspa, Benjamin Talks, Keval Sidhpura, Anna Pascual-Reguant, Anja E. Hauser, Muzlifah Haniffa, Felipe Prosper, Ralf Kueppers, Ivo Glynne Gut, Elia Campo, Jose Ignacio Martin-Subero, and Holger Heyn. An atlas of cells in the human tonsil. *Immunity*, 57(2), February 2024. ISSN 1074-7613. doi: 10.1016/j.immuni.2024.01.006.

[297] Kristen R. Maynard, Leonardo Collado-Torres, Lukas M. Weber, Cedric Uytingco, Brianna K. Barry, Stephen R. Williams, Joseph L. Catallini, Matthew N. Tran, Zachary Besich, Madhavi Tippani, Jennifer Chew, Yifeng Yin, Joel E. Kleinman, Thomas M. Hyde, Nikhil Rao, Stephanie C. Hicks, Keri Martinowich, and Andrew E. Jaffe. Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nature Neuroscience*, 24(3):425–436. ISSN 1546-1726. doi: 10.1038/s41593-020-00787-0.

[298] Paolo Mazzarello. A unifying concept: The history of cell theory. *Nature Cell Biology*, 1(1):E13–E15, May 1999. ISSN 1476-4679. doi: 10.1038/8964.

[299] John Alan McDonald, Werner Stuetzle, and Andreas Buja. Painting multiple views of complex objects. *ACM SIGPLAN Notices*, 25(10):245–257, September 1990. ISSN 0362-1340. doi: 10.1145/97946.97975.

[300] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, September 2020. Referenced in Table C.1.

[301] Roger McLendon, Allan Friedman, Darrell Bigner, Erwin G. Van Meir, Daniel J. Brat, Gena M. Mastrogianakis, Jeffrey J. Olson, Tom Mikkelsen, Norman Lehman, Ken Aldape, W. K. Alfred Yung, Oliver Bogler, Scott VandenBerg, Mitchel Berger, Michael Prados, Donna Muzny, Margaret Morgan, Steve Scherer, Aniko Sabo, Lynn Nazareth, Lora Lewis, Otis Hall, Yiming Zhu, Yanru Ren, Omar Alvi, Jiqiang Yao, Alicia Hawes, Shalini Jhangiani, Gerald Fowler, Anthony San Lucas, Christie Kovar, Andrew Cree, Huyen Dinh, Jireh Santibanez, Vandita Joshi, Manuel L. Gonzalez-Garay, Christopher A. Miller, Aleksandar Milosavljevic, Larry Donehower, David A. Wheeler, Richard A. Gibbs, Kristian Cibulskis, Carrie Sougnez, Tim Fennell, Scott Mahan, Jane Wilkinson, Liuda Ziaugra, Robert Onofrio, Toby Bloom, Rob Nicol, Kristin Ardlie, Jennifer Baldwin, Stacey Gabriel, Eric S. Lander, Li Ding, Robert S. Fulton, Michael D. McLellan, John Wallis, David E. Larson, Xiaoqi Shi, Rachel Abbott, Lucinda Fulton, Ken Chen, Daniel C. Koboldt, Michael C. Wendl, Rick Meyer, Yuzhu Tang, Ling Lin, John R. Osborne, Brian H. Dunford-Shore, Tracie L. Miner, Kim Delehaunty, Chris Markovic, Gary Swift, William Courtney, Craig Pohl, Scott Abbott, Amy Hawkins, Shin Leong, Carrie Haipek, Heather Schmidt, Maddy Wiechert, Tammi Vickery, Sacha Scott, David J. Dooling, Asif Chinwalla, George M. Weinstock, Elaine R. Mardis, Richard K. Wilson, Gad Getz, Wendy Winckler, Roel G. W. Verhaak, Michael S. Lawrence, Michael O'Kelly, Jim Robinson, Gabriele Alexe, Rameen Beroukhim, Scott Carter, Derek Chiang, Josh Gould, Supriya Gupta, Josh Korn, Craig Mermel, Jill Mesirov, Stefano Monti, Huy Nguyen, Melissa Parkin, Michael Reich, Nicolas Stransky, Barbara A. Weir, Levi Garraway, Todd Golub, Matthew Meyerson, Lynda Chin, Alexei Protopopov, Jianhua Zhang, Ilana Perna, Sandy Aronson, Narayan Sathiamoorthy, Georgia Ren, Jun Yao, W. Ruprecht Wiedemeyer, Hyunsoo Kim, Sek Won Kong, Yonghong Xiao, Isaac S. Kohane, Jon Seidman, Peter J. Park, Raju Kucherlapati, Peter W. Laird, Leslie Cope, James G. Herman, Daniel J. Weisenberger, Fei Pan, David Van Den Berg, Leander Van Neste, Joo Mi Yi, Kornel E. Schuebel, Stephen B. Baylin, Devin M. Absher, Jun Z. Li, Audrey Southwick, Shannon Brady, Amita Aggarwal, Tisha Chung, Gavin Sher-

lock, James D. Brooks, Richard M. Myers, Paul T. Spellman, Elizabeth Purdom, Lakshmi R. Jakkula, Anna V. Lapuk, Henry Marr, Shannon Dorton, Yoon Gi Choi, Ju Han, Amrita Ray, Victoria Wang, Steffen Durinck, Mark Robinson, Nicholas J. Wang, Karen Vranizan, Vivian Peng, Eric Van Name, Gerald V. Fontenay, John Ngai, John G. Conboy, Bahram Parvin, Heidi S. Feiler, Terence P. Speed, Joe W. Gray, Cameron Brennan, Nicholas D. Socci, Adam Olshen, Barry S. Taylor, Alex Lash, Nikolaus Schultz, Boris Reva, Yevgeniy Antipin, Alexey Stukalov, Benjamin Gross, Ethan Cerami, Wei Qing Wang, Li-Xuan Qin, Venkatraman E. Seshan, Liliana Villafania, Magali Cavatore, Laetitia Borsu, Agnes Viale, William Gerald, Chris Sander, Marc Ladanyi, Charles M. Perou, D. Neil Hayes, Michael D. Topal, Katherine A. Hoadley, Yuan Qi, Sai Balu, Yan Shi, Junyuan Wu, Robert Penny, Michael Bittner, Troy Shelton, Elizabeth Lenkiewicz, Scott Morris, Debbie Beasley, Sheri Sanders, Ari Kahn, Robert Sfeir, Jessica Chen, David Nassau, Larry Feng, Erin Hickey, Jinghui Zhang, John N. Weinstein, Anna Barker, Daniela S. Gerhard, Joseph Vockley, Carolyn Compton, Jim Vaught, Peter Fielding, Martin L. Ferguson, Carl Schaefer, Subhashree Madhavan, Kenneth H. Buetow, Francis Collins, Peter Good, Mark Guyer, Brad Ozenberger, Jane Peterson, Elizabeth Thomson, The Cancer Genome Atlas Research Network, Tissue source sites: Duke University Medical School, Emory University, Henry Ford Hospital, MD Anderson Cancer Center, University of California San Francisco, Genome sequencing centres: Baylor College of Medicine, Broad Institute of MIT and Harvard, Washington University in St Louis, Cancer genome characterization centres: Broad Institute/Dana-Farber Cancer Institute, Harvard Medical School/Dana-Farber Cancer Institute, Johns Hopkins/University of Southern California, HudsonAlpha Institute/Stanford University, Lawrence Berkeley National Laboratory, Memorial Sloan-Kettering Cancer Center, Chapel Hill University of North Carolina, Biospecimen Core Resource, Data Coordinating Center, Project teams: National Cancer Institute, and National Human Genome Research Institute. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216):1061–1068, October 2008. ISSN 1476-4687. doi: 10.1038/nature07385.

[302] Andrew P. McMahon, Bruce J. Aronow, Duncan R. Davidson, Jamie A. Davies, Kevin W. Gaido, Sean Grimmond, James L. Lessard, Melissa H. Little, S. Steven Potter, Elizabeth L. Wilder, Pumin Zhang, and GUDMAP project. GUDMAP: The genitourinary developmental molecular anatomy project. *Journal of the American Society of Nephrology: JASN*, 19(4):667–671, April 2008. ISSN 1533-3450. doi: 10.1681/ASN.2007101078.

[303] Colin Megill, Bruce Martin, Charlotte Weaver, Sidney Bell, Lia Prins, Seve Badajoz, Brian McCandless, Angela Oliveira Pisco, Marcus Kinsella, Fiona

Griffin, Justin Kiggins, Genevieve Haliburton, Arathi Mani, Matthew Weiden, Madison Dunitz, Maximilian Lombardo, Timmy Huang, Trent Smith, Signe Chambers, Jeremy Freeman, Jonah Cool, and Ambrose Carr. Cellxgene: A performant, scalable exploration platform for high dimensional sparse matrices. page 2021.04.05.438318. doi: 10.1101/2021.04.05.438318.

[304] Shenglin Mei, Qian Qin, Qiu Wu, Hanfei Sun, Rongbin Zheng, Chongzhi Zang, Muyuan Zhu, Jiaxin Wu, Xiaohui Shi, Len Taing, Tao Liu, Myles Brown, Clifford A. Meyer, and X. Shirley Liu. Cistrome Data Browser: A data portal for ChIP-Seq and chromatin accessibility data in human and mouse. *Nucleic Acids Research*, 45(D1):D658–D662, January 2017. ISSN 0305-1048. doi: 10.1093/nar/gkw983.

[305] Zhen Miao and Junhyong Kim. Uniform quantification of single-nucleus ATAC-seq data with Paired-Insertion Counting (PIC) and a model-based insertion rate estimator. *Nature Methods*, 21(1):32–36, January 2024. ISSN 1548-7105. doi: 10.1038/s41592-023-02103-7.

[306] Alistair Miles, John Kirkham, Martin Durant, James Bourbeau, Tarik Onalan, Joe Hamman, Zain Patel, Shikharsg, Matthew Rocklin, Raphael Dussin, Vincent Schut, Elliott Sales De Andrade, Ryan Abernathey, Charles Noyes, Sbalmer, Pyup.Io Bot, Tommy Tran, Stephan Saalfeld, Justin Swaney, Josh Moore, Joe Jevnik, Jerome Kelleher, Jan Funke, George Sakkis, Chris Barnes, and Anderson Banihirwe. Zarr-developers/zarr-python: V2.4.0. Zenodo, January 2020.

[307] R. Minghim, F. Paulovich, and A. Lopes. Content-based text mapping using multi-dimensional projections for exploration of document collections. 6060, 2006. Referenced in Table C.1.

[308] Felix Mölder, Kim Philipp Jablonski, Brice Letcher, Michael B Hall, Christopher H Tomkins-Tinch, Vanessa Sochat, Jan Forster, Soohyun Lee, Sven O Twardziok, Alexander Kanitz, Andreas Wilm, Manuel Holtgrewe, Sven Rahmann, Sven Nahnsen, and Johannes Köster. Sustainable data analysis with Snakemake. *F1000Res.*, 10:33, January 2021.

[309] Aboozar Monavarfeshan, Wenjun Yan, Christian Pappas, Kenechukwu A. Odenigbo, Zhigang He, Ayellet V. Segre, Tave van Zyl, Gregory S. Hageman, and Joshua R. Sanes. Transcriptomic analysis of the ocular posterior segment completes a cell atlas of the human eye. *Proceedings of the National Academy of Sciences of the United States of America*, 120(e2306153120), August 2023. ISSN 0027-8424. doi: 10.1073/pnas.2306153120.

[310] Kevin R. Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B. Burkhardt, William S. Chen, Kristina Yim, Antonia van den Elzen, Matthew J. Hirn, Ronald R. Coifman, Natalia B. Ivanova, Guy Wolf, and Smita Krishnaswamy. Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology*, 37(12):1482–1492, December 2019. ISSN 1546-1696. doi: 10.1038/s41587-019-0336-3.

[311] Josh Moore, Chris Allan, Sébastien Besson, Jean-Marie Burel, Erin Diel, David Gault, Kevin Kozlowski, Dominik Lindner, Melissa Linkert, Trevor Manz, Will Moore, Constantin Pape, Christian Tischer, and Jason R. Swedlow. OME-NGFF: A next-generation file format for expanding bioimaging data-access strategies. *Nature Methods*, 18(12):1496–1498, . ISSN 1548-7091, 1548-7105. doi: 10.1038/s41592-021-01326-w.

[312] Josh Moore, Melissa Linkert, Colin Blackburn, Mark Carroll, Richard K. Ferguson, Helen Flynn, Kenneth Gillen, Roger Leigh, Simon Li, Dominik Lindner, William J. Moore, Andrew J. Patterson, Blazej Pindelski, Balaji Ramalingam, Emil Rozbicki, Aleksandra Tarkowska, Petr Walczysko, Chris Allan, Jean-Marie Burel, and Jason Swedlow. OMERO and Bio-Formats 5: Flexible access to large bioimaging datasets at scale. In *Medical Imaging 2015: Image Processing*, volume 9413, pages 37–42. SPIE, . doi: 10.1117/12.2086370.

[313] Josh Moore, Daniela Basurto-Lozada, Sébastien Besson, John Bogovic, Jordão Bragantini, Eva M. Brown, Jean-Marie Burel, Xavier Casas Moreno, Gustavo de Medeiros, Erin E. Diel, David Gault, Satrajit S. Ghosh, Ilan Gold, Yaroslav O. Halchenko, Matthew Hartley, Dave Horsfall, Mark S. Keller, Mark Kittisopikul, Gabor Kovacs, Aybüke Küpcü Yoldaş, Koji Kyoda, Albane le Tournoulx de la Villegeorges, Tong Li, Prisca Liberali, Dominik Lindner, Melissa Linkert, Joel Lüthi, Jeremy Maitin-Shepard, Trevor Manz, Luca Marconato, Matthew McCormick, Merlin Lange, Khaled Mohamed, William Moore, Nils Norlin, Wei Ouyang, Bugra Özdemir, Giovanni Palla, Constantin Pape, Lucas Pelkmans, Tobias Pietzsch, Stephan Preibisch, Martin Prete, Norman Rzepka, Sameeul Samee, Nicholas Schaub, Hythem Sidky, Ahmet Can Solak, David R. Stirling, Jonathan Striebel, Christian Tischer, Daniel Toloudis, Isaac Virshup, Petr Walczysko, Alan M. Watson, Erin Weisbart, Frances Wong, Kevin A. Yamauchi, Omer Bayraktar, Beth A. Cimini, Nils Gehlenborg, Muzlifah Haniffa, Nathan Hotaling, Shuichi Onami, Loic A. Royer, Stephan Saalfeld, Oliver Stegle, Fabian J. Theis, and Jason R. Swedlow. OME-Zarr: A cloud-optimized bioimaging file format with international community support. *Histochemistry and Cell Biology*, 160(3):223–251, September 2023. ISSN 1432-119X. doi: 10.1007/s00418-023-02209-1.

[314] Eric Mörth, Cydney Nielsen, Morgan Turner, Liu Xianhao, Mark S Keller, Johanna Beyer, Hanspeter Pfister, Chen Zhu-Tian, and Nils Gehlenborg. A Mixed Reality and 2D Display Hybrid Approach for Visual Analysis of 3D Tissue Maps, April 2024.

[315] Matthew P. Mulè, Andrew J. Martins, and John S. Tsang. Normalizing and denoising protein expression data from droplet-based single cell profiling. *Nature Communications*, 13(1):2099, April 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-29356-8.

[316] Tamara Munzner. A Nested Model for Visualization Design and Validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, November 2009. ISSN 1941-0506. doi: 10.1109/TVCG.2009.111.

[317] Christoph Muus, Malte D. Luecken, Gökcen Eraslan, Lisa Sikkema, Avinash Waghray, Graham Heimberg, Yoshihiko Kobayashi, Eeshit Dhaval Vaishnav, Ayshwarya Subramanian, Christopher Smillie, Karthik A. Jagadeesh, Elizabeth Thu Duong, Evgenij Fiskin, Elena Torlai Triglia, Meshal Ansari, Peiwen Cai, Brian Lin, Justin Buchanan, Sijia Chen, Jian Shu, Adam L. Haber, Hattie Chung, Daniel T. Montoro, Taylor Adams, Hananeh Aliee, Samuel J. Allon, Zaneta Andrusivova, Ilias Angelidis, Orr Ashenberg, Kevin Bassler, Christophe Bécavin, Inbal Benhar, Joseph Bergenstråhle, Ludvig Bergenstråhle, Liam Bolt, Emelie Braun, Linh T. Bui, Steven Callori, Mark Chaffin, Evgeny Chichelnitskiy, Joshua Chiou, Thomas M. Conlon, Michael S. Cuoco, Anna S. E. Cuomo, Marie Deprez, Grant Duclos, Denise Fine, David S. Fischer, Shila Ghazanfar, Astrid Gillich, Bruno Giotti, Joshua Gould, Minzhe Guo, Austin J. Gutierrez, Arun C. Habermann, Tyler Harvey, Peng He, Xiaomeng Hou, Lijuan Hu, Yan Hu, Alok Jaiswal, Lu Ji, Peiyong Jiang, Theodoros S. Kapellos, Christin S. Kuo, Ludvig Larsson, Michael A. Leney-Greene, Kyungtae Lim, Monika Litviňuková, Leif S. Ludwig, Soeren Lukassen, Wendy Luo, Henrike Maatz, Elo Madissoon, Lira Mamanova, Kasidet Manakongtreecheep, Sylvie Leroy, Christoph H. Mayr, Ian M. Mbano, Alexi M. McAdams, Ahmad N. Nabhan, Sarah K. Nyquist, Lolita Penland, Olivier B. Poirion, Sergio Poli, Can-Can Qi, Rachel Queen, Daniel Reichart, Ivan Rosas, Jonas C. Schupp, Conor V. Shea, Xingyi Shi, Rahul Sinha, Rene V. Sit, Kamil Slowikowski, Michal Slyper, Neal P. Smith, Alex Sountoulidis, Maximilian Strunz, Travis B. Sullivan, Dawei Sun, Carlos Talavera-López, Peng Tan, Jessica Tantivit, Kyle J. Travaglini, Nathan R. Tucker, Katherine A. Vernon, Marc H. Wadsworth, Julia Waldman, Xiuting Wang, Ke Xu, Wenjun Yan, William Zhao, and Carly G. K. Ziegler. Single-cell meta-analysis of SARS-CoV-2 entry genes across tissues and demographics. *Nature Medicine*, 27(3):546–559, March 2021. ISSN 1546-170X. doi: 10.1038/s41591-020-01227-z.

[318] Alexandra Naba. Mechanisms of assembly and remodelling of the extracellular matrix. *Nature Reviews Molecular Cell Biology*, 25(11):865–885, November 2024. ISSN 1471-0080. doi: 10.1038/s41580-024-00767-3.

[319] Ashwin Narayan, Bonnie Berger, and Hyunghoon Cho. Assessing single-cell transcriptomic variability through density-preserving data visualization. *Nature Biotechnology*, 39(6):765–774, June 2021. ISSN 1546-1696. doi: 10.1038/s41587-020-00801-7.

[320] Luka Nicin, Sam Michael Schroeter, Simone Franziska Glaser, Ralf Schulze-Bruening, Minh-Duc Pham, Susanne S. Hille, Michail Yekelchyk, Badder Kattih, Wesley Tyler Abplanalp, Lukas Tombor, Oliver J. Mueller, Thomas Braun, Benjamin Meder, Christoph Reich, Mani Arsalan, Tomas Holubec, Thomas Walther, Fabian Emrich, Jaya Krishnan, Andreas M. Zeiher, David John, and Stefanie Dimmeler. A human cell atlas of the pressure-induced hypertrophic heart. *Nature Cardiovascular Research*, 1(2):174+, February 2022. doi: 10.1038/s44161-022-00019-7.

[321] Thierry M. Nordmann, Holly Anderton, Akito Hasegawa, Lisa Schweizer, Peng Zhang, Pia-Charlotte Stadler, Ankit Sinha, Andreas Metousis, Florian A. Rosenberger, Maximilian Zwiebel, Takashi K. Satoh, Florian Anzengruber, Maximilian T. Strauss, Maria C. Tanzer, Yuki Saito, Ting Gong, Marvin Thielert, Haruna Kimura, Natasha Silke, Edwin H. Rodriguez, Gaetana Restivo, Hong Ha Nguyen, Annette Gross, Laurence Feldmeyer, Lukas Joerg, Mitchell P. Levesque, Peter J. Murray, Saskia Ingen-Housz-Oro, Andreas Mund, Riichiro Abe, John Silke, Chao Ji, Lars E. French, and Matthias Mann. Spatial proteomics identifies JAKi as treatment for a lethal skin disease. *Nature*, 635(8040):1001–1009, November 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-08061-0.

[322] Chris North and Ben Shneiderman. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In *Proc. AVI*, pages 128–135, New York, NY, USA, 2000. Association for Computing Machinery. doi: 10.1145/345513.345282.

[323] Noa Novershtern, Aravind Subramanian, Lee N. Lawton, Raymond H. Mak, W. Nicholas Haining, Marie E. McConkey, Naomi Habib, Nir Yosef, Cindy Y. Chang, Tal Shay, Garrett M. Frampton, Adam C. B. Drake, Ilya Leskov, Bjorn Nilsson, Fred Preffer, David Dombkowski, John W. Evans, Ted Liefeld, John S. Smutko, Jianzhu Chen, Nir Friedman, Richard A. Young, Todd R. Golub, Aviv Regev, and Benjamin L. Ebert. Densely interconnected transcriptional circuits control cell states in human hematopoiesis. *Cell*, 144(2):296–309, January 2011. ISSN 0092-8674. doi: 10.1016/j.cell.2011.01.004.

[324] Anna Nowogrodzki. The Cell Seeker. *Nature*, 547(7661):24–26, July 2017. ISSN 1476-4687. doi: 10.1038/547024a.

[325] S. Nusrat, T. Harbig, and N. Gehlenborg. Tasks, Techniques, and Tools for Genomic Data Visualization. *Computer Graphics Forum*, 38(3):781–805. ISSN 1467-8659. doi: 10.1111/cgf.13727.

[326] Michelli F. Oliveira, Juan P. Romero, Meii Chung, Stephen Williams, Andrew D. Gottscho, Anushka Gupta, Susan E. Pilipauskas, Syrus Mohabbat, Nandhini Raman, David Sukovich, David Patterson, Visium HD Development Team, and Sarah E. B. Taylor. Characterization of immune cell populations in the tumor microenvironment of colorectal cancer using high definition spatial profiling, June 2024.

[327] Luz D. Orozco, Hsu-Hsin Chen, Christian Cox, Kenneth J. Katschke, Jr., Rommel Arceo, Carmina Espiritu, Patrick Caplazi, Sarajane Saturnio Nghiem, Ying-Jiun Chen, Zora Modrusan, Amy Dressen, Leonard D. Goldstein, Christine Clarke, Tushar Bhangale, Brian Yaspan, Marion Jeanne, Michael J. Townsend, Menno van Lookeren Campagne, and Jason A. Hackney. Integration of eQTL and a single-cell atlas in the human eye identifies causal genes for age-related macular degeneration. *Cell Reports*, 30(4):1246+, January 2020. ISSN 2211-1247. doi: 10.1016/j.celrep.2019.12.082.

[328] Mahyar Osanlouy, Anita Bandrowski, Bernard de Bono, David Brooks, Antonino M. Cassarà, Richard Christie, Nazanin Ebrahimi, Tom Gillespie, Jeffrey S. Grethe, Leonardo A. Guercio, Maci Heal, Mabelle Lin, Niels Kuster, Maryann E. Martone, Esra Neufeld, David P. Nickerson, Elias G. Soltani, Susan Tappan, Joost B. Wagenaar, Katie Zhuang, and Peter J. Hunter. The SPARC DRC: Building a Resource for the Autonomic Nervous System Community. *Frontiers in Physiology*, 12:693735, 2021. ISSN 1664-042X. doi: 10.3389/fphys.2021.693735.

[329] Johannes Ostner, Salomé Carcy, and Christian L. Müller. tascCODA: Bayesian Tree-Aggregated Analysis of Compositional Amplicon and Single-Cell Data. *Frontiers in Genetics*, 12:766405, December 2021. ISSN 1664-8021. doi: 10.3389/fgene.2021.766405.

[330] Mineto Ota, Jeffrey P. Spence, Tony Zeng, Emma Dann, Alexander Marson, and Jonathan K. Pritchard. Causal modeling of gene effects from regulators to programs to traits: Integration of genetic associations and Perturb-seq, January 2025.

[331] John F. Ouyang, Uma S. Kamaraj, Elaine Y. Cao, and Owen J. L. Rackham. ShinyCell: Simple and sharable visualization of single-cell gene expression data.

Bioinformatics (Oxford, England), 37(19):3374–3376, September 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab209.

[332] Leif Oxburgh, Thomas J. Carroll, Ondine Cleaver, Daniel R. Gossett, Deborah K. Hoshizaki, Jeffrey A. Hubbell, Benjamin D. Humphreys, Sanjay Jain, Jan Jensen, David L. Kaplan, Carl Kesselman, Christian J. Ketchum, Melissa H. Little, Andrew P. McMahon, Stuart J. Shankland, Jason R. Spence, M. Todd Valerius, Jason A. Wertheim, Oliver Wessely, Ying Zheng, and Iain A. Drummond. (Re)Building a Kidney. *Journal of the American Society of Nephrology: JASN*, 28(5):1370–1378, May 2017. ISSN 1533-3450. doi: 10.1681/ASN.2016101077.

[333] C. Paige and M. Saunders. Towards a generalized singular value decomposition. *SIAM Journal on Numerical Analysis*, 18:398–405, 1981. Referenced in Table C.1.

[334] Giovanni Palla, David S. Fischer, Aviv Regev, and Fabian J. Theis. Spatial components of molecular tissue biology. *Nature Biotechnology*, pages 1–11, . ISSN 1546-1696. doi: 10.1038/s41587-021-01182-1.

[335] Giovanni Palla, Hannah Spitzer, Michal Klein, David Fischer, Anna Christina Schaar, Louis Benedikt Kuemmerle, Sergei Rybakov, Ignacio L. Ibarra, Olle Holmberg, Isaac Virshup, Mohammad Lotfollahi, Sabrina Richter, and Fabian J. Theis. Squidpy: A scalable framework for spatial omics analysis. *Nature Methods*, pages 1–8, . ISSN 1548-7105. doi: 10.1038/s41592-021-01358-2.

[336] Andrew Palmer, Prasad Phapale, Ilya Chernyavsky, Regis Lavigne, Dominik Fay, Artem Tarasov, Vitaly Kovalev, Jens Fuchser, Sergey Nikolenko, Charles Pineau, Michael Becker, and Theodore Alexandrov. FDR-controlled metabolite annotation for high-resolution imaging mass spectrometry. *Nature Methods*, 14 (1):57–60, January 2017. ISSN 1548-7105. doi: 10.1038/nmeth.4072.

[337] Lu Pan, Paolo Parini, Roman Tremmel, Joseph Loscalzo, Volker M. Lauschke, Bradley A. Maron, Paola Paci, Ingemar Ernberg, Nguan Soon Tan, Zehuan Liao, Weiyao Yin, Sundararaman Rengarajan, Xuexin Li, and The SCA Consortium. Single Cell Atlas: A single-cell multi-omics human cell encyclopedia. *Genome Biology*, 25(1):104, April 2024. ISSN 1474-760X. doi: 10.1186/s13059-024-03246-2.

[338] Efthymia Papalexi and Rahul Satija. Single-cell RNA sequencing to explore immune cell heterogeneity. *Nature Reviews Immunology*, 18(1):35–45, January 2018. ISSN 1474-1741. doi: 10.1038/nri.2017.76.

[339] Jong-Eun Park, Rachel A. Botting, Cecilia Dominguez Conde, Dorin-Mirel Popescu, Marieke Lavaert, Daniel J. Kunz, Issac Goh, Emily Stephenson, Roberta Ragazzini, Elizabeth Tuck, Anna Wilbrey-Clark, Kenny Roberts, Veronika R. Kedlian, John R. Ferdinand, Xiaoling He, Simone Webb, Daniel Maunder, Niels Vandamme, Krishnaa T. Mahbubani, Krzysztof Polanski, Lira Mamanova, Liam Bolt, David Crossland, Fabrizio de Rita, Andrew Fuller, Andrew Filby, Gary Reynolds, David Dixon, Kourosh Saeb-Parsy, Steven Lisgo, Deborah Henderson, Roser Vento-Tormo, Omer A. Bayraktar, Roger A. Barker, Kerstin B. Meyer, Yvan Saeys, Paola Bonfanti, Sam Behjati, Menna R. Clatworthy, Tom Taghon, Muzlifah Haniffa, and Sarah A. Teichmann. A cell atlas of human thymic development defines T cell repertoire formation. *Science*, 367 (eaay3224):868+, February 2020. ISSN 0036-8075. doi: 10.1126/science.aay3224.

[340] Peter J. Park. ChIP–seq: Advantages and challenges of a maturing technology. *Nature Reviews Genetics*, 10(10):669–680, October 2009. ISSN 1471-0064. doi: 10.1038/nrg2641.

[341] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 721, pages 8026–8037. Curran Associates Inc., Red Hook, NY, USA, December 2019.

[342] F. Paulovich and R. Minghim. Text map explorer: a tool to create and explore document maps. *Tenth International Conference on Information Visualisation (IV'06)*, pages 245–251, 2006. Referenced in Table C.1.

[343] F. Paulovich, Cláudio T. Silva, and L. G. Nonato. Two-phase mapping for projecting massive data sets. *IEEE Transactions on Visualization and Computer Graphics*, 16:1281–1290, 2010. Referenced in Table C.1.

[344] F. Paulovich, D. M. Eler, Jorge Poco, C. Botha, R. Minghim, and L. G. Nonato. Piece wise laplacian-based projection for interactive data exploration and organization. *Computer Graphics Forum*, 30, 2011. Referenced in Table C.1.

[345] Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901. Referenced in Table C.1.

[346] Fabian Pedregosa, Fabian Pedregosa, Gael Varoquaux, Gael Varoquaux, Normalesup Org, Alexandre Gramfort, Alexandre Gramfort, Vincent Michel, Vincent Michel, Logilab Fr, Bertrand Thirion, Bertrand Thirion, Olivier Grisel, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, Alexandre Tp, and David Cournapeau. Scikit-learn: Machine Learning in Python. *MACHINE LEARNING IN PYTHON*.

[347] E. Pekalska and R. Duin. Spatial representation of dissimilarity data via lower-complexity linear and nonlinear mappings. pages 488–497, 2002. Referenced in Table C.1.

[348] Richard K. Perez, M. Grace Gordon, Meena Subramaniam, Min Cheol Kim, George C. Hartoularos, Sasha Targ, Yang Sun, Anton Ogorodnikov, Raymund Bueno, Andrew Lu, Mike Thompson, Nadav Rappoport, Andrew Dahl, Cristina M. Lanata, Mehrdad Matloubian, Lenka Maliskova, Serena S. Kwek, Tony Li, Michal Slyper, Julia Waldman, Danielle Dionne, Orit Rozenblatt-Rosen, Lawrence Fong, Maria Dall'Era, Brunilda Balliu, Aviv Regev, Jinoos Yazdany, Lindsey A. Criswell, Noah Zaitlen, and Chun Jimmie Ye. Single-cell RNA-seq reveals cell type-specific molecular and genetic associations to lupus. *Science (New York, N.Y.)*, 376(6589):eabf1970, April 2022. ISSN 1095-9203. doi: 10.1126/science.abf1970.

[349] Viktor Petukhov, Rosalind J. Xu, Ruslan A. Soldatov, Paolo Cadinu, Konstantin Khodosevich, Jeffrey R. Moffitt, and Peter V. Kharchenko. Cell segmentation in imaging-based spatial transcriptomics. *Nature Biotechnology*, pages 1–10, October 2021. ISSN 1546-1696. doi: 10.1038/s41587-021-01044-w.

[350] Viktor Petukhov, Anna Igolkina, Rasmus Rydbirk, Shenglin Mei, Lars Christoffersen, Konstantin Khodosevich, and Peter V. Kharchenko. Case-control analysis of single-cell RNA-seq studies, March 2022.

[351] Nicolas Pielawski, Axel Andersson, Christophe Avenel, Andrea Behanova, Eduard Chelebian, Anna Klemm, Fredrik Nysjö, Leslie Solorzano, and Carolina Wählby. TissUUmaps 3: Improvements in interactive visualization, exploration, and quality assessment of large-scale spatial omics data. *Heliyon*, 9(5):e15306, May 2023.

[352] Mikael J. Pittet and Ralph Weissleder. Intravital Imaging. *Cell*, 147(5):983–991, November 2011. ISSN 0092-8674. doi: 10.1016/j.cell.2011.11.004.

[353] Krzysztof Polański, Raquel Bartolomé-Casado, Ioannis Sarropoulos, Chuan Xu, Nick England, Frode L Jahnsen, Sarah A Teichmann, and Nadav Yayon.

Bin2cell reconstructs cells from high resolution Visium HD data. *Bioinformatics*, 40(9):btae546, September 2024. ISSN 1367-4811. doi: 10.1093/bioinformatics/btae546.

[354] A. J. Pretorius, I. A. Khan, and R. J. Errington. A Survey of Visualization for Live Cell Imaging. *Computer Graphics Forum*, 36(1):46–63. ISSN 1467-8659. doi: 10.1111/cgf.12784.

[355] Yimiao Qu and Kyle M. Loh. Reshaping Waddington's developmental landscape. *Nature Reviews Genetics*, pages 1–1, September 2024. ISSN 1471-0064. doi: 10.1038/s41576-024-00777-z.

[356] Ellen M. Quardokus, Diane C. Saunders, Elizabeth McDonough, John W. Hickey, Christopher Werlein, Christine Surrette, Presha Rajbhandari, Anna Martinez Casals, Hua Tian, Lisa Lowery, Elizabeth K. Neumann, Frida Björklund, Taruna V. Neelakantan, Josh Croteau, Anne E. Wiblin, Jeremy Fisher, April J. Livengood, Karen G. Dowell, Jonathan C. Silverstein, Jeffrey M. Spraggins, Gloria S. Pryhuber, Gail Deutsch, Fiona Ginty, Garry P. Nolan, Simon Melov, Danny Jonigk, Michael A. Caldwell, Ioannis S. Vlachos, Werner Muller, Nils Gehlenborg, Brent R. Stockwell, Emma Lundberg, Michael P. Snyder, Ronald N. Germain, Jeannie M. Camarillo, Neil L. Kelleher, Katy Börner, and Andrea J. Radtke. Organ Mapping Antibody Panels: A community resource for standardized multiplexed tissue imaging. *Nature Methods*, 20(8):1174–1178, August 2023. ISSN 1548-7105. doi: 10.1038/s41592-023-01846-7.

[357] Rumana Rashid, Yu-An Chen, John Hoffer, Jeremy L. Muhlich, Jia-Ren Lin, Robert Krueger, Hanspeter Pfister, Richard Mitchell, Sandro Santagata, and Peter K. Sorger. Narrative online guides for the interpretation of digital-pathology images and tissue-atlas data. *Nature Biomedical Engineering*, 6(5): 515–526, May 2022. ISSN 2157-846X. doi: 10.1038/s41551-021-00789-8.

[358] Austin D. Reed, Sara Pensa, Adi Steif, Jack Stenning, Daniel J. Kunz, Linsey J. Porter, Kui Hua, Peng He, Alecia-Jane Twigger, Abigail J. Q. Siu, Katarzyna Kania, Rachel Barrow-McGee, Iain Goulding, Jennifer J. Gomm, Valerie Speirs, J. Louise Jones, John C. Marioni, and Walid T. Khaled. A single-cell atlas enables mapping of homeostatic cellular shifts in the adult human breast. *Nature Genetics*, 56(4), April 2024. ISSN 1061-4036. doi: 10.1038/s41588-024-01688-9.

[359] 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, Nir Yosef, and Human Cell Atlas Meeting Participants. The Human Cell Atlas. *eLife*, 6: e27041, December 2017. ISSN 2050-084X. doi: 10.7554/eLife.27041.

[360] Helge Ritter and T. Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61:241–254, 1989. Referenced in Table C.1.

[361] Jonathan C Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proc. CMV*, pages 61–71, Zurich, Switzerland, 2007. IEEE. doi: 10.1109/CMV.2007.20.

[362] James T. Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S. Lander, Gad Getz, and Jill P. Mesirov. Integrative genomics viewer. *Nature Biotechnology*, 29(1):24–26, January 2011. ISSN 1546-1696. doi: 10.1038/nbt.1754.

[363] Mark D. Robinson, Davis J. McCarthy, and Gordon K. Smyth. edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, January 2010. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp616.

[364] 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, March 2019.

[365] Jennifer E. Rood, Aidan Maartens, Anna Hupalowska, Sarah A. Teichmann, and Aviv Regev. Impact of the Human Cell Atlas on medicine. *Nature Medicine*, 28(12):2486–2496. ISSN 1546-170X. doi: 10.1038/s41591-022-02104-7.

[366] Jennifer E. Rood, Tim Stuart, Shila Ghazanfar, Tommaso Biancalani, Eyal Fisher, Andrew Butler, Anna Hupalowska, Leslie Gaffney, William Mauck, Gökçen Eraslan, John C. Marioni, Aviv Regev, and Rahul Satija. Toward a Common Coordinate Framework for the Human Body. *Cell*, 179(7):1455–1467, December 2019. ISSN 0092-8674, 1097-4172. doi: 10.1016/j.cell.2019.11.019.

[367] Jennifer E. Rood, Samantha Wynne, Lucia Robson, Anna Hupalowska, John Randell, Sarah A. Teichmann, and Aviv Regev. The Human Cell Atlas from a cell census to a unified foundation model. *Nature*, pages 1–2, November 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-08338-4.

[368] Florian A. Rosenberger, Marvin Thielert, Maximilian T. Strauss, Lisa Schweizer, Constantin Ammar, Sophia C. Mädler, Andreas Metousis, Patricia Skowronek, Maria Wahle, Katherine Madden, Janine Gote-Schniering, Anna Semenova, Herbert B. Schiller, Edwin Rodriguez, Thierry M. Nordmann, Andreas Mund, and Matthias Mann. Spatial single-cell mass spectrometry defines zonation of the hepatocyte proteome. *Nature Methods*, 20(10):1530–1536, October 2023. ISSN 1548-7105. doi: 10.1038/s41592-023-02007-6.

[369] David A. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77:125–141, 2008. Referenced in Table C.1.

[370] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290 5500:2323–6, 2000. Referenced in Table C.1.

[371] Orit Rozenblatt-Rosen, Michael J. T. Stubbington, Aviv Regev, and Sarah A. Teichmann. The Human Cell Atlas: From vision to reality. *Nature*, 550(7677): 451–453, October 2017. ISSN 1476-4687. doi: 10.1038/550451a.

[372] Orit Rozenblatt-Rosen, Aviv Regev, Philipp Oberdoerffer, Tal Nawy, Anna Hupalowska, Jennifer E. Rood, Orr Ashenberg, Ethan Cerami, Robert J. Coffey, Emek Demir, Li Ding, Edward D. Esplin, James M. Ford, Jeremy Goecks, Sharmistha Ghosh, Joe W. Gray, Justin Guinney, Sean E. Hanlon, Shannon K. Hughes, E. Shelley Hwang, Christine A. Iacobuzio-Donahue, Judit Jané-Valbuena, Bruce E. Johnson, Ken S. Lau, Tracy Lively, Sarah A. Mazzilli, Dana Pe'er, Sandro Santagata, Alex K. Shalek, Denis Schapiro, Michael P. Snyder, Peter K. Sorger, Avrum E. Spira, Sudhir Srivastava, Kai Tan, Robert B. West, Elizabeth H. Williams, and Human Tumor Atlas Network. The Human Tumor Atlas Network: Charting Tumor Transitions across Space and Time at Single-Cell Resolution. *Cell*, 181(2):236–249, April 2020. ISSN 0092-8674. doi: 10.1016/j.cell.2020.03.053.

[373] Laurie Rumker, Saori Sakaue, Yakir Reshef, Joyce B. Kang, Seyhan Yazar, Jose Alquicira-Hernandez, Cristian Valencia, Kaitlyn A. Lagattuta, Annelise Mah-Som, Aparna Nathan, Joseph E. Powell, Po-Ru Loh, and Soumya Raychaudhuri. Identifying genetic variants that influence the abundance of cell states in single-cell data. *Nature Genetics*, 56(10):2068–2077, October 2024. ISSN 1546-1718. doi: 10.1038/s41588-024-01909-1.

[374] Nagiza F. Samatova, Paul Breimyer, William Hendrix, Matthew C. Schmidt, and Theresa-Marie Rhyne. An outlook into ultra-scale visualization of large-scale biological data. In *2008 Workshop on Ultrascale Visualization*, pages 29–39. doi: 10.1109/ULTRAVIS.2008.5154061.

[375] J. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18:401–409, 1969. Referenced in Table C.1.

[376] Irzam Sarfraz, Yichen Wang, Amulya Shastry, Wei Kheng Teh, Artem Sokolov, Brian R. Herb, Heather H. Creasy, Isaac Virshup, Ruben Dries, Kylee Degatano, Anup Mahurkar, Daniel J. Schnell, Pedro Madrigal, Jason Hilton, Nils Gehlenborg, Timothy Tickle, and Joshua D. Campbell. MAMS: Matrix and analysis metadata standards to facilitate harmonization and reproducibility of single-cell data. *Genome Biology*, 25(1):205. ISSN 1474-760X. doi: 10.1186/s13059-024-03349-w.

[377] Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33(5):495–502, May 2015.

[378] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. Vega-Lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, August 2017. doi: 10.1109/TVCG.2016.2599030.

[379] Moritz Schaefer, Peter Peneder, Daniel Malzl, Mihaela Peycheva, Jake Burton, Anna Hakobyan, Varun Sharma, Thomas Krausgruber, Jörg Menche, Eleni M. Tomazou, and Christoph Bock. Multimodal learning of transcriptomes and text enables interactive single-cell RNA-seq data exploration with natural-language chats, October 2024.

[380] Denis Schapiro, Artem Sokolov, Clarence Yapp, Yu-An Chen, Jeremy L. Muhlich, Joshua Hess, Allison L. Creason, Ajit J. Nirmal, Gregory J. Baker, Maulik K. Nariya, Jia-Ren Lin, Zoltan Maliga, Connor A. Jacobson, Matthew W. Hodgman, Juha Ruokonen, Samouil L. Farhi, Domenic Abbondanza, Eliot T. McKinley, Daniel Persson, Courtney Betts, Shamilene Sivagnanam, Aviv Regev, Jeremy Goecks, Robert J. Coffey, Lisa M. Coussens, Sandro Santagata, and Peter K. Sorger. MCMICRO: A scalable, modular image-processing pipeline for multiplexed tissue imaging. *Nature Methods*, 19 (3):311–315, March 2022. ISSN 1548-7105. doi: 10.1038/s41592-021-01308-y.

[381] Denis Schapiro, Clarence Yapp, Artem Sokolov, Sheila M Reynolds, Yu-An Chen, Damir Sudar, Yubin Xie, Jeremy Muhlich, Raquel Arias-Camison,

Sarah Arena, Adam J Taylor, Milen Nikolov, Madison Tyler, Jia-Ren Lin, Erik A Burlingame, Human Tumor Atlas Network, Young H Chang, Samouil L Farhi, Vésteinn Thorsson, Nithya Venkatamohan, Julia L Drewes, Dana Pe'er, David A Gutman, Markus D Herrmann, Nils Gehlenborg, Peter Bankhead, Joseph T Roland, John M Herndon, Michael P Snyder, Michael Angelo, Garry Nolan, Jason R Swedlow, Nikolaus Schultz, Daniel T Merrick, Sarah A Mazzili, Ethan Cerami, Scott J Rodig, Sandro Santagata, and Peter K Sorger. MITI minimum information guidelines for highly multiplexed tissue images. *Nature Methods*, 19(3):262–267, March 2022.

[382] Maximilian Scherr. Multiple and coordinated views in information visualization. Technical report, Trends in information visualization, 2008.

[383] Caroline A Schneider, Wayne S Rasband, and Kevin W Eliceiri. NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, 9(7):671–675, July 2012. ISSN 1548-7091, 1548-7105. doi: 10.1038/nmeth.2089.

[384] B. Scholkopf, Alex Smola, and K. Müller. Kernel principal component analysis. pages 583–588, 1997. Referenced in Table C.1.

[385] Kinnary H Shah, Boyi Guo, and Stephanie C Hicks. Addressing the mean-variance relationship in spatially resolved transcriptomics data with spoon, November 2024.

[386] B Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, September 1996.

[387] V. Silva and J. Tenenbaum. Sparse multidimensional scaling using land-mark points. 2004. Referenced in Table C.1.

[388] Mitchell E. Skinner, Andrew V. Uzilov, Lincoln D. Stein, Christopher J. Mungall, and Ian H. Holmes. JBrowse: A next-generation genome browser. *Genome Research*, 19(9):1630–1638, September 2009. ISSN 1549-5469. doi: 10.1101/gr.094607.109.

[389] Alison Smith-Renner, Ron Fan, Melissa Birchfield, Tongshuang Wu, Jordan Boyd-Graber, Daniel S. Weld, and Leah Findlater. No Explainability without Accountability: An Empirical Study of Explanations and Feedback in Interactive ML. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–13, New York, NY, USA, April 2020. Association for Computing Machinery. ISBN 978-1-4503-6708-0. doi: 10.1145/3313831.3376624.

[390] Luke S. Snyder and Jeffrey Heer. DIVI: Dynamically Interactive Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):403–413, January 2024. ISSN 1941-0506. doi: 10.1109/TVCG.2023.3327172.

[391] Michael P Snyder, Shin Lin, Amanda Posgai, Mark Atkinson, Aviv Regev, Jennifer Rood, Orit Rozenblatt-Rosen, Leslie Gaffney, Anna Hupalowska, Rahul Satija, Nils Gehlenborg, Jay Shendure, Julia Laskin, Pehr Harbury, Nicholas A Nystrom, Jonathan C Silverstein, Ziv Bar-Joseph, Kun Zhang, Katy Börner, Yiing Lin, Richard Conroy, Dena Procaccini, Ananda L Roy, Ajay Pillai, Marishka Brown, Zorina S Galis, Long Cai, Jay Shendure, Cole Trapnell, Shin Lin, Dana Jackson, Michael P Snyder, Garry Nolan, William James Greenleaf, Yiing Lin, Sylvia Plevritis, Sara Ahadi, Stephanie A Nevins, Hayan Lee, Christian Martijn Schuerch, Sarah Black, Vishal Gautham Venkataraaman, Ed Esplin, Aaron Horning, Amir Bahmani, Kun Zhang, Xin Sun, Sanjay Jain, James Hagood, Gloria Pryhuber, Peter Kharchenko, Mark Atkinson, Bernd Bodenmiller, Todd Brusko, Michael Clare-Salzler, Harry Nick, Kevin Otto, Amanda Posgai, Clive Wasserfall, Marda Jorgensen, Maigan Brusko, Sergio Maffioletti, Richard M Caprioli, Jeffrey M Spraggins, Danielle Gutierrez, Nathan Heath Patterson, Elizabeth K Neumann, Raymond Harris, Mark deCaestecker, Agnes B Fogo, Raf van de Plas, Ken Lau, Long Cai, Guo-Cheng Yuan, Qian Zhu, Ruben Dries, Peng Yin, Sinem K Saka, Jocelyn Y Kishi, Yu Wang, Isabel Goldaracena, Julia Laskin, Donghye Ye, Kristin E Burnum-Johnson, Paul D Piehowski, Charles Ansong, Ying Zhu, Pehr Harbury, Tushar Desai, Jay Mulye, Peter Chou, Monica Nagendran, Ziv Bar-Joseph, Sarah A Teichmann, Benedict Paten, Robert F Murphy, Jian Ma, Vladimir Yu Kiselev, Carl Kingsford, Allyson Ricarte, Maria Keays, Sushma A Akoju, Matthew Ruffalo, Nils Gehlenborg, Peter Kharchenko, Margaret Vella, Chuck McCallum, Katy Börner, Leonard E Cross, Samuel H Friedman, Randy Heiland, Bruce Herr, Paul Macklin, Ellen M Quardokus, Lisel Record, James P Sluka, Griffin M Weber, Nicholas A Nystrom, Jonathan C Silverstein, Philip D Blood, Alexander J Ropelewski, William E Shirey, Robin M Scibek, Paula Mabee, W Christopher Lenhardt, Kimberly Robasky, Stavros Michailidis, Rahul Satija, John Marioni, Aviv Regev, Andrew Butler, Tim Stuart, Eyal Fisher, Shila Ghazanfar, Jennifer Rood, Leslie Gaffney, Gokcen Eraslan, Tommaso Biancalani, Eeshit D Vaishnav, Richard Conroy, Dena Procaccini, Ananda Roy, Ajay Pillai, Marishka Brown, Zorina Galis, Pothur Srinivas, Aaron Pawlyk, Salvatore Sechi, Elizabeth Wilder, James Anderson, HuBMAP Consortium, Writing Group, T M C Caltech-UW, T M C Stanford-WashU, Ucsd Tmc, University of Florida TMC, Vanderbilt University TMC, California Institute of Technology TTD, T T D Harvard, T T D Purdue, T T D Stanford, HuBMAP Integration, Visualization, and Engagement (HIVE) Collaboratory: Carnegie Mellon, Tools Component,

Harvard Medical School, Tools Component, Indiana University Bloomington, Mapping Component, Pittsburgh Supercomputing Center and University of Pittsburgh, Infrastructure and Engagement Component, University of South Dakota, Collaboration Core, New York Genome Center, Mapping Component, and NIH HuBMAP Working Group. The human body at cellular resolution: The NIH Human Biomolecular Atlas Program. *Nature*, 574(7777):187–192, October 2019.

[392] Nicholas Sofroniew, Talley Lambert, Grzegorz Bokota, Juan Nunez-Iglesias, Peter Sobolewski, Andrew Sweet, Lorenzo Gaifas, Kira Evans, Alister Burt, Draga Doncila Pop, Kevin Yamauchi, Melissa Weber Mendonça, Genevieve Buckley, Wouter-Michiel Vierdag, Loic Royer, Ahmet Can Solak, Kyle I. S. Harrington, Jannis Ahlers, Daniel Althviz Moré, Oren Amsalem, Ashley Anderson, Andrew Annex, Peter Boone, Jordão Bragantini, Matthias Bussonnier, Clément Caporal, Jan Eglinger, Andreas Eisenbarth, Jeremy Freeman, Christoph Gohlke, Kabilar Gunalan, Hagai Har-Gil, Mark Harfouche, Volker Hilsenstein, Katherine Hutchings, Jessy Lauer, Gregor Lichtner, Ziyang Liu, Lucy Liu, Alan Lowe, Luca Marconato, Sean Martin, Abigail McGovern, Lukasz Migas, Nadalyn Miller, Hector Muñoz, Jan-Hendrik Müller, Christopher Nauroth-Kreß, David Palecek, Constantin Pape, Eric Perlman, Kim Pevey, Gonzalo Peña-Castellanos, Andrea Pierré, David Pinto, Jaime Rodríguez-Guerra, David Ross, Craig T. Russell, James Ryan, Gabriel Selzer, MB Smith, Paul Smith, Konstantin Sofiiuk, Johannes Soltwedel, David Stansby, Jules Vanaret, Pam Wadhwa, Martin Weigert, Jonas Windhager, Philip Winston, and Rubin Zhao. Napari: A multi-dimensional image viewer for Python. Zenodo, March 2025.

[393] Antonios Somarakis, Marieke E. Ijsselsteijn, Sietse J. Luk, Boyd Kenkhuis, Noel F.C.C. de Miranda, Boudewijn P.F. Lelieveldt, and Thomas Höllt. Visual cohort comparison for spatial single-cell omics-data. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):733–743, 2021. doi: 10.1109/TVCG.2020.3030336.

[394] Andrew H. Song, Guillaume Jaume, Drew F. K. Williamson, Ming Y. Lu, Anurag Vaidya, Tiffany R. Miller, and Faisal Mahmood. Artificial intelligence for digital and computational pathology. *Nature Reviews Bioengineering*, 1(12): 930–949, December 2023. ISSN 2731-6092. doi: 10.1038/s44222-023-00096-8.

[395] C. Spearman. General intelligence objectively determined and measured. *American Journal of Psychology*, 15:201–293, 1904. Referenced in Table C.1.

[396] Matthew L Speir, Aparna Bhaduri, Nikolay S Markov, Pablo Moreno, Tomasz J Nowakowski, Irene Papatheodorou, Alex A Pollen, Brian J Raney, Lucas

Seninge, W James Kent, and Maximilian Haeussler. UCSC Cell Browser: Visualize your single-cell data. *Bioinformatics*, 37(23):4578–4580. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab503.

[397] Benjamin J. Stear, Taha Mohseni Ahooyi, J. Alan Simmons, Charles Kollar, Lance Hartman, Katherine Beigel, Aditya Lahiri, Shubha Vasisht, Tiffany J. Callahan, Christopher M. Nemarich, Jonathan C. Silverstein, and Deanne M. Taylor. Petagraph: A large-scale unifying knowledge graph framework for integrating biomolecular and biomedical data. *Scientific Data*, 11(1):1338, December 2024. ISSN 2052-4463. doi: 10.1038/s41597-024-04070-w.

[398] 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, September 2017.

[399] Chase Stokes, Vidya Setlur, Bridget Cogley, Arvind Satyanarayan, and Marti A. Hearst. Striking a Balance: Reader Takeaways and Preferences when Integrating Text and Charts. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2022. ISSN 1077-2626, 1941-0506, 2160-9306. doi: 10.1109/TVCG.2022.3209383.

[400] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: A generalist algorithm for cellular segmentation. *Nature Methods*, 18(1): 100–106, January 2021. ISSN 1548-7105. doi: 10.1038/s41592-020-01018-x.

[401] Yijing Su, Yi Zhou, Mariko L. Bennett, Shiying Li, Marc Carceles-Cordon, Lu Lu, Sooyoung Huh, Dennisse Jimenez-Cyrus, Benjamin C. Kennedy, Sudha K. Kessler, Angela N. Viaene, Ingo Helbig, Xiaosong Gu, Joel E. Kleinman, Thomas M. Hyde, Daniel R. Weinberger, David W. Nauen, Hongjun Song, and Guo-li Ming. A single-cell transcriptome atlas of glial diversity in the human hippocampus across the postnatal lifespan. *Cell Stem Cell*, 29(11):1594+, November 2022. ISSN 1934-5909. doi: 10.1016/j.stem.2022.09.010.

[402] Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550, October 2005. doi: 10.1073/pnas.0506580102.

[403] Valentine Svensson, Eduardo da Veiga Beltrame, and Lior Pachter. A curated database reveals trends in single-cell transcriptomics. *Database*, 2020. doi: 10.1093/database/baaa073.

[404] Valentine Svensson, Roser Vento-Tormo, and Sarah A. Teichmann. Exponential scaling of single-cell RNA-seq in the past decade. *Nature Protocols*, 13(4):599–604, April 2018. ISSN 1750-2799. doi: 10.1038/nprot.2017.149.

[405] Shaocong Tan, Chufan Lai, Xiaolong Luke Zhang, and Xiaoru Yuan. An Empirical Guide for Visualization Consistency in Multiple Coordinated Views. In *2023 IEEE 16th Pacific Visualization Symposium (PacificVis)*, pages 31–40, April 2023. doi: 10.1109/PacificVis56936.2023.00011.

[406] Leyla Tarhan, Jon Bistline, Jean Chang, Bryan Galloway, Emily Hanna, and Eric Weitz. Single Cell Portal: An interactive home for single-cell genomics data. *bioRxiv*, page 2023.07.13.548886, July 2023. doi: 10.1101/2023.07.13.548886.

[407] Y. Teh and S. Roweis. Automatic alignment of local representations. pages 841–848, 2002. Referenced in Table C.1.

[408] E. Tejada, R. Minghim, and L. G. Nonato. On improved projection techniques to support visual exploration of multi-dimensional data sets. *Information Visualization*, 2:218 – 231, 2003. Referenced in Table C.1.

[409] J. Tenenbaum, V. Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290 5500:2319–23, 2000. Referenced in Table C.1.

[410] Marina Terekhova, Amanda Swain, Pavla Bohacova, Ekaterina Aladyeva, Laura Arthur, Anwesha Laha, Denis A. Mogilenko, Samantha Burdess, Vladimir Sukhov, Denis Kleverov, Barbora Echalar, Petr Tsurinov, Roman Chernyatchik, Kamila Husarcikova, and Maxim N. Artyomov. Single-cell atlas of healthy human blood unveils age-related loss of NKG2C+GZMB-CD8+memory T cells and accumulation of type 2 memory T cells. *Immunity*, 56(12), December 2023. ISSN 1074-7613. doi: 10.1016/j.immuni.2023.10.013.

[411] Hua Tian, Presha Rajbhandari, Jay Tarolli, Aubrianna M Decker, Taruna V Neelakantan, Tina Angerer, Fereshteh Zandkarimi, Jacob Daniels, Helen Remotti, Gilles Frache, Nicholas Winograd, and Brent R Stockwell. Multi-modal mass spectrometry imaging reveals single-cell metabolic states in mammalian liver. September 2022.

[412] Michael E. Tipping and Charles M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61, 1999. Referenced in Table C.1.

[413] Michael Totty, Stephanie C. Hicks, and Boyi Guo. SpotSweeper: Spatially-aware quality control for spatial transcriptomics, June 2024.

[414] F. William Townes, Stephanie C. Hicks, Martin J. Aryee, and Rafael A. Irizarry. Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model. *Genome Biology*, 20(1):295, December 2019. ISSN 1474-760X. doi: 10.1186/s13059-019-1861-6.

[415] Kyle J. Travaglini, Ahmad N. Nabhan, Lolita Penland, Rahul Sinha, Astrid Gillich, Rene V. Sit, Stephen Chang, Stephanie D. Conley, Yasuo Mori, Jun Seita, Gerald J. Berry, Joseph B. Shrager, Ross J. Metzger, Christin S. Kuo, Norma Neff, Irving L. Weissman, Stephen R. Quake, and Mark A. Krasnow. A molecular cell atlas of the human lung from single-cell RNA sequencing. *Nature*, 587(7835), November 2020. ISSN 0028-0836. doi: 10.1038/s41586-020-2922-4.

[416] Pavol Ulbrich, Manuela Waldner, Katarina Furmanova, Sergio M Marques, David Bednar, Barbora Kozlikova, and Jan Byska. sMolBoxes: Dataflow model for molecular dynamics exploration. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):581–590, September 2023. doi: 10.1109/TVCG.2022.3209411.

[417] Astrid van den Brandt, Sehi L'Yi, Huyen N. Nguyen, Anna Vilanova, and Nils Gehlenborg. Understanding Visualization Authoring Techniques for Genomics Data in the Context of Personas and Tasks. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2024. ISSN 1941-0506. doi: 10.1109/TVCG.2024.3456298.

[418] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.

[419] Tave van Zyl, Wenjun Yan, Alexi M. McAdams, Aboozar Monavarfeshani, Gregory S. Hageman, and Joshua R. Sanes. Cell atlas of the human ocular anterior segment: Tissue-specific and shared cell types. *Proceedings of the National Academy of Sciences of the United States of America*, 119(e2200914119), July 2022. ISSN 0027-8424. doi: 10.1073/pnas.2200914119.

[420] Katy Vandereyken, Alejandro Sifrim, Bernard Thienpont, and Thierry Voet. Methods and applications for single-cell and spatial multi-omics. *Nature Reviews Genetics*, pages 1–22, March 2023. ISSN 1471-0064. doi: 10.1038/s41576-023-00580-2.

[421] Jacob VanderPlas, Brian Granger, Jeffrey Heer, Dominik Moritz, Kanit Wongsuphasawat, Arvind Satyanarayan, Eitan Lees, Ilia Timofeev, Ben Welsh, and Scott Sievert. Altair: Interactive Statistical Visualizations for Python. *Journal of Open Source Software*, 3(32):1057, December 2018. ISSN 2475-9066. doi: 10.21105/joss.01057.

[422] Michael VanInsberghe and Alexander van Oudenaarden. Sequencing technologies to measure translation in single cells. *Nature Reviews Molecular Cell Biology*, pages 1–10, January 2025. ISSN 1471-0080. doi: 10.1038/s41580-024-00822-z.

[423] Aarthi Venkat, Sam Leone, Scott E. Youlten, Eric Fagerberg, John Attanasio, Nikhil S. Joshi, Michael Perlmutter, and Smita Krishnaswamy. Mapping the gene space at single-cell resolution with gene signal pattern analysis. *Nature Computational Science*, 4(12):955–977, December 2024. ISSN 2662-8457. doi: 10.1038/s43588-024-00734-0.

[424] Jarkko Venna, J. Peltonen, K. Nybo, Helena Aidos, and Samuel Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *J. Mach. Learn. Res.*, 11:451–490, 2010. Referenced in Table C.1.

[425] A. Vieth, T. Kroes, J. Thijssen, B. van Lew, J. Eggermont, S. Basu, E. Eisemann, A. Vilanova, T. Hollt, and B. Lelieveldt. ManiVault: A flexible and extensible visual analytics framework for high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 30(01):175–185, January 2024. ISSN 1941-0506. doi: 10.1109/TVCG.2023.3326582.

[426] Isaac Virshup, Danila Bredikhin, Lukas Heumos, Giovanni Palla, Gregor Sturm, Adam Gayoso, Ilia Kats, Mikaela Koutrouli, Bonnie Berger, Dana Pe'er, Aviv Regev, Sarah A. Teichmann, Francesca Finotello, F. Alexander Wolf, Nir Yosef, Oliver Stegle, and Fabian J. Theis. The scverse project provides a computational ecosystem for single-cell omics data analysis. *Nature Biotechnology*, 41(5):604–606, May 2023. ISSN 1546-1696. doi: 10.1038/s41587-023-01733-8.

[427] Isaac Virshup, Sergei Rybakov, Fabian J. Theis, Philipp Angerer, and F. Alexander Wolf. Anndata: Access and store annotated data matrices. *Journal of Open Source Software*, 9(101):4371, September 2024. ISSN 2475-9066. doi: 10.21105/joss.04371.

[428] C. H. Waddington. *The Strategy of the Genes*. Routledge, London, April 1957. ISBN 978-1-315-76547-1. doi: 10.4324/9781315765471.

[429] Thomas Waelchli, Moheb Ghobrial, Marc Schwab, Shigeki Takada, Hang Zhong, Samuel Suntharalingham, Sandra Vetiska, Dayme Rodrigues Gonzalez, Ruilin Wu, Hubert Rehrauer, Anuroopa Dinesh, Kai Yu, Edward L. Y. Chen, Jeroen Bisschop, Fiona Farnhammer, Ann Mansur, Joanna Kalucka, Itay Tirosh, Luca Regli, Karl Schaller, Karl Frei, Troy Ketela, Mark Bernstein, Paul Kongkham, Peter Carmeliet, Taufik Valiante, Peter B. Dirks, Mario L. Suva, Gelareh Zadeh, Viviane Tabar, Ralph Schlapbach, Hartland W. Jackson, Katrien De Bock, Jason E. Fish, Philippe P. Monnier, Gary D. Bader, and Ivan

Radovanovic. Single-cell atlas of the human brain vasculature across development, adulthood and disease. *Nature*, 2024 JUL 10 2024. ISSN 0028-0836. doi: 10.1038/s41586-024-07493-y.

[430] Daniel E. Wagner and Allon M. Klein. Lineage tracing meets single-cell omics: Opportunities and challenges. *Nature Reviews Genetics*, 21(7):410–427, July 2020. ISSN 1471-0064. doi: 10.1038/s41576-020-0223-2.

[431] Shu Wang, Eduardo D. Sontag, and Douglas A. Lauffenburger. What cannot be seen correctly in 2D visualizations of single-cell 'omics data? *Cell Systems*, 14 (9):723–731, September 2023. ISSN 2405-4712, 2405-4720. doi: 10.1016/j.cels.2023.07.002.

[432] Tianyu Wang, Boyang Li, Craig E. Nelson, and Sheida Nabavi. Comparative analysis of differential gene expression analysis tools for single-cell RNA sequencing data. *BMC Bioinformatics*, 20(1):40, January 2019. ISSN 1471-2105. doi: 10.1186/s12859-019-2599-6.

[433] Yang Wang. Deck.gl: Large-scale web-based visual analytics made easy. October 2019.

[434] Yijie Wang, Jan Hoinka, and Teresa M. Przytycka. Subpopulation Detection and Their Comparative Analysis across Single-Cell Experiments with scPopCorn. *Cell Systems*, 8(6):506–513.e5, June 2019. ISSN 2405-4720. doi: 10.1016/j.cels.2019.05.007.

[435] Zeshuai Wang, Zhisheng Wu, Hao Wang, Ruoqing Feng, Guanlin Wang, Muxi Li, Shuang-Yin Wang, Xiaoyan Chen, Yiyi Su, Jun Wang, Weiwen Zhang, Yuzhou Bao, Zhenwei Lan, Zhuo Song, Yiheng Wang, Xianyang Luo, Lingyu Zhao, Anli Hou, Shuye Tian, Hongliang Gao, Wenbin Miao, Yingyu Liu, Huilin Wang, Cui Yin, Zhi-Liang Ji, Mingqian Feng, Hongkun Liu, Lianghui Diao, Ido Amit, Yun Chen, Yong Zeng, Florent Ginhoux, Xueqing Wu, Yuanfang Zhu, and Hanjie Li. An immune cell atlas reveals the dynamics of human macrophage specification during prenatal development. *Cell*, 186(20):4454+, September 2023. ISSN 0092-8674. doi: 10.1016/j.cell.2023.08.019.

[436] Zheng Wang, Chengyan Chai, Rui Wang, Yimei Feng, Lei Huang, Yiming Zhang, Xia Xiao, Shijie Yang, Yunfang Zhang, and Xi Zhang. Single-cell transcriptome atlas of human mesenchymal stem cells exploring cellular heterogeneity. *Clinical and Translational Medicine*, 11(e650), December 2021. ISSN 2001-1326. doi: 10.1002/ctm2.650.

[437] S. Warchol, J. Troidl, J. Muhlich, R. Krueger, J. Hoffer, T. Lin, J. Beyer, E. Glassman, P. Sorger, and H. Pfister. Psudo: Exploring Multi-Channel

Biomedical Image Data with Spatially and Perceptually Optimized Pseudocoloring. *Computer Graphics Forum*, 43(3):e15103, June 2024. ISSN 0167-7055, 1467-8659. doi: 10.1111/cgf.15103.

[438] Griffin M. Weber, Yingnan Ju, and Katy Börner. Considerations for Using the Vasculature as a Coordinate System to Map All the Cells in the Human Body. *Frontiers in Cardiovascular Medicine*, 7, 2020. ISSN 2297-055X. doi: 10.3389/fcvm.2020.00029.

[439] Kilian Q. Weinberger and L. Saul. Distance metric learning for large margin nearest neighbor classification. pages 1473–1480, 2005. Referenced in Table C.1.

[440] Kilian Q. Weinberger, Fei Sha, and L. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *International Conference on Machine Learning*, 2004. Referenced in Table C.1.

[441] Kilian Q. Weinberger, Ben Packer, and L. Saul. Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. 2005. Referenced in Table C.1.

[442] Daniel Wendisch, Oliver Dietrich, Tommaso Mari, Saskia von Stillfried, Ignacio L. Ibarra, Mirja Mittermaier, Christin Mache, Robert Lorenz Chua, Rainer Knoll, Sara Timm, Sophia Brumhard, Tobias Krammer, Henrik Zauber, Anna Luisa Hiller, Anna Pascual-Reguant, Ronja Mothes, Roman David Bülow, Jessica Schulze, Alexander M. Leipold, Sonja Djudjaj, Florian Erhard, Robert Geffers, Fabian Pott, Julia Kazmierski, Josefine Radke, Panagiotis Pergantis, Kevin Baßler, Claudia Conrad, Anna C. Aschenbrenner, Birgit Sawitzki, Markus Landthaler, Emanuel Wyler, David Horst, Deutsche COVID-19 OMICS Initiative (DeCOI), Stefan Hippenstiel, Andreas Hocke, Frank L. Heppner, Alexander Uhrig, Carmen Garcia, Felix Machleidt, Susanne Herold, Sefer Elezkurtaj, Charlotte Thibeault, Martin Witzenrath, Clément Cochain, Norbert Suttorp, Christian Drosten, Christine Goffinet, Florian Kurth, Joachim L. Schultze, Helena Radbruch, Matthias Ochs, Roland Eils, Holger Müller-Redetzky, Anja E. Hauser, Malte D. Luecken, Fabian J. Theis, Christian Conrad, Thorsten Wolff, Peter Boor, Matthias Selbach, Antoine-Emmanuel Saliba, and Leif Erik Sander. SARS-CoV-2 infection triggers profibrotic macrophage responses and lung fibrosis. *Cell*, 184(26):6243–6261.e27, December 2021. ISSN 1097-4172. doi: 10.1016/j.cell.2021.11.033.

[443] Brandon T. Wesley, Alexander D. B. Ross, Daniele Muraro, Zhichao Miao, Sarah Saxton, Rute A. Tomaz, Carola M. Morell, Katherine Ridley, Ekaterini D. Zacharis, Sandra Petrus-Reurer, Judith Kraiczy, Krishnaa T. Mahbubani, Stephanie Brown, Jose Garcia-Bernardo, Clara Alsinet, Daniel Gaffney,

Dave Horsfall, Olivia C. Tysoe, Rachel A. Botting, Emily Stephenson, Dorin-Mirel Popescu, Sonya MacParland, Gary Bader, Ian D. McGilvray, Daniel Ortmann, Fotios Sampaziotis, Kourosh Saeb-Parsy, Muzlifah Haniffa, Kelly R. Stevens, Matthias Zilbauer, Sarah A. Teichmann, and Ludovic Vallier. Single-cell atlas of human liver development reveals pathways directing hepatic cell fates. *Nature Cell Biology*, 24(10):1487–1498, October 2022. ISSN 1465-7392, 1476-4679. doi: 10.1038/s41556-022-00989-7.

[444] Michael N. Wicks, Michael Glinka, Bill Hill, Derek Houghton, Mehran Sharghi, Ingrid Ferreira, David Adams, Shahida Din, Irene Papatheodorou, Kathryn Kirkwood, Michael Cheeseman, Albert Burger, Richard A. Baldock, and Mark J. Arends. The Comparative Pathology Workbench: Interactive visual analytics for biomedical data. *Journal of Pathology Informatics*, 14:100328, January 2023. ISSN 2153-3539. doi: 10.1016/j.jpi.2023.100328.

[445] Drake Winslow Williams, Teresa Greenwell-Wild, Laurie Brenchley, Nicolas Dutzan, Andrew Overmiller, Andrew Phillip Sawaya, Simone Webb, Daniel Martin, George Hajishengallis, Kimon Divaris, Maria Morasso, Muzlifah Haniffa, Niki Maria Moutsopoulos, and NIDCD NIDCR Genomics Computational. Human oral mucosa cell atlas reveals a stromal-neutrophil axis regulating tissue immunity. *Cell*, 184(e15):4090+, July 2021. ISSN 0092-8674. doi: 10.1016/j.cell.2021.05.013.

[446] Ethan A. Winkler, Chang N. Kim, Jayden M. Ross, Joseph H. Garcia, Eugene Gil, Irene Oh, Lindsay Q. Chen, David Wu, Joshua S. Catapano, Kunal Raygor, Kazim Narsinh, Helen Kim, Shantel Weinsheimer, Daniel L. Cooke, Brian P. Walcott, Michael T. Lawton, Nalin Gupta, Berislav Zlokovic, V, Edward F. Chang, Adib A. Abla, Daniel A. Lim, and Tomasz J. Nowakowski. A single-cell atlas of the normal and malformed human brain vasculature. *Science*, 375 (eabi7377):992+, March 2022. ISSN 0036-8075. doi: 10.1126/science.abi7377.

[447] Laurenz Wiskott and T. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14:715–770, 2002. Referenced in Table C.1.

[448] F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. SCANPY: Large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15. ISSN 1474-760X. doi: 10.1186/s13059-017-1382-0.

[449] Lucy Xia, Christy Lee, and Jingyi Jessica Li. Statistical method scDEED for detecting dubious 2D single-cell embeddings and optimizing t-SNE and UMAP hyperparameters. *Nature Communications*, 15(1):1753, February 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-45891-y.

[450] Yichi Xu, Tengjiao Zhang, Qin Zhou, Mengzhu Hu, Yao Qi, Yifang Xue, Yuxiao Nie, Lihui Wang, Zhirong Bao, and Weiyang Shi. A single-cell transcriptome atlas profiles early organogenesis in human embryos. *Nature Cell Biology*, 25(4): 604–615, April 2023. ISSN 1476-4679. doi: 10.1038/s41556-023-01108-w.

[451] Feng Yan, David R. Powell, David J. Curtis, and Nicholas C. Wong. From reads to insight: A hitchhiker's guide to ATAC-seq data analysis. *Genome Biology*, 21 (1):22, February 2020. ISSN 1474-760X. doi: 10.1186/s13059-020-1929-3.

[452] Wenjun Yan, Yi-Rong Peng, Tave van Zyl, Aviv Regev, Karthik Shekhar, Dejan Juric, and Joshua R. Sanes. Cell atlas of the human fovea and peripheral retina. *Scientific Reports*, 10(9802), June 2020. ISSN 2045-2322. doi: 10.1038/s41598-020-66092-9.

[453] Karren Dai Yang, Anastasiya Belyaeva, Saradha Venkatachalapathy, Karthik Damodaran, Abigail Katcoff, Adityanarayanan Radhakrishnan, G. V. Shivashankar, and Caroline Uhler. Multi-domain translation between single-cell imaging and sequencing data using autoencoders. *Nature Communications*, 12 (1):31. ISSN 2041-1723. doi: 10.1038/s41467-020-20249-2.

[454] Jianing Yao, Jinglun Yu, Brian Caffo, Stephanie C. Page, Keri Martinowich, and Stephanie C. Hicks. Spatial domain detection using contrastive self-supervised learning for spatial multi-omics technologies, February 2024.

[455] Clarence Yapp, Ajit J. Nirmal, Felix Zhou, Zoltan Maliga, Juliann B. Tefft, Paula Montero Llopis, George F. Murphy, Christine G. Lian, Gaudenz Danuser, Sandro Santagata, and Peter K. Sorger. Multiplexed 3D Analysis of Immune States and Niches in Human Tissue, March 2024.

[456] Seyhan Yazar, Jose Alquicira-Hernandez, Kristof Wing, Anne Senabouth, M. Grace Gordon, Stacey Andersen, Qinyi Lu, Antonia Rowson, Thomas R. P. Taylor, Linda Clarke, Katia Maccora, Christine Chen, Anthony L. Cook, Chun Jimmie Ye, Kirsten A. Fairfax, Alex W. Hewitt, and Joseph E. Powell. Single-cell eQTL mapping identifies cell type–specific genetic control of autoimmune disease. *Science*, 376(6589):eabf3041, April 2022. doi: 10.1126/science.abf3041.

[457] Ji Soo Yi, Youn ah Kang, John Stasko, and J.A. Jacko. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, November 2007. ISSN 1941-0506. doi: 10.1109/TVCG.2007.70515.

[458] Hujun Yin. ViSOM - a novel method for multivariate data projection and structure visualization. *IEEE transactions on neural networks*, 13 1:237–43, 2002. Referenced in Table C.1.

[459] Satoshi Yoshiji, Tianyuan Lu, Guillaume Butler-Laporte, Julia Carrasco-Zanini-Sanchez, Chen-Yang Su, Yiheng Chen, Kevin Liang, Julian Daniel Sunday Willett, Shidong Wang, Darin Adra, Yann Ilboudo, Takayoshi Sasako, Satoshi Koyama, Tetsushi Nakao, Vincenzo Forgetta, Yossi Farjoun, Hugo Zeberg, Sirui Zhou, Michael Marks-Hultström, Mitchell J. Machiela, Rama Kaalia, Hesam Dashti, Melina Claussnitzer, Jason Flannick, Nicholas J. Wareham, Vincent Mooser, Nicholas J. Timpson, Claudia Langenberg, and J. Brent Richards. Integrative proteogenomic analysis identifies COL6A3-derived endotrophin as a mediator of the effect of obesity on coronary artery disease. *Nature Genetics*, 57 (2):345–357, February 2025. ISSN 1546-1718. doi: 10.1038/s41588-024-02052-7.

[460] Matthew D. Young, Thomas J. Mitchell, Felipe A. Vieira Braga, Maxine G. B. Tran, Benjamin J. Stewart, John R. Ferdinand, Grace Collord, Rachel A. Botting, Dorin-Mirel Popescu, Kevin W. Loudon, Roser Vento-Tormo, Emily Stephenson, Alex Cagan, Sarah J. Farndon, Martin Del Castillo Velasco-Herrera, Charlotte Guzzo, Nathan Richoz, Lira Mamanova, Tevita Aho, James N. Armitage, Antony C. P. Riddick, Imran Mushtaq, Stephen Farrell, Dyanne Rampling, James Nicholson, Andrew Filby, Johanna Burge, Steven Lisgo, Patrick H. Maxwell, Susan Lindsay, Anne Y. Warren, Grant D. Stewart, Neil Sebire, Nicholas Coleman, Muzlifah Haniffa, Sarah A. Teichmann, Menna Clatworthy, and Sam Behjati. Single-cell transcriptomes from human kidneys reveal the cellular identity of renal tumors. *Science*, 361(6402):594–599, August 2018. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aat1699.

[461] Bowen Yu and Claudio T Silva. VisFlow - web-based visualization framework for tabular data with a subset flow model. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):251–260, January 2017. doi: 10.1109/TVCG.2016.2598497.

[462] Guo-Cheng Yuan, Long Cai, Michael Elowitz, Tariq Enver, Guoping Fan, Guoji Guo, Rafael Irizarry, Peter Kharchenko, Junhyong Kim, Stuart Orkin, John Quackenbush, Assieh Saadatpour, Timm Schroeder, Ramesh Shivdasani, and Itay Tirosh. Challenges and emerging directions in single-cell analysis. *Genome Biology*, 18(1):84, May 2017.

[463] Luke Zappia and Fabian J. Theis. Over 1000 tools reveal trends in the single-cell RNA-seq analysis landscape. *Genome Biology*, 22(1):301. ISSN 1474-760X. doi: 10.1186/s13059-021-02519-4.

[464] Luke Zappia, Belinda Phipson, and Alicia Oshlack. Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database. *PLOS Computational Biology*, 14(6):e1006245. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1006245.

[465] Ron Zeira, Max Land, Alexander Strzalkowski, and Benjamin J. Raphael. Alignment and integration of spatial transcriptomics data. *Nature Methods*, 19(5):567–575, May 2022. ISSN 1548-7105. doi: 10.1038/s41592-022-01459-6.

[466] Bao Zhang, Peng He, John E. G. Lawrence, Shuaiyu Wang, Elizabeth Tuck, Brian A. Williams, Kenny Roberts, Vitalii Kleshchevnikov, Lira Mamanova, Liam Bolt, Krzysztof Polanski, Tong Li, Rasa Elmentaite, Eirini S. Fasouli, Martin Prete, Xiaoling He, Nadav Yayon, Yixi Fu, Hao Yang, Chen Liang, Hui Zhang, Raphael Blain, Alain Chedotal, David R. Fitzpatrick, Helen Firth, Andrew Dean, Omer Ali Bayraktar, John C. Marioni, Roger A. Barker, Mekayla A. Storer, Barbara J. Wold, Hongbo Zhang, and Sarah A. Teichmann. A human embryonic limb cell atlas resolved in space and time. *Nature*, 2023 DEC 6 2023. ISSN 0028-0836. doi: 10.1038/s41586-023-06806-x.

[467] Kai Zhang, James D. Hocker, Michael Miller, Xiaomeng Hou, Joshua Chiou, Olivier B. Poirion, Yunjiang Qiu, Yang E. Li, Kyle J. Gaulton, Allen Wang, Sebastian Preissl, and Bing Ren. A single-cell atlas of chromatin accessibility in the human genome. *Cell*, 184(24):5985+, November 2021. ISSN 0092-8674. doi: 10.1016/j.cell.2021.10.024.

[468] Tianhao Zhang, Jie Yang, Deli Zhao, and Xinliang Ge. Linear local tangent space alignment and application to face recognition. *Neurocomputing*, 70:1547–1553, 2007. Referenced in Table C.1.

[469] Zhenyue Zhang and J. Wang. MLLE: Modified locally linear embedding using multiple weights. pages 1593–1600, 2006. Referenced in Table C.1.

[470] Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *Journal of Shanghai University (English Edition)*, 8:406–424, 2004. Referenced in Table C.1.

[471] Ziqi Zhang, Haoran Sun, Ragunathan Mariappan, Xi Chen, Xinyu Chen, Mika S. Jain, Mirjana Efremova, Sarah A. Teichmann, Vaibhav Rajan, and Xiuwei Zhang. scMoMaT jointly performs single cell mosaic integration and multi-modal bio-marker detection. *Nature Communications*, 14(1):384, January 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-36066-2.

[472] LiangYu Zhao, Sha Han, HengChuan Su, JianYing Li, ErLei Zhi, Peng Li, ChenCheng Yao, RuHui Tian, HuiXing Chen, HuiRong Chen, JiaQiang Luo,

ChenKun Shi, ZhiYong Ji, JianLin Hu, Gang Wu, WeiDong Zhou, YuXin Tang, YuZhuo Chen, GuiTing Lin, Tom F. Lue, DengLong Wu, and Zheng Li. Single-cell transcriptome atlas of the human corpus cavernosum. *Nature Communications*, 13(4302), July 2022. doi: 10.1038/s41467-022-31950-9.

[473] Rongbin Zheng, Changxin Wan, Shenglin Mei, Qian Qin, Qiu Wu, Hanfei Sun, Chen-Hao Chen, Myles Brown, Xiaoyan Zhang, Clifford A. Meyer, and X. Shirley Liu. Cistrome Data Browser: Expanded datasets and new tools for gene regulatory analysis. *Nucleic Acids Research*, 47(D1):D729–D735, January 2019. ISSN 0305-1048. doi: 10.1093/nar/gky1094.

[474] Xin Zhou, Rebecca F. Lowdon, Daofeng Li, Heather A. Lawson, Pamela A. F. Madden, Joseph F. Costello, and Ting Wang. Exploring long-range genome interactions using the WashU Epigenome Browser. *Nature Methods*, 10(5):375–376, May 2013. ISSN 1548-7105. doi: 10.1038/nmeth.2440.

[475] Xiaowei Zhuang. Spatially resolved single-cell genomics and transcriptomics by imaging. *Nature Methods*, 18(1):18–22, January 2021.

[476] Carly G. K. Ziegler, Vincent N. Miao, Anna H. Owings, Andrew W. Navia, Ying Tang, Joshua D. Bromley, Peter Lotfy, Meredith Sloan, Hannah Laird, Haley B. Williams, Micayla George, Riley S. Drake, Taylor Christian, Adam Parker, Campbell B. Sindel, Molly W. Burger, Yilianys Pride, Mohammad Hasan, George E. Abraham, Michal Senitko, Tanya O. Robinson, Alex K. Shalek, Sarah C. Glover, Bruce H. Horwitz, and Jose Ordovas-Montanes. Impaired local intrinsic immunity to SARS-CoV-2 infection in severe COVID-19. *Cell*, 184(18):4713–4733.e22, September 2021. ISSN 1097-4172. doi: 10.1016/j.cell.2021.07.023.

[477] Marinka Zitnik, Michelle M Li, Aydin Wells, Kimberly Glass, Deisy Morselli Gysi, Arjun Krishnan, T M Murali, Predrag Radivojac, Sushmita Roy, Anaïs Baudot, Serdar Bozdag, Danny Z Chen, Lenore Cowen, Kapil Devkota, Anthony Gitter, Sara J C Gosline, Pengfei Gu, Pietro H Guzzi, Heng Huang, Meng Jiang, Ziynet Nesibe Kesimoglu, Mehmet Koyuturk, Jian Ma, Alexander R Pico, Nataša Pržulj, Teresa M Przytycka, Benjamin J Raphael, Anna Ritz, Roded Sharan, Yang Shen, Mona Singh, Donna K Slonim, Hanghang Tong, Xinan Holly Yang, Byung-Jun Yoon, Haiyuan Yu, and Tijana Milenković. Current and future directions in network biology. *Bioinformatics Advances*, 4 (1):vbae099, January 2024. ISSN 2635-0041. doi: 10.1093/bioadv/vbae099.

[478] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15:265 – 286, 2006. Referenced in Table C.1.