

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

Interpretable Quantification of Molecular Heterogeneity and Sample Effects in Single-cell Omics

Permalink

<https://escholarship.org/uc/item/68q7z0q7>

ISBN

9798288862625

Author

Boyeau, Pierre

Publication Date

2025-05-15

Peer reviewed|Thesis/dissertation

Interpretable Quantification of Molecular Heterogeneity and Sample Effects in
Single-cell Omics

By

Pierre Boyeau

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering — Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Michael I. Jordan, Co-Chair
Associate Professor Nir Yosef, Co-Chair
Professor Sandrine Dudoit
Professor Jennifer Listgarten

Spring 2025

Interpretable Quantification of Molecular Heterogeneity and Sample Effects in
Single-cell Omics

Copyright 2025
by
Pierre Boyeau

Abstract

Interpretable Quantification of Molecular Heterogeneity and Sample Effects in
Single-cell Omics

by

Pierre Boyeau

Doctor of Philosophy in Engineering — Electrical Engineering and Computer
Sciences

University of California, Berkeley

Professor Michael I. Jordan, Co-Chair

Associate Professor Nir Yosef, Co-Chair

Rapid advances in the scale and scope of single-cell omics technologies now enable detailed characterization of cellular heterogeneity. Moving beyond the analysis of a few samples, current approaches leverage large cohorts and integrate multi-modal data to investigate cell state variation more comprehensively. Computational methods supporting analysis of single-cell omics have coevolved, with integrated modeling frameworks, notably latent variable models, emerging as powerful tools for learning representations disentangled from technical confounders. While these advanced models have gained traction for core tasks including data integration and annotation, their application to core statistical inference required for rigorous biological interpretation—such as differential expression (DE) analysis or association testing—remained comparatively underdeveloped.

This dissertation addresses this gap by developing interpretable probabilistic frameworks designed for reliable statistical inference with minimal assumptions across diverse single-cell and spatial omics settings. These frameworks cover differential expression analysis, conditional association testing, and annotation-free analysis of large-scale, multi-sample datasets. Complementing these approaches, this work also introduces a framework for model evaluation and selection that addresses the lack of ground truth in omics through prediction-powered inference. Furthermore, this research addresses the challenge of preprocessing errors in spatial transcriptomics

by quantifying their impact, particularly from cell segmentation, to improve the reliability of downstream differential expression analysis.

To Mathilde, Zoe, Janine, and Catherine

Acknowledgments

My time at Berkeley has been a powerful and passionate experience, and I would like to thank all the people who contributed to it. My first acknowledgement is for my two advisors, Nir Yosef and Michael Jordan. They have supported and guided my research with passion, dedication, and kindness. Nir, thank you for sharing your passion for biology, for teaching me to trust my intuition and ideas. I also thank you for making our collaboration, remote and across the world, possible. I thank you for giving your best to understand my accent and unshaped ideas at early and late hours, and for your boundless generosity, which has not only supported my research but my overall well-being throughout my PhD.

Mike, I am grateful for your infectious enthusiasm for research and interdisciplinary collaboration, and for demonstrating how far kindness, tolerance, and entrepreneurial spirit can lead. The clarity with which you answer (many) questions about my current and future research or career choices has instilled, I think, a necessary dose of confidence I lacked when I joined Berkeley. The time spent inside and outside your lab has marked my weeks and has been a constant source of joy, from Monday morning group meetings to Friday evening social gatherings. These moments have given my time at Berkeley, marked by the pandemic and remote collaborations across the world, a warmth and a sense of community that I will deeply miss.

I am also thankful for the remarkable friends I have had the pleasure to work with during my time at Berkeley. Romain, I owe my presence at Berkeley to you, and will be forever grateful for your self-determination, and overwhelming kindness and generosity, which I have missed dearly in the latter stages of my PhD. If you read this, please note that I am still waiting for an invite to eat Thau oysters and drink beers with you and your brother in Bédarieux. Stephen, meeting you has been one of the highlights of my PhD. I will always remember the true joy of our first in-person meeting, which started one of many meetings under the sun, discussing conditional independence and multiple testing. I know that our next encounters in Boston won't be as sunny, but they will be just as warm.

Adam, thank you for the love you showed to brainstorm software and probabilistic models in Stanley, and equally important, for your care to ensure I only drank the best coffee in Berkeley. As you probably know, I am carrying the torch, and I am doing my best, daily and sometimes hourly, to make you proud. Although we shared three years of overlap in our time at Berkeley, I wish we had more time to work together, and we still hope we will have the chance to do so in the future.

Justin, it has been an absolute pleasure to work with you, and I wish I knew you'd leave us to New York so soon, I'd have grabbed many more opportunities to seize your true knowledge of San Francisco. Anastasios, your boundless energy and

enthusiasm have been both inspiring and contagious throughout our collaboration, and I already miss our brainstorming sessions in Soda. Can, your friendship has been a true blessing in the latter stages of my PhD, and I hope our paths will cross again soon. The ease with which you went all the way from interning to leading cutting-edge research in single-cell analysis is a true inspiration, and I will do my best to follow your example in the future. I also thank the many colleagues and friends, Tiffany, Sebastian, Bear, Carlos, Gonzalo, and many others, I have made, who made my time at Berkeley an experience going beyond research.

I would also like to thank my family for the constant waves of love they sent me throughout my PhD, and for carrying the price of my long absences. I would like to thank my mother for her boundless love and for her constant support throughout my early life and then studies. Her advice (“tu as obligation de moyens, pas de résultats”) has been a true blessing to go through the ups and downs of my PhD. I am also thankful to my father for his constant support and for the passion he instilled in me for science. I would like to thank my brother for the special moments we share when we finally get to see each other, and for the example he has always been for me. I am also thanking my other siblings, and in particular my sister Amandine, for the true support and love she has kept giving us, across the Atlantic. Eloy, I also need to thank you, for your multiple visits to Berkeley, which have been rays of sunshine, for Mathilde, Zoe and I.

I will forever be thankful to my wife, Mathilde, for her endless support and love. Thank you for being my best friend and the most beautiful person with whom I would like to share my life. I am also thankful to my daughter, Zoe, for the bottomless joy she has brought to our lives.

Contents

Contents	iv
List of Figures	vi
List of Tables	ix
1 Introduction	1
2 An Empirical Bayes Method for Differential Expression Analysis of Single Cells with Deep Generative Models	8
2.1 Introduction	8
2.2 Background	10
2.3 Methods	13
2.4 Results	20
2.5 Discussion	24
2.6 Supplementary Information	27
3 Variable Inference for Variable Selection	45
3.1 Introduction	45
3.2 Methodology and Statistical Framework	48
3.3 Results	57
3.4 Discussion	66
3.5 Supplementary Information	69
4 Multi-Resolution Variational Inference	83
4.1 Introduction	83
4.2 Results	85
4.3 Discussion	99
4.4 Methods	102
4.5 Supplementary Information	113

5	AutoEval Done Right: Using Synthetic Data for Model Evaluation	157
5.1	Introduction	157
5.2	Autoevaluating Accuracy and other Metrics	159
5.3	Evaluating Model Performance from Pairwise Comparisons	167
5.4	Discussion	171
5.5	Code snippets	171
5.6	Experimental details	172
5.7	Additional experiments	174
6	Uncertainty Quantification of Algorithmic Errors onto Cellular Measurements in Spatial Omics	177
6.1	Introduction	177
6.2	Related work	179
6.3	Method	180
6.4	Results	186
6.5	Supplementary Figures	195
6.6	Supplementary Tables	198
7	Conclusion	200
	Bibliography	203

List of Figures

2.1	Overview of lvm-DE	34
2.2	SymSim results	35
2.3	PBMCs results	36
2.4	Batch harmonization on <i>PbmcBench</i>	37
2.5	SARS-CoV2 dataset results	38
2.6	Continuous covariate experiment	39
2.7	SymSim-DA results	40
2.8	MUSCAT results	41
2.9	Inter cell-types analysis for the PBMC dataset	42
2.10	Inter cell-types analysis for the Blish dataset	42
2.11	Within cell-types analysis for the Blish dataset	43
2.12	Predicted LFC correlations with reference RNA data	43
2.13	Effect of molecular cross-validation on LFC estimation	44
2.14	Power and calibration of lvm-DE for varying sample sizes	44
3.1	Correlated variables may be functionally unrelated	46
3.2	VI-VS overview	49
3.3	Application of VI-VS to semi-synthetic data	60
3.4	Examples of gene groups identified by VI-VS in the semi-synthetic experiment	61
3.5	Application of VI-VS to perturb-seq data	62
3.6	Application of VI-VS to CITE-seq data	67
3.7	Application of VI-VS to spatial transcriptomic data	68
3.8	Semisynthetic experiment from a Poisson model	72
3.9	Influence of several parameters of importance in the semi-synthetic experiment	72
3.10	Effect of the development set size of VI-VS	74
3.11	Additional metrics for the perturb-seq dataset	74
3.12	Additional metrics for the CITE-seq dataset	75
3.13	Comparison of significance scores across cell types	75
3.14	Effect of correlation strength on the conditional independence test	79

3.15	Illustration of VI-VS multiresolution testing	80
3.16	Running time comparisons	81
4.1	Overview of MrVI	87
4.2	Application of MrVI to a semi-synthetic experiment	89
4.3	Analysis of a COVID-19 cohort with MrVI	114
4.4	Analysis of a chemical perturbation screen with MrVI	115
4.5	Analysis of a Crohn's disease cohort with MrVI	117
4.6	Functional description of MrVI	118
4.7	Illustration of the multi-head attention mechanism used in MrVI	118
4.8	Supplementary results for the semi-synthetic experiment	119
4.9	SCIB metrics on the COVID-19 experiment	120
4.10	Distance matrix for B cells in the COVID-19 experiment	120
4.11	Comparison of time since symptom onset for the two COVID-19 patient clusters	121
4.12	Comparison of the log density ratio across the different cell types in the COVID-19 experiment	121
4.13	Heatmap of MrVI LFCs for the DE genes identified by in the COVID-19 experiment	122
4.14	Histograms of the number of DEGs for the maximum dose (10000nM) of each drug-cell-line combination with respect to the vehicle for the three cell lines in the sci-Plex dataset	122
4.15	Effect of parameter combinations on MrVI performance on the sci-Plex dataset	123
4.16	Labeled distance matrix for the A549 cell line	124
4.17	Sample distances of all samples in the A549 cell line	125
4.18	MrVI z latent space representation colored by cluster assignment	126
4.19	Elbow plot for the sci-Plex dataset	127
4.20	Visualization of the top DE genes identified by MrVI in the A549 cell line	128
4.21	Analysis of the MCF-7 cell line in the sci-Plex experiment	129
4.22	Analysis of the K562 cell line in the sci-Plex experiment	130
4.23	Copy of Figure 4.21e with row labels for each drug-dose combination.	131
4.24	Sample distances of all samples in the MCF7 cell line	132
4.25	MrVI z latent space representation colored by cluster assignment	133
4.26	Heatmap of LFCs of the top DE genes identified by MrVI in the MCF7 cell line data	134
4.27	Labeled distance matrix for the MCF7 cell line	135
4.28	Sample distances of all samples in the K562 cell line	136
4.29	MrVI z latent space representation colored by cluster assignment	137

4.30	Heatmap of LFCs of the top DE genes according to MrVI over the K562 cell line data	138
4.31	SCIB metrics computed on the IBD dataset	139
4.32	Unbiased analysis of the IBD dataset	140
4.33	Effect sizes of B2 disease behavior split by original cell-type labels	141
4.34	Additional analysis of stromal cells in stenosis	142
4.35	u cell representations using the unsupervised setting of MrVI	143
4.36	Ablation study of MrVI conducted on the semi-synthetic dataset	143
4.37	Admissibility scores vs. held-out reconstruction log probability densities in the COVID dataset	144
4.38	Cartoon of semi-synthetic benchmark construction	144
4.39	Code snippet for the use of MrVI in a single-cell analysis	153
5.1	Imagenet experiment	164
5.2	Protein fitness experiment	165
5.3	Effective sample size against annotator model performance in the protein fitness experiment	167
5.4	LLM experiment	170
5.5	Effective sample size against annotator model performance in the LLM experiment	170
5.6	Interval metrics for the different experiments	175
6.1	Visualization of the approach	180
6.2	Example of images used for validation-based annotation	185
6.3	Lung experiment	193
6.4	Colon experiment	194
6.5	Canonical marker gene expression across cell-types in the lung dataset . .	195
6.6	Visualization of cells in the main lung slide	195
6.7	Cells with their manual annotations in the main lung slide	196
6.8	Visualization of cells in the main colon slide	197

List of Tables

2.1	Ablation study	29
2.2	scVI ablation study	30
2.3	ScSphere ablation study	30
2.4	Running times on the Blish dataset	33
3.1	Study of the linear generative model	73
3.2	Comparison of the number of significant hits across methods	76
4.1	Execution times of MrVI on the COVID-19 dataset	153
4.2	Integration metrics of MrVI	154
4.3	Sensitivity analysis for key hyperparameters of MrVI	156
5.1	Overview of the evaluated language models	174
5.2	Running times for the different experiments	175
6.1	Markers employed to label clusters	198
6.2	Cell counts from ML-derived segmentation and classification	198
6.3	List of non-T cell marker genes	199

Chapter 1

Introduction

Evolution of computational methods for single-cell omics

Closely followed by modeling development, single-cell omics technological improvements have provided considerable help to study gene expression variation at the single cell level [1]. Early single-cell applications, typically focusing on transcriptomics, mapped cell states and phenotypes within complex tissues from a few biological replicates [2–4]. In immunology, for instance, these efforts enabled the construction of immune cell atlases, which not only confirmed known cell states but also led to the discovery of novel cell populations [5, 6]. These atlases provided an unprecedented mapping of the molecular content of immune cell types and their partition into stable (and sometimes transient) states, whose realization proved critical for studying how immune cells develop [7], respond to pathogens, and interact with their tissue environments [8].

In the past few years, single-cell omics have undergone an expansion in scale and scope, enabling population-level studies and multi-omic profiling [9, 10]. Among the many factors allowing this expansion, multiplexing has enabled the parallel sequencing of hundreds of thousands of cells across hundreds of samples [11] and the concurrent profiling of multiple omic modalities within individual cells [12, 13]. These advances have facilitated a deeper understanding of cellular heterogeneity across biological conditions. Cohort-scale single-cell omic studies are now being used to investigate disease mechanisms across many individuals, and understand variability in treatment response [14, 15]. Highlighting the relevance of these approaches, numerous scRNA-seq studies characterized immune responses to SARS-CoV-2 infection and identified cellular changes associated with disease severity and long COVID, offering strategies

for therapeutic interventions [16, 17].

In parallel, preprocessing and analysis computational methods have co-evolved with these technological improvements to capture relevant biological variation contained in these data while correcting for technical and biological confounders. For instance, after alignment, canonical preprocessing of single-cell transcriptomic data requires sequential steps [18]. Quality control aims to remove measurements likely to be affected by technical artifacts. Normalization adjusts for differences in sequencing depth across cells. Batch correction removes variation in measurements linked to experimental design rather than biological variation. Dimensionality reduction produces low-dimensional cell representations that capture most of the biological signal of interest. Clustering groups cells into coherent populations that are likely to share similar transcriptional profiles that are then annotated, either manually or automatically, using predictive models.

Around the beginning of my PhD, the field increasingly recognized the limitations of such sequential preprocessing steps [19]. In particular, the difficulty of correcting for technical variation while preserving relevant biological information posed a challenge to the analysis of larger-scale single-cell omics data [20]. Efforts intensified to develop integrated models designed to learn a comprehensive representation of the data that simultaneously accounts for technical factors and biological structure, effectively performing multiple steps within one framework [21, 22].

Latent variable models (LVMs) have been a predominant approach to integrate multiple preprocessing steps within a single framework. Typically implemented as deep generative models [23], they operate directly on raw data rather than requiring separate normalization steps. They model technical and biological confounders, e.g., batch effects, sequencing depth, or cell cycling events, as covariates of the probabilistic model to learn low-dimensional latent representations of cells where biological signal is disentangled from technical variation. For this reason, latent variable models have become essential components of single-cell omics analysis pipelines [24], applied in a wide range of settings, including transcriptomics [22, 25–27], epigenomics [28], spatial transcriptomics [29], and other multi-omic settings [30, 31].

Thesis contributions

Despite their success for tasks including data integration, annotation, or imputation, the application of LVMs to core statistical inference tasks, such as differential expression (DE) analysis [32, 33], or association testing [34], remained surprisingly underdeveloped. This gap was notable given that the scalability and flexible struc-

ture of deep generative models allow them to accurately capture complex biological variations and technical noise, overcoming limitations of simpler methods prone to misspecification [35], while their Bayesian formulation provides a natural framework for principled hypothesis testing [36].

In this context, the early efforts of my PhD concentrated on developing solutions to conduct statistical hypothesis testing with latent variable models. **Chapter 2** details the first contribution of my PhD, *lvm-DE*, introducing a framework to conduct DE analysis with latent variable models on single-cell transcriptomic data. Early efforts leveraging latent variable models for DE analysis often employed heuristics or significance measures like Bayes factors. Interpreting the resulting discoveries was challenging, either because they did not control for standard error rates (e.g., the false discovery rate (FDR) [37]), or because they identified many genes with small effect sizes. *lvm-DE* addressed these limitations by identifying DE genes based on interpretable posterior distributions of log fold change (LFC) via composite null hypotheses to prioritize meaningful effect sizes while controlling the FDR.

Identifying DE genes is often the first step in analyzing single-cell data. Researchers frequently follow this with post-hoc interpretation methods, such as enrichment analysis [38], to infer underlying biological processes from the list of DE genes. While valuable, these knowledge-based approaches are inherently limited by the scope and accuracy of existing biological annotations. Concurrently, the increasing availability of multi-omic technologies (e.g., CITE-seq, measuring gene expression and surface proteins simultaneously) presented an opportunity to move beyond such post-hoc interpretations towards direct, data-driven discovery of molecular relationships within cells and across molecular layers. Exploring how gene expression might shape cellular function through downstream protein expression, for instance, requires methods capable of rigorously identifying associations directly from the multi-modal data. This interest in identifying candidate functional associations directly from multi-omic data, necessitating robust association testing methodologies, motivated the development of the following method.

Chapter 3 details *VI-VS*, introducing a method to identify candidate functional associations between molecular or cellular features from multi-omic data via conditional independence testing. Based on the conditional randomization test [39], *VI-VS* offers a statistically rigorous approach to identify conditional dependencies between features while controlling the false discovery rate. Leveraging latent variable models, it makes minimal assumptions on the data or the nature of the potential interactions, such as linearity.

In parallel to the development of multi-omics, the field saw during my PhD the emergence of large-scale, multi-sample single-cell studies. Analyzing these datasets,

where biological samples (subjects, perturbations, organs, etc.) rather than individual cells are the primary units of variation, required rethinking analysis strategies. A common approach for analysis involved performing differential expression or abundance analysis within predefined cell clusters and across predefined groups of samples [40], e.g., cases vs. controls. Relying on predefined cell clusters, however, is a key limitation, as difficulty in defining appropriate cluster granularity can dilute DE/DA signals specific to certain cell subsets or continuous variations [41].

A more fundamental challenge arises in determining how best to compare biological samples themselves. Predefined groupings, e.g., simple case versus control comparisons, may not always capture the most pertinent axes of variation, potentially obscuring or oversimplifying true sample heterogeneity. A central goal of large-scale cohort studies is consequently not only comparative but also *exploratory*: to challenge and refine our understanding of sample-level heterogeneity by uncovering more nuanced sample groupings directly from the data. Such data-driven stratification can be particularly insightful when prior metadata is limited or when established classifications can be refined, as demonstrated in pan-cancer studies where molecular signatures improved upon traditional histopathological categories [42].

Another argument supporting the need for data-driven sample stratification is the observation that the nature of sample-level variation can be context-dependent, varying across different cell populations within the samples. Identifying these cell-state-specific stratifications is crucial for a comprehensive understanding of complex biological systems. As illustrated in recent studies of COVID-19 [16], analyses focused on distinct cell lineages, such as T cells versus myeloid cells within the same cohort, have revealed divergent sample stratifications and biological interpretations.

Chapter 4 details the third contribution of my PhD, *MrVI*, a probabilistic framework designed for the exploratory and comparative analysis of large-scale, multi-sample single-cell data. Addressing the limitations of methods reliant on predefined cell clusters or sample groups, MrVI enables annotation-free differential expression and abundance analysis at high resolution. It automatically identifies relevant sample groupings and cellular contexts associated with sample-level variation, while accounting for uncertainty and technical confounders.

As illustrated by the work presented in this dissertation, addressing the analytical challenges posed by increasingly complex single-cell datasets required substantial methodological innovation. This need, coupled with the exponential growth in data scale [43], fueled a rapid expansion in the number of computational tools available for omics data analysis during my PhD. In 2023, at least 1,400 single-cell analysis tools were available, and the field shows no signs of slowing down [44, 45].

This proliferation of methods has created significant challenges for rigorous model

evaluation and selection [46]. In omics, the lack of ground-truth, which is often not available for real data [47], exacerbates the difficulty of model evaluation.

Motivated by these evaluation challenges, and in particular ground-truth scarcity in omics, **Chapter 5** describes *AutoEval*, a method designed to enhance the power of model evaluation and selection in these scenarios. Building on prediction-powered inference [48], *AutoEval* leverages a small set of annotated data to estimate and correct for potential biases in a large set of synthetically-labeled data, thereby improving the power of model selection without compromising the statistical validity of inference.

The final research area explored during my PhD addresses the challenge of understanding how cellular function is influenced by spatial context within tissues. Spatial transcriptomics assays provide the means to study the influence of spatial context on cell states by measuring gene expression *in situ* within tissues. Analyzing these data to identify spatially-influenced differential expression (DE) critically relies on accurate upstream processing, particularly cell segmentation, which assigns transcripts to individual cells [49, 50]. However, a growing concern is that systematic errors in automated segmentation introduce significant biases leading to spurious discoveries in spatial DE analyses [51]. Addressing this problem requires methods to quantify the downstream impact of these preprocessing errors.

Chapter 6 details the fourth contribution of my PhD, presenting a framework designed to quantify the impact of segmentation errors on spatial DE analysis. Using limited and high-quality ground-truth annotations, this work introduces a statistical framework to quantify how segmentation errors influence spatial DE, producing more reliable estimates of differential expression across spatially-resolved cell populations.

Related contributions

In addition to the core components of my PhD described above, I would also like to highlight the following contributions that closely relate to the subject of this dissertation.

Decision-Making with Auto-Encoding Variational Bayes To make decisions based on a model fit with auto-encoding variational Bayes (AEVB), practitioners often let the variational distribution serve as a surrogate for the posterior distribution. This approach yields biased estimates of the expected risk and therefore leads to poor decisions for two reasons. First, the model fit with AEVB may not equal the underlying data distribution. Second, the variational distribution may not equal the posterior distribution under the fitted model. We explore how fitting the variational distribution based on several objective functions other than the ELBO, while continuing to fit the

generative model based on the ELBO, affects the quality of downstream decisions. For the probabilistic principal component analysis model, we investigate how importance sampling error, as well as the bias of the model parameter estimates, varies across several approximate posteriors when used as proposal distributions. Our theoretical results suggest that a posterior approximation distinct from the variational distribution should be used for making decisions. Motivated by these theoretical results, we propose learning several approximate proposals for the best model and combining them using multiple importance sampling for decision-making. In addition to toy examples, we present a full-fledged case study of single-cell RNA sequencing. In this challenging instance of multiple hypothesis testing, our proposed approach surpasses the current state of the art [52].

DestVI Identifies Continuums of Cell Types in Spatial Transcriptomics

Data Most spatial transcriptomics technologies are limited by their resolution, with spot sizes larger than that of a single cell. Although joint analysis with single-cell RNA sequencing can alleviate this problem, current methods are limited to assessing discrete cell types, revealing the proportion of cell types inside each spot. To identify continuous variation of the transcriptome within cells of the same type, we developed Deconvolution of Spatial Transcriptomics profiles using Variational Inference (DestVI). Using simulations, we demonstrate that DestVI outperforms existing methods for estimating gene expression for every cell type inside every spot. Applied to a study of infected lymph nodes and of a mouse tumor model, DestVI provides high-resolution, accurate spatial characterization of the cellular organization of these tissues and identifies cell-type-specific changes in gene expression between different tissue regions or between conditions [29].

scvi-tools: A Python library for probabilistic analysis of single-cell omics

data Methods for analyzing single-cell data perform a core set of computational tasks. These tasks include dimensionality reduction, cell clustering, cell-state annotation, removal of unwanted variation, analysis of differential expression, identification of spatial patterns of gene expression, and joint analysis of multi-modal omics data. Probabilistic models provide principled ways to capture uncertainty in biological systems and are convenient for decomposing the many sources of variation that give rise to omics data. However, several obstacles impede their community-wide adoption. The first obstacle, coming from the perspective of the end user, relates to the difficulty of implementing and running such models. A second obstacle relates to the development of new probabilistic models. From the perspective of developers, there are many necessary routines to implement in support of a probabilistic model,

including data handling, tensor computations, training routines, and the underlying optimization, sampling and numerical procedures. To address these limitations, we present scvi-tools, a Python library for deep probabilistic analysis of single-cell omics data [53].

CherryML: Scalable Maximum Likelihood Estimation of Phylogenetic Models Phylogenetic models of molecular evolution are central to numerous biological applications spanning diverse timescales, from hundreds of millions of years involving orthologous proteins to just tens of days relating to single cells within an organism. A fundamental problem in these applications is estimating model parameters, for which maximum likelihood estimation is typically employed. Unfortunately, maximum likelihood estimation is a computationally expensive task, in some cases prohibitively so. To address this challenge, we here introduce CherryML, a broadly applicable method that achieves several orders of magnitude speedup by using a quantized composite likelihood over cherries in the trees. The massive speedup offered by our method should enable researchers to consider more complex and biologically realistic models than previously possible. Here we demonstrate CherryML’s utility by applying it to estimate a general 400 *imes* 400 rate matrix for residue–residue coevolution at contact sites in three-dimensional protein structures; we estimate that using current state-of-the-art methods such as the expectation-maximization algorithm for the same task would take >100,000 times longer [54].

Chapter 2

An Empirical Bayes Method for Differential Expression Analysis of Single Cells with Deep Generative Models

2.1 Introduction

Single-cell omics measurements promise to resolve the heterogeneity of cellular identities, characterizing subtle molecular changes between cells [8]. In particular, the measurement of gene expression in single cells—with single-cell RNA-sequencing (scRNA-seq)—creates opportunities to characterize differences in complex scenarios such as rare cell-type detection or inter-cell-type/inter-condition comparative analyses. However, technical factors such as the sequencing depth and batch effects obscure the underlying biological signal, requiring careful probabilistic modeling of the data to allow accurate detection of differentially expressed genes.

Many differential expression (DE) methods have been developed to handle the specificity of scRNA-seq data [55–57], often building on earlier work on bulk RNA-seq [32, 33]. However, there is a gap between the way these methods are applied and the structural characteristics of modern datasets. First, the performance of these approaches in modern datasets is highly dependent on their ability to correct for batch effects [58]. Indeed, a large number of these approaches rely on generalized linear models (GLMs), while recent benchmarking studies in scRNA-seq data integration focus on nonlinear methods (e.g., MNNs [20]) for their evaluation [24]. The linear method present in that benchmarking study, Combat [59], was found to be suitable

only for simple scenarios. Second, most methods for DE analysis are unable to take advantage of the large sample sizes of current scRNA-seq experiments in learning their underlying gene expression model. Indeed, DE methods are often run on pairs of cell groups, subsampled from the original data. Gene expression level estimates could be improved by leveraging shared information across the whole dataset for model fitting, e.g., to refine given gene expression levels from gene-gene correlation structures across batches.

Recent advances in deep generative modeling [23] have led to the development of a range of methods for normalization and analysis of gene expression data at scale, able to deal with millions of cells and able to account for complex study designs [60]. Most of these models (e.g., scSphere [25], scVI [22], scANVI [26], scGen [27] and CellBender [61]) represent the expected gene expression in cell n as a nonlinear function of a latent low-dimensional summary of the cell’s state z_n and its batch identifier s_n . Such approaches intrinsically lead to information sharing across cells and across genes and promise to address the aforementioned issues. However, little attention has been given to carrying out rigorous and accurate DE analyses with those models. Two exceptions are scVI [22] and scANVI [26] in which the uncertainty in the latent variable z_n is used to approximate a Bayes factor and detect differential expression. However, the Bayesian approach studied in [22, 26] has three fundamental limitations. First, the underlying definition of DE genes is too sensitive and may tag genes with negligible differences in expression and often of limited biological relevance. Second, interpreting Bayes factors is challenging, because of the difficulty in translating them to established multiple hypothesis testing metrics, e.g., false discovery rate (FDR). Third, scVI proposes to average Bayes factors from randomly sampled pairs of cells from each group, which lacks theoretical justification.

After presenting background information on differential expression and deep generative models (Section 2.2), we introduce lvm-DE, a general Bayesian framework for detecting differential expression that addresses the aforementioned issues (Section 2.3). lvm-DE takes as input a fitted deep generative model of scRNA-seq data, a pair of cell groups and a target α , and provides as output an estimate of the log fold change for every gene, as well as a list of DE genes, with FDR control at level α . Our Bayesian formulation of differential expression hypothesis testing use a composite alternative, built from the log fold change (as in DESeq2 [33]) to avoid detecting spurious or weakly expressed genes. We benchmark lvm-DE applied to scVI [22] and scSphere [25] against classical differential expression approaches on simulated and real-world datasets (Section 2.4). Our results demonstrate that lvm-DE is a valuable tool for understanding differential expression effects in scRNA. An open-source implementation of lvm-DE is available in beta as part of the scvi-tools Python package [53] (<https://github.com/scverse/scvi-tools>).

2.2 Background

Before introducing our statistical framework, we describe two lines of work in scRNA-seq data analysis on which our work builds: differential expression and deep generative modeling.

Differential expression

Differential expression analysis of transcriptomics data aims to determine which genes have significantly different expression levels between two groups of cells A and B (e.g., representing different cell types or cells obtained under different experimental conditions). It is known that the direct approach of comparing counts across groups is subject to noise and bias because of the various sources of technical factors that confound the data, including variation in library size and batch effects [62]. To address this, all DE methods define (either implicitly or explicitly) a model with a latent variable h_g^c that represents the *denoised* or *normalized* expression level for each gene g and each condition $c \in \{A, B\}$. Differential expression for gene g between condition A and B is then assessed by comparing some statistic measuring the differences between the distributions of h_g^A and h_g^B . Finally, the significance of these differences is corrected to account for the multiplicity of hypotheses (one per gene) by estimating the false discovery rate (FDR).

Several popular DE frameworks for scRNA-seq data were originally developed for bulk RNA-seq data. Those methods are still competitive although not explicitly designed for scRNA-seq [63]. Two of the most prevalent methods, edgeR [32] and DESeq2 [33] model gene expressions as negative binomial distributions. Another widely used method, Limma-voom [64], uses log-transform counts instead. A common theme for all three methods is that they rely on generalized linear models (GLM), making it convenient to represent and control for possible biases (e.g., due to batch effects or GC content). Although these GLMs are fit for every gene separately, it is important to note that their training procedures pool information across genes, relying on mean-variance relationships and empirical Bayes methods to improve the fit of some of the parameters. Another key contribution, featured in DESeq2, is to propose a *composite* null hypothesis in which the hypotheses are formulated in terms of the fold change being higher than some threshold. This results in exclusion of genes whose log fold change (LFC) is low, even if the differences in their expression are statistically significant. In practice, this procedure helps exclude genes with little biological interest.

Another set of DE analysis frameworks have been designed specifically for scRNA-seq data, with popular methods including MAST [55], SCDE [56], and scDDboost [65].

MAST uses a hurdle model that was designed to account for the abundance of so-called dropouts in scRNA-seq data; these are zero counts that are observed mostly with weakly expressed genes, due to limitation in RNA capture efficiencies. SCDE takes a conceptually similar approach and employs a Bayesian framework to model scRNA-seq data as a mixture of negative binomial and Poisson distributions. The negative binomial component of the mixture accounts for counts that are attributable to gene expression while the Poisson component models dropout events [66]. For within-cell-type DE analyses across conditions, scDDboost employs an empirical Bayes approach to detect DE from compositional changes within a cell type from subtype assignments obtained from clustering. In particular, it posits an empirical Bayes prior for subtype counts and capture differences in gene expression within subtypes from a negative binomial model. In the following, we compare our approach for DGM-based DE against this collection of DE methods, coming from both bulk (DESeq2, edgeR, limma-voom) and the scRNA-seq literature (MAST).

Deep generative models for scRNA-seq analysis

Given the steadily growing sizes of datasets [67], recent work has begun to adopt more flexible and nonlinear approaches for modeling gene expression in single cells. While these methods require more data for training, they are often complemented by scalable inference procedures. One popular line of work relies on advances in deep generative modeling [60], in which models make use of deep neural networks as link functions in their conditional distributions, see, e.g., [22, 25–27, 61, 68–73].

Most of these methods fit a hierarchical probabilistic model (i.e., a graphical model), that includes a low-dimensional latent variable embedding each cell in some cell state space.

To ensure that these cell embeddings reflect biological variation as much as possible, and remain disentangled from technical variation, a common practice is to model nuisance factors as parameters or random variables within the generative model. One factor that is commonly accounted for is variation between cells due to sequencing depth. For instance, scVI [22] and scANVI [26] models include a random variable that represents the sequencing depth of each cell. This variable is used as a scaling factor, thus providing normalized expression levels. The hierarchical model also links the latent representation of cell state directly to the normalized expression values, thus decoupling the cell embeddings from variations in sequencing depth. An alternative approach, scSphere [25], uses the sum of observed transcripts as a deterministic scaling factor, but relies on an additional penalty term to ensure that the cell embeddings are decoupled from it. Deep generative models provide accommodating strategies to correct for other nuisance factor(s) present in single cells. Multiple approaches,

including scVI, scANVI, or scPhere assume model normalized gene expression as a function of both cell state embeddings and these nuisance factors. This function learned during the fitting process and parameterized as a multi-layered perceptron can model intricate nonlinear nuisance effects on gene expression. Such nuisance factors can include batch or patient identity in multi-donor datasets, but have also been generalized to other covariates such as cell cycles [53].

Distributional assumptions are other key components that can improve cell-state representation and gene expression modeling. For example, in scGen [27], the counts are normalized, log-transformed and treated as samples from a normal distribution. Conversely, both scVI [22] and ScPhere [25] model gene expression counts under a negative binomial likelihood. Other models (e.g., the original work of scVI) include the addition of an additional low-magnitude term, which aims to handle zero inflation in the data, due to limitations in sensitivity and the effects of transcription bursting discussed in more depth in [74, 75]. Another component of importance is the modeling of the cell embedding. DCA [70] relies on a deterministic embedding, treated as a parameter optimized using maximum likelihood estimation. Another line of work, variational autoencoders (VAEs), posits a prior distribution on the embedding that regularizes the embedding space and aims to learn more useful representations [23]. While most unsupervised VAEs posit a non-informative isotropic Gaussian prior, alternative priors may improve cell representation in specific contexts. For instance, scANVI [26] leverages partial data annotations to learn more structured priors on the latent variable. scPhere embeds biological states in a hyper-spherical space with a uniform prior, which could improve the modeling of rare cell types and hierarchical structures.

While most applications of deep generative models in single cells revolve around the cell state space, few approaches consider normalized gene expression values and their uncertainty, that are central for DE. Aiming in this direction, DCA feeds denoised gene expression predicted by the model as inputs for DESeq2 instead of raw counts to better reduce the noise present in scRNA-seq datasets. This is unsuitable from a computational perspective because DESeq2 may be too long to run when dealing with groups of tens of thousands of cells, creating a bottleneck. Also, as DCA relies on an autoencoder, it does not provide uncertainty estimates on normalized gene expression, and summarizes the data via its mean instead. This is known to potentially create spurious signal in other downstream applications (e.g., gene-gene correlation estimation [76]).

scVI and scANVI [22, 26] both provide integrated tools to perform differential expression between two groups. First, expected expressions are sampled from the posteriors of representative cells. These values are then subject to Bayes factor computation to assess significance. This framework has several limitations. First,

although Bayes factors are an appealing alternative to frequentist hypothesis testing for Bayesian models, they may be hard to interpret while analyzing real-world data and are difficult to combine with frequentist false discovery rate control guarantees. Second, these approaches annotate DE genes as significant without considering the effect size (i.e., LFC), which can be extremely low for lowly expressed genes, which are relatively abundant in scRNA-seq data. Third, this approach is convenient since model fitting is performed only once using all cells, thus avoiding the need to refit a new model for each DE test (each considering different pairs of cell subsets). However, scVI does not explicitly account for cell type in its generative model. As a heuristic, scVI aggregates cells by averaging approximate Bayes factors across randomly sampled pairs of cells to compare cell populations. We provide more theoretical insight and an improved approach in this manuscript.

2.3 Methods

We introduce lvm-DE, a principled framework to enable the use of deep generative models as a means for quantifying and assessing the significance of differences in gene expression (Figure 2.1A). lvm-DE takes as input a latent variable model, a scRNA-seq dataset with covariates (such as batch identifiers) and two groups of cells to be compared, as well as a significance level α . lvm-DE provides an estimate of the LFC of each gene between cells of the two groups. It also provides a list of significantly differentially expressed genes that controls the FDR at level α . These estimates are derived while controlling for differences in library sizes and for variability due to observed technical confounders, such as the association of samples with different batch identifiers.

lvm-DE comprises five steps (Figure 2.1B). First, a model is fitted, such that the observed expression values in each cell (X) are represented by a low-dimensional latent random variable (often denoted as z ; in VAE-based models such as scVI, this amounts to learning an encoder network). This representation provides a summary of the state of each cell, while controlling for differences due to batch effects (s) and sequencing depth, but it does not use the annotation of cells into the two groups that we wish to compare (Figure 2.1B1, Section 2.3). Performing DE with an annotation-agnostic model is convenient as it does not require refitting the model when we want to compare different groups of cells. Second, annotations are used to build a posterior distribution of the latent representation of each of the two groups, providing a summary of their characteristic cell states (Figure 2.1B2, Section 2.3). Third, cell states posteriors are used to infer a distribution of normalized expression for every gene (Figure 2.1B3) in each cell group. These high-dimensional distributions are

generated by propagating the low-dimensional representations through the latent variable models (LVM). In VAE-based models, this amounts to applying the decoder network. Fourth, the normalized expression values are used to estimate the LFC and a posterior probability for the LFC to be larger than a given cutoff δ (Figure 2.1B4, Section 2.3) while correcting for batch effects (Section 2.3). Finally, these posterior probabilities of DE are used to call genes whose significance is within a desired FDR α (Figure 2.1B5, Section 2.3).

Selection of deep generative models for lvm-DE

Our formulation of lvm-DE is general, such that it applies broadly to many types of deep generative models, and can operate as a wrapper function for differential expression with these models. In the following, we provide the modeling prerequisites for using lvm-DE and then discuss additional considerations of modeling choices that can lead to better accuracy.

Let x_{ng} be the observed numbers of transcripts in cell n that originate from each gene g and let s_n correspond to the batch annotation. A general form of generative process that qualifies for use by lvm-DE is as follows. First, a low-dimensional latent variable z_n representing a cell’s biological state is sampled from a prior distribution. A neural network then takes a sample from z_n and the batch information s_n and maps it to another latent variable, h_{ng} , representing the underlying (i.e., normalized) expression level for gene g . Finally, the observations x_{ng} are drawn from model-specific conditional likelihood distributions, such as negative binomial or log-normal, with or without zero inflation [22, 27, 77]. The likelihood of a single observation (cell) in these models decomposes as follows:

$$p(x_n, h_n, z_n \mid s_n) = p(z_n) \prod_{g=1}^G p(x_{ng} \mid h_{ng}) p(h_{ng} \mid z_n, s_n). \quad (2.1)$$

Furthermore, lvm-DE assumes that there exists a tractable approximation to the posterior $p(z_n \mid x_n)$ [23, 78]. While many published methods satisfy this basic prerequisite, here we selected two such models—scVI and scPhere [22, 25]—both of which are based on a VAE but differ in their loss function and distributional assumptions. These two models employ variational approximations to the posterior. In particular, they fit distributions $q(z_n \mid x_n)$, whose parameters are functions of the input data x_n , and which are learned during model fitting.

Beyond this basic prerequisite, we identified two considerations for improved performance (implementation details in Supplement B). The first consideration is the need to model the normalized expression levels h_{ng} , while controlling for nuisance

variation in sequencing depth. In scVI and scPhere, this is addressed by modeling the gene expression count x_{ng} as a negative binomial distribution with mean parameter $l_n h_{ng}$, where l_n is a normalizing factor that can be interpreted as the inferred library size. The second consideration stems from the fact that lvm-DE exploits the variational distribution (i.e., an approximation of $p(z_n | x_n)$) for Bayesian hypothesis testing. In this setting, it has been reported that VAE trained with the classical evidence lower bound (ELBO) may underestimate the variance of the variational posterior and thus underperform in tasks of decision making, compared to models trained with alternate variational bounds [52]. Consequently, we train scVI and scPhere using the importance-weighted evidence lower bound (IWELBO), which has been demonstrated to lead to improved performance in other application domains [79].

Aggregating individual cell information to capture population expression levels

lvm-DE takes as input a stratification of cells into groups ($c_n \in \{1, \dots, K\}$, representing, e.g., different cell types or different conditions under which cells were captured), along with a generative model that conforms with Equation 2.1.

To estimate the distribution of gene expression in every group of cells, we assume that there exists an oracle function Φ mapping a point in latent space z to the corresponding covariate $c = \Phi(z) \in \{1, \dots, K\}$. This is a reasonable assumption because the low-dimensional representation given by scVI or other deep generative models often successfully stratifies cells according to their groups [22].

Given a particular value C of the covariate c (e.g., a cell type or a condition), we denote by h_g^C the normalized expression level of gene g in condition C . We define h_g^C , conditioned on a given batch s , as

$$p(h_g^C | s) = \int p(h_g | s, z) p(z | Z \in \Phi^{-1}(C)) dz, \quad (2.2)$$

where $p(z | Z \in \Phi^{-1}(C))$ denotes the prior density $p(z)$ truncated to the preimage of $\{C\}$ under the oracle function Φ . For any latent code z and batch s , it is assumed possible to sample from $p(h_g | s, z)$ using the generative model. Hence, our goal in the rest of this section is to define a strategy to sample from the left-hand side of the integral, $p(z | Z \in \Phi^{-1}(C))$. Because Φ is inaccessible to us, we may estimate the prior density from the data, in the spirit of empirical Bayes methods. More precisely, we first approximate it using the posterior of the annotated data:

$$p(z | Z \in \Phi^{-1}(C)) \approx \int p(z | x) p(x | c = C) dx. \quad (2.3)$$

Second, we estimate this quantity using the truncated dataset with cells belonging to condition C , yielding the mixture density

$$p(z \mid Z \in \Phi^{-1}(C)) \approx \hat{p}_{\mathcal{N}_C}(z) = \frac{1}{|\mathcal{N}_C|} \sum_{n \in \mathcal{N}_C} p(z \mid x_n), \quad (2.4)$$

where $\mathcal{N}_C$ is the set of cell indices that belong to the condition C . To reduce the possible sensitivity to outlier cells, we developed a procedure to identify and remove these from the set $\mathcal{N}_C$. To do so, assuming that the latent mean representations within a cluster are normally distributed, we estimate the covariance matrix from posterior mean samples $\{\mathbb{E}_{p(z|x_i)}[z] \mid i \in \mathcal{N}_C\}$, and remove observations whose variational mean falls outside the 90%-confidence ellipse described by the covariance estimate. As demonstrated previously, the empirical prior appearing in Equation 2.4 is intractable. A first solution is to replace the individual posterior distributions $p(z \mid x_n)$ with their variational approximations $q_\phi(z \mid x_n)$, yielding the so-called plugin estimator

$$\hat{p}_{\mathcal{N}_C}^{\text{plugin}}(z) = \frac{1}{|\mathcal{N}_C|} \sum_{n \in \mathcal{N}_C} q_\phi(z \mid x_n). \quad (2.5)$$

A remaining caveat of this estimator is that it lends equal importance to each sample from the cell population and of the variational distribution. This is in a way similar to the original scVI, in which pairs of cells are sampled uniformly at random from the two populations and compared to each other. It is reasonable to expect, however, that in practice some samples may be of significantly better quality, which explains the previously reported increased performance from using importance-sampling-based estimators [52]. Based on this observation, we use the mixture prior from Equation 2.4, approximated by self-normalized importance sampling (SNIS) with the mixture of Equation 2.5 as the proposal distribution. In this formulation, we first sample $z_1, \dots, z_N \sim \hat{p}_{\mathcal{N}_C}^{\text{plugin}}$, and the contribution of each sample z_i is proportional to:

$$w_i \propto p(z_i) \frac{\sum_{n \in \mathcal{N}_C} p(x_n | z_i) / p(x_n)}{\sum_{n \in \mathcal{N}_C} q(z_i \mid x_n)}, \quad (2.6)$$

where $p(z_i)$ denotes the prior density for z_i , and the intractable evidence terms $p(x_n)$ are estimated with IWELBO estimators ($K = 5,000$ particles). The final samples $\tilde{z}_j$ correspond to random samples with replacement from $\{z_i, i \leq N\}$, where each entry i of the set has weight $p_i = w_i / \sum_j w_j$ [79], such that

$$\begin{aligned} \tilde{z} &\sim \text{Categorical}(\mathbf{z}, \mathbf{p}), \\ \mathbf{z} &= \{z_1, \dots, z_N\}, \mathbf{p} = \{p_1, \dots, p_N\}. \end{aligned} \quad (2.7)$$

Consequently, samples from z that have high likelihood levels across a large pool of cells from $\mathcal{N}_C$ will have higher weights in the posterior, thus further improving the robustness to outliers and to misassignment of cells into groups. In order to ensure that the importance sampling procedure is sufficiently stable, several diagnosis tools such as the Pareto smoothed importance sampling (PSIS) estimator [80] may be employed.

Once we have samples from the distribution $p(z \mid Z \in \Phi^{-1}(C))$, we may follow Equation 2.2 to link these samples to the normalized expression levels h_g^C for DE analysis.

Sampling population expression levels To access $p(h_g^C \mid s)$, it hence suffices to sample latent codes z from the variational posteriors $q_\phi(z \mid x)$, to reweight them via Equation 2.6, and to decode them using $p(h_g \mid s, z)$ using the generative model of choice. In the case of scVI and scPhere, expression levels h and cell states z are deterministically linked by a neural network $h = f(z, s)$.

Fold change estimation for DE

A practical limitation of DE analysis is the identification of genes in which the differences in expression are significant and yet of low magnitude (low LFC), and thus of less or no interest. To address this, lvm-DE treats the LFC itself as a random variable and uses a Bayesian analysis to retain only genes in which the effect size is significantly larger than a cutoff.

To quantify our confidence that the LFC is beyond a desirable level in a given batch s , we introduce the effect size random variable,

$$r_{A,B}^g = \log_2 \frac{h_g^A + \epsilon}{h_g^B + \epsilon}, \quad (2.8)$$

where $h_g^A \sim p(h_g^A \mid s)$, $h_g^B \sim p(h_g^B \mid s)$ are approximated using the SNIS procedure described above. Here, ϵ is a small offset used for stability. When gene g has very low expression levels in both conditions, the pseudo-count ϵ ensures that the associated LFC has a low magnitude [33, 81]. Details regarding the choice of ϵ can be found in Appendix D.

To test for differential expression in lvm-DE, we introduce three models for which gene g is either up-regulated ($\mathcal{M}_+^g$), down-regulated ($\mathcal{M}_-^g$) or not differentially expressed ($\mathcal{M}_0^g$) in population A compared to B . These three models are defined via the change variable $r_{A,B}^g$ as follows:

$$\mathcal{M}_+^g : r_{A,B}^g > \delta, \mathcal{M}_-^g : r_{A,B}^g < -\delta, \text{ and } \mathcal{M}_0^g : |r_{A,B}^g| \leq \delta, \quad (2.9)$$

where δ is a threshold ensuring that the selected genes are interesting in practice. We discuss heuristics for the choice of δ in Appendix E. The probability of each of those models may be estimated from the data, using the empirical Bayes method and the SNIS estimator. In particular, we note:

$$p^g := \max [p(\mathcal{M}_+^g | s), p(\mathcal{M}_-^g | s)], \quad (2.10)$$

the probability that gene g is DE (i.e., with absolute LFC greater than δ).

The LFC is a widely used measure of effect size for expression measurements because of its interpretability [18, 33, 82]. However, our framework could be extended to any difference measure between expression levels, including logit fold change for DNA methylation, as in scMET [83].

Batch effect correction

Our exposition in the previous section assumed that all the cells come from a single batch s . However, DE analyses are often required in multi-batch or multi-donor settings.

Two scenarios, reflecting different perspectives on what *batches* refer to, appear when performing differential expression, depending on whether all batches contain cells from both conditions.

Comparing subpopulations in overlapping batches In the first scenario, the two compared subpopulations A and B have cells in overlapping batches. In the most natural setting, such as a biological replicates, we expect a representation of all the compared conditions in all batches. This often is the case when batches refer to biological replicates, or to different technologies; we then hope to detect DE genes reproducible across all the batches. In this case, let S be any set of batch indices for which there exists at least one observation of each condition. Following our empirical Bayes approach, we define batch-specific empirical priors relying on the subset of observations $\mathcal{N}_C^s = \{n \mid s_n = s, c_n = C\}$. Then, the probability of each of the three models may be computed by averaging across batches (similar calculations were proposed in [26]). For example, the probability of up-regulation is computed as

$$p(\mathcal{M}_+^g) = \frac{1}{|S|} \sum_s p(\mathcal{M}_+^g | s) p(s). \quad (2.11)$$

A possible extension to continuous nuisance covariates is discussed in Supplement A.

Comparing subpopulations lying in different batches In some setups however, there may not be perfect overlap of each condition in all batches. (e.g., when comparing tumors, or inflamed tissues to healthy controls) Particularly, when comparing tumors, or inflamed tissues to healthy controls, batches may denote cells from individual donors, and one aim is to detect differentially expressed genes for a cluster of cells between subsets of batches. In such a case, we redefine the LFC in Equation 2.8 to make it independent of the batch assignments by sampling normalized gene expression in each cell using observed batches. In this more ambiguous scenario, we condition the posterior scales distributions w.r.t. observed batches, i.e., $p(h_g^C \mid s_n = s_n^{\text{obs}})$, s_n^{obs} being the batch that cell n originates from.

Controlling the false discovery rate

Bayesian decision theory [36] shows that the optimal decision rule for our DE task consists of selecting the top genes ranked by p^g (Equation 2.10). However, selecting the number of “top” genes is not straightforward. The use of Bayes factor to guide this choice carries the complication of not being interpretable when it comes to the expected error rates. A more intuitive frequentist solution is to set the significance threshold α over p^g in a way that controls for the false discovery rate (FDR).

To this end, let d_g be the (unknown) binary random variable denoting whether or not gene g is DE. Consider the multiple binary decision rule $\mu^k = (\mu_g^k, g \in G)$ that consists in calling DE the k genes with the highest differential expression probability p^g . The false discovery proportion (FDP) of the decision rule μ^k can be expressed as

$$\text{FDP} := \frac{\sum_g (1 - d_g) \mu_g^k}{\sum_g \mu_g^k}. \quad (2.12)$$

We use the formulation of the posterior expected FDP, as introduced in [84] and applied to genomics data in [85, 86], corresponding to

$$\overline{\text{FDP}} := \frac{\sum_g (1 - p^g) \mu_g^k}{\sum_g \mu_g^k}. \quad (2.13)$$

Because this quantity is tractable after inference, a natural choice for k corresponds to the maximum value for which the posterior expected FDP is below the desired level α .

2.4 Results

lvm-DE addresses two key tasks in differential expression analysis. The first corresponds to the robust estimation of the LFC in noisy and sparse data, including for rarely observed cell states.. The second task is evaluating the statistical significance of the observed effect size, and calling genes as differentially expressed for any target false discovery rate. For both tasks, we apply the lvm-DE framework to scVI and scPhere. We denote scVI-lvm and scPhere-lvm the associated combinations, in contrast to scVI that natively provides a test for differential expression. We evaluate our models in comparison to the current state-of-the-art, including MAST [55], DESeq2 [33], edgeR [32], and Limma-voom [64] (referred to as voom).

For each of these models, we follow the guidelines and implementation details specified in a recent comparative study of DE methods for scRNA-seq [63] (further details in Appendix F). We assess lvm-DE performance for LFC estimation, and for the prediction of correct and calibrated DE gene sets using both synthetic and real datasets. In the real datasets, we focused the analysis on the top 3,000 highly variable genes using Seurat. Annotations were taken from the respective publications.

Accurate and calibrated results on simulated scRNA-seq data

As a simulation scheme, we employ SymSim [87], which models biological and technical noise factors (e.g., transcriptional bursting and PCR amplification), as well as statistical dependencies between genes (relying on low-dimensional representation of the kinetics of promoters in each cell). We used SymSim to generate a hierarchy of cell states and focus our analysis on the comparison of two “sibling” types, denoted A and B (Figure 2.2A). We generated a dataset of 10,000 cells and 1,000 genes across five cell types. The data corresponds to two batches (5,000 cells each) with unique molecular identifiers (UMIs), which have similar properties to 10x Chromium data. To estimate the ground-truth LFC, we use the expected gene expression means, derived from the expected promoter-kinetic parameters in each gene and cell type [87]. Ground-truth gene expression varies across clusters for all genes in SymSim, i.e., there are no true negative genes. We restricted the analysis to genes that either had large (≥ 1 in absolute value) or very small (≤ 0.2 in absolute value) LFCs between the two compared clusters, to define “true” DE genes without ambiguity. The resulting dataset contains a total number of 396 genes, including 127 differentially expressed genes. We selected these genes such that DE genes have non-ambiguous high expression differences (absolute LFC higher than 1), while negatives have negligible expression differences (absolute LFC lower than 0.2).

We first evaluated LFC estimates from the models. The lvm-DE-based models offered accurate estimates (Figure 2.2B), on par with edgeR and DESeq2. This result suggests that lvm-DE properly aggregates individual cells to infer expression levels in the two subpopulations. An outlier in this analysis was MAST that exhibited a high mean-squared error range. This likely comes from the fact that this method relies on a hurdle model, whose LFC is difficult to compare to the ground-truth LFC. It was therefore discarded from further LFC estimation experiments.

By definition, scVI-lvm and scPhere-lvm make use of all cells in the input data, including “external” cells that are not in the groups being compared. In our simulation these are simulated cells that are not in groups A or B (Figure 2.2A). To test whether the inclusion of more external cells affect accuracy, we varied the number of external cells and conducted differential expression analysis between groups A and B (which remain unchanged). We find that our FDR control procedure is well calibrated, and remains at the expected level in all cases. Furthermore, we find a steady increase in the true positive rate (Figure 2.2C). These results indicate that increase in the overall dataset sizes, but not necessarily at the compared populations, leads to increase in power, thus supporting the notion the DGMs can benefit from information on all cell types (e.g., gene-gene correlations) in a dataset. This is in contrast with the other benchmark methods listed here which do not absorb any information from cells outside the compared groups.

The results in Figure 2.2D support the suitability of lvm-DE for characterizing rare cell types, as capturing gene couplings can help alleviate the issue of small sample size. We tested this by leaving in the input data only a few cells of group *B* (representing a rare cell type), while keeping group A as before (representing a more prevalent background; see Figure 2.2A). In particular, lvm-DE was retrained each time the rare subpopulation size was modified, such that all algorithms were trained on the same input data. For a rare subpopulation of size 50 and up, both lvm-DE methods provide gene lists that are accurate and control the FDR for several significance levels. In this simple experiment, we considered first a simple baseline, relying on a two-side t-test on log-counts-per-millions (CPM) counts, which fails at controlling the FDR. For the GLMs, we observed acceptable FDR control in small sample size setups, but interestingly their FDR becomes higher than its expectation, with larger sizes of the rare subpopulation. An exception is the default DESeq2 procedure, which uses a composite null hypothesis (i.e., effect size threshold $\delta > 0$). This method does not suffer from large FDR, but loses power because of the difficulty to properly set the effect size threshold in practice. Finally, we observe that the Bayes factor approach originally used in scVI yields high FDR levels. Consequently, we excluded this procedure from the next experiments.

To further our confidence in these results, we expanded the analysis in two ways.

In the first analysis, we modified our simulation process to generate additional, and possibly more nuanced scenarios of differential expression, such as bimodality of the ground truth gene expression [88]. Our procedure compares favorably to the other algorithms as it provides a high number of true positive genes, and few false positives (Figure S2). In a second analysis, we used MUSCAT [89] as our simulation framework. Unlike SymSim, the data generated by MUSCAT is based on independent sampling from an underlying (here, negative binomial) distribution. Therefore, in this case, there is no benefit in using latent variable models, as no couplings between genes (which is to be expected in real datasets) can be leveraged. In such setups, the lvm-DE methods provided calibrated predictions, while having similar or lower level of power compared to the GLMs, as expected (Figure S3).

Favorable comparison to current methods on single-batch, medium-sized PBMC data

To test the performance of lvm-DE with data, we used a 10X Chromium dataset of 9,432 peripheral blood mononuclear cells (PBMC) from a human donor [90] (Figure 2.3A). This data provides an opportunity to establish the relative performance of lvm-DE in a configuration in which batch effects are not present.

While there is no ground truth for differential expression in this data, several strategies can help evaluate the estimation of differential expression. First, we can ensure that lvm-DE does not detect differences in expression when comparing groups of cells of the same state. We therefore construct negative controls (Figure 2.3B) by randomly splitting the B cell population (460 cells) into two groups (230 cells each) and predicting the LFCs with the different algorithms. In this configuration, we expect to measure negligible expression differences. lvm-DE LFC estimation compares favorably to its competitors, since scVI-lvm along with scPhere-lvm predict LFC of significantly lower amplitude than their GLMs counterparts.

Another benchmark strategy is to compare the outcome of the DE analysis to results obtained from an independent dataset. To this end, we used estimations of LFCs and p-values from a bulk-RNA-seq dataset [91], in which DESeq2 was used for pairwise comparisons of monocytes, B cells, plasmacytoid, and myeloid dendritic cells.

The bulk-level LFCs and significance scores were then compared against the results of applying the different algorithms on the single cell data (Figures 2.3C and 2.3D). We find that scVI-lvm and scPhere-lvm compare favorably to the GLMs in recapitulating both the LFC and the significance scores of the reference bulk data. Notably, the cell groups we analyzed had varying sizes (Figure 2.3A), demonstrating

that lvm-DE can capture state differences for abundant as well as less common cell types. A detailed comparison of LFC estimations for every method and for every pair of cell types is depicted in Figure S4.

lvm-DE pools information from different sequencing technologies for better differential expression

lvm-DE can be employed to perform DE in large-scale multi-batched data. As an illustration, we consider the Pbmcbench data collection [92], consisting of two biological samples sequenced with seven different protocols, either droplet (drop-seq, 10Xv2, 10Xv3, inDrop and CEL-seq) or well-based (seq-well and Smart-Seq2). This data provides the opportunity to further validate the ability of our procedure to alleviate technical factors' impact on DE.

We compare the estimated LFC of scVI-lvm with bulk-RNA using the same reference as in the previous experiment. The inclusion of additional data sequenced with different technologies consistently increase the correlation between lvm-DE effect-size and significance predictions with bulk-RNA data (Figure 2.4). In contrast, observing more replicates does not systematically improve the match of GLM's predictions with bulk-RNA. This suggests that contrary to GLMs, lvm-DE succeeds in pooling information from heterogeneous experiments to better capture differences between cell states.

In summary, lvm-DE benefits when either the number of observed cells increases or when more samples are available, even if the samples differ technically.

lvm-DE better recapitulates inflammatory gene expression patterns in COVID-19 patient data

We now focus on a larger PBMC dataset, consisting of 44,721 cells from seven healthy donors and six patients with confirmed SARS-CoV2 infection [93] (Blish). The latent variable models were trained using donor ids as batches. This dataset provides an opportunity to evaluate procedures for differential expression, not only at comparing different cell types (to find their respective markers), but also comparing donor groups *within* each cell type. The latter comparison, revealing the influence of inflammation on different immune compartments, is usually more challenging with a more nuanced and less reproducible signal [94] (Figure 2.5A). In this new scenario, a popular approach consists in *pseudobulking* the data, i.e., aggregating counts by sample prior to testing. We include this additional baseline for this experiment, relying on a two-sided t-test on log-CPM pseudobulk.

Starting with between-cluster analysis, we divided the cells into types using annotations from [93]. We started by applying the algorithms on a negative control setting, obtained by evenly splitting dendritic cells as two clusters (203 cells each) and comparing the two clusters (Figure 2.5B). As in the previous experiment, the LFC values estimated by scVI-lvm and scPhere-lvm in this negative control are significantly smaller than the rest of the algorithms. We next proceeded to a comparison between different cell types (irrespective of disease status). Consistent with our results above we find that the differential expression analysis scVI-lvm and scPhere-lvm more closely match the bulk reference than the GLM-based methods, in terms of both effect sizes (LFC) and significance (ordering by FDR or p-values) estimates (Figure 2.5C and Figure S5 for the individual plots). We also note that the lvm based methods scale better in terms of computation time for this large dataset (Appendix K).

We next moved to the more challenging task of differential expression between the patient and control population within each cell type. In this case, the lvm-DE approach uses the modified LFC to compare these populations, given that case and control subpopulations correspond to disjoint batches. SARS-CoV-2 infection has been demonstrated to lead to aberrant secretion of pro-inflammatory cytokines such as interferon gamma [95]. As a reference to evaluate our DE analyses, we therefore used bulk transcriptome measurement of responses to different cytokines by different immune cell types. We first used a Microarray dataset [96], which provides reference LFC for cell-type-specific effects of interferon alpha. We find that scVI-lvm and scPhere-lvm better correlate with the reference microarray (significant under a t-test at significance level $\alpha = 0.05$; Figure 2.5D and Figure S6 for the individual plots). We repeated this analysis with another reference dataset of bulk RNA-sequencing measurement of responses to other cytokines (IL2, IL4, IL7, IL9, IL15, and IL21), and again find that the LFC and significance evaluations by lvm-DE had consistently high correlations with the reference (Figure S7). In summary, we find that also in this more challenging task of within-cell type comparisons, lvm-DE provides more accurate estimations of gene expression compared to our benchmark methods.

2.5 Discussion

While much effort has been made recently to design deep generative models for scRNA-seq and more generally multi-omics data, there has been a paucity of research on leveraging these models for DE. In this work, we present lvm-DE, a novel empirical Bayes method for differential expression, taking full advantage of the flexibility and scalability of latent variable models. This flexibility permits nonlinear batch effects to be accommodated in the model, improving on global linear models for batch effect

correction that assume that each batch has the same cell-type proportions, often an invalid assumption in scRNA-seq experiments [20]. Furthermore, most DE pipelines, with few exceptions [97], do not take advantage of all available samples for between-type comparisons. Latent variable models that lvm-DE rely on can harness the learned shared gene-gene correlations refine gene expression modelling and hence, increase the number of detected DE genes. As such, lvm-DE helps characterize gene expression differences between cell populations in complex large-scale, batch-confounded datasets, and provides calibrated DE gene sets relying on posterior expectations of the FDP.

While we demonstrated broad applicability of lvm-DE, we discuss now two caveats that should be considered while implementing or using lvm-DE with a deep generative model. First, it is possible that lvm-DE may require slight modification of existing latent variable models, in order to ensure reasonable performance. In particular, we observed empirically that the library size modeling was key, e.g., in DE comparisons involving two populations with different RNA content. In scVI, the library size is a latent variable, while in ScPhere it is deterministically assigned to the number of transcripts. We noticed that latter scheme does not always successfully normalize counts, and outlined what specific elements were required to unlock differential expression for scPhere (Supplement C). Second, lvm-DE may not be suitable to analyze very small datasets, containing less than a few thousand cells, as this setup may not be suitable to properly fit deep generative models. In such cases, it is more natural to use traditional methods, that (i) scale well and (ii) are not prone to overfitting.

As ground-truth cell types labels are not available in most scRNA experiments, the success of DE heavily relies on a careful annotation strategy. Practical realities, including limited data availability or mislabeling, may deteriorate cell-type prediction accuracy and desired resolution, hence the range of DE analyses. The emergence of high-quality, expert-annotated cell atlases [98, 99], along with scalable transfer-learning procedures [100], promises to provide straightforward and steady ways to annotate cells. As DE pipelines often follow the canonical approach of using the same gene counts to annotate (i.e., formulate hypotheses) and compare cell populations (testing these hypotheses), a core challenge is that the detected DE genes may reflect spurious differences, initially captured during the annotation process [101]. Fortunately, emerging approaches may limit the circularity that is central to DE for scRNA [102]. We evaluated how such a procedure, molecular cross-validation, could reduce spuriousness using synthetic data, showing marginal improvement over the canonical approach (Figure S8).

As lvm-DE outlines a generic procedure to conduct DE for latent variable models, improving the LVM of choice can be a direction to improve the quality of the predictions. One idea could be to leverage available information on gene localization to improve the underlying representation of gene-gene correlations. For instance,

hmmSeq [86] is a hierarchical Bayesian model for RNA-seq taking gene positioning into account to better capture co-expression of genes in the same chromosome. Designing models that handle the sparsity of scRNA-seq experiments is another way to improve DE for latent variable models. From a biological perspective, understanding the nature of zeros and the necessity to model their excess is an active topic of research [55, 74, 75]. This question depends on the technology used for sequencing and has already been illustrated through the analysis of zero inflation for scRNA in latent variable models [26]. Besides, our aim is to formulate a general framework for differential expression for such models, that is agnostic to such modeling choices. Still, we found that lvm-DE manages to provide meaningful and calibrated decisions for zero-inflated models (Supplement J, Figure S9).

Designing better optimization procedures for latent variable models in sparse, high-dimensional setups is another objective for scRNA-seq data. Sparsity can indeed hinder lvms' ability to handle larger number of genes. In scRNA-seq data, this limitation could be overcome in several ways for models resorting on amortized variational inference. The use of more refined training procedures, relying on tighter bounds to improve the fitness of the generative model [103] could unlock DE on larger datasets. The introduction of group-sparsity priors in the decoder could effectively leverage the fact that many genes share generalized biological functions to speed up inference [104]. We hypothesize that scaling up latent variables models to larger datasets might not only help detect more DE genes, but could increase the resolution of cell population studies.

While providing interpretability of decisions in terms of FDR was the main focus of our work, FDR control may not always be the most appropriate way to calibrate decisions for DE analyses. Predicting the direction of the LFC may be more important than detecting the presence of a DE in some applications, in which case False Sign Rate control [105] provides an alternative that should be considered.

Another extension of our work is the formulation of differential analyses for latent variable models that integrate several omic modalities. scMET [83] is a Bayesian model for differential methylation. Leveraging existing correlations [106] between methylation and gene transcription in a unified latent variable model and formulating meaningful hypotheses could be a step forward to better understand complex differentiation scenarios, e.g., for embryonic stem cells characterization [107].

Code availability

The implementation to reproduce the experiments of this paper is available at <https://github.com/PierreBoyeau/lvm-DE-reproducibility>. The reference implementation of scVI is available at <https://github.com/scverse/scvi-tools>.

2.6 Supplementary Information

Correcting continuous nuisance variations

Procedure While lvm-DE assumes categorical batch covariates, it can easily be extended to continuous covariates. First, training the generative model with a continuous covariate is straightforward in most cases [53]. The remaining question is to understand how to account for the continuous covariate during differential expression. In the case where these covariates have similar distributions in both cell subpopulations, we generalize Equation 11 to

$$p(\mathcal{M}_+^g) = \int_s p(\mathcal{M}_+^g | s) p(s) ds, \quad (2.14)$$

where $p(s)$ denotes some prior for the batch. Several strategies can be employed to approximate the above quantity, e.g., using Monte Carlo sampling, or discretization. We employ the latter strategy. We note $r_1, \dots, r_K$ the subintervals of $\mathbb{R}$ associated with the K -quantiles of the empirical distribution of s . For a compared cell population $C \in \{A, B\}$ we stratify cells into K groups $C_1, \dots, C_K$, where $C_k := \{n \in C, s_n \in r_k\}$. We then approximate Equation 2.14 by the average of the posterior probabilities computed from the comparison of A_k to B_k , for $k \in \{1, \dots, K\}$, where the LFC is estimated by additionally sampling the technical covariate from the empirical truncated distribution of s to r_k .

Validation on synthetic data To illustrate the validity of this approach, we simulated a synthetic scRNA-seq dataset for two cell-types A and B affected by a continuous technical covariate, denoted as $s_n \sim \text{Uniform}(-1, 1)$ for cell n . The resulting dataset contains 1,000 cells (500 observations per cell-type), and 500 genes. Observed counts x_{ng} for gene g and cell n of type $c_n \in \{A, B\}$ are obtained as independent draws, such that

$$x_{ng} \stackrel{\text{indep}}{\sim} \text{Poisson}(h_{ng}) \quad \text{where} \quad \log h_{ng} = \log h_g^{c_n} + s_n \mathbf{I}(g \leq 100). \quad (2.15)$$

In particular, 100 genes are affected by the technical covariate, and the ground-truth expression levels $\log h_g^A, \log h_g^B$ are constructed such that half the genes are differentially expressed. We compare the performance of different batch correction strategies for the analysis of this data, and show that the proposed strategy controls the FDR on this synthetic dataset (Figure 2.6).

Implementation of the different LVMs

ScSphere was reimplemented in PyTorch using the same architecture than the original structure. However, to avoid the scalability and numerical instability of the Fisher Von-Mises distribution for consequent latent dimensions, this distribution is replaced by the Power Spherical, which does not suffer from those effects [108]. The main difference we introduced lies in the counts normalization procedure. scSphere uses observed library sizes $l_n = \sum_g x_{ng}$. To ensure that the h are properly normalized for DE applications, we introduce a latent variable l accounting for library-size differences to augment scSphere. Next, we also empirically observed that the cell-specific overdispersion used by scSphere sometime caused instabilities in LFC estimation (see Appendix 2.6), and for this reason opted for gene-specific overdispersion parameters. Originally, both scSphere and scVI use batch normalization for the encoder and decoder. After facing several stability issues in differential expression tasks with this normalization scheme, batch normalization was replaced by layer normalization. The two models originally use dropout as a regularization scheme. This choice implies an infinite ensemble of variational distributions is used during training. During evaluation on such networks, a common practice consists in disabling dropout by reweighting the regularized layers' parameters. It is unclear why the associated distribution should be used for importance sampling. Consequently, we discard dropout layers from the architecture of the models. The remaining of the architectures follow the different models' original implementations. We train each model with minibatches of size 1024 using the Adam optimizer to speed up convergence without computational toll. For better variational distribution coverage, we optimize the IWELBO with $K = 25$ particles, and resort to SNIS to estimate posterior expectations as described above. For these latent variable models, the optimization scheme, and in particular the learning rate is a vital hyperparameter. To avoid manual tunings of the learning rate in each experiment for each model, we rely on the automatic learning rate range test [109]. They provide data-driven, automatic learning rate choices for each latent variable model and experiment.

Ablation study

Table 2.1: lvm-DE features ablation study (Symsim dataset). The reference model corresponds to scVI-lvm, trained with the IWELBO with 25 particles, and layer normalization. From left to right: FDR Mean absolute error (MAE; lower is better) LFC MSE of non-DE genes (NC, lower is better)

	FDR MAE	LFC MSE
Ref.	0.0197	0.0998
Ref. without outlier removal	0.3185	0.7285
Ref. with $\epsilon = 0$ (Eq. 8)	0.0391	0.1201

First, we analyze the importance of each feature of our DE framework (Table 2.1). We observe that our heuristic to remove outliers prior to compute posterior z samples improves FDR as LFC estimation for null genes. Using nonnull values for ϵ in Equation 8 also reduces the risk to attribute high fold changes to null genes and improves FDR control. This approach can hence be viewed as a regularization scheme to ensure that the detected genes have practical interest, on top of reducing the risk of predicting False Positive genes. The automatic effect-size threshold and outlier removal procedures are key components to better approximate the False Discovery Rate. Using pseudo-counts slightly affects the gene rankings, but it also reduces the risk to attribute high fold changes to null genes. In the absence of this feature, the MSE of the LFC for non expressed genes nearly doubles. Pseudo-counts can hence be viewed as a regularization scheme to ensure that the detected genes have practical interest, on top of reducing the risk of predicting False Positive genes.

Next, we conduct ablation studies of the objective functions and network architectures of the different algorithms to have a clearer understanding of the hyperparameters’ influence (Tables 2.2 and 2.3). In particular, we see that the choice of overdispersion in scPhere does not modify the ability of the algorithm to properly control the FDR. However, cell-specific overdispersion seem to increase the LFC error.

Using pseudo-counts

Most generative models rely on softmax operations to ensure that the normalized means. In such cases, the underlying expression levels have a straightforward interpretation, as they quantify gene frequencies. The softmax can however create artifacts when the two conditions A and B do not share the same library size distributions,

Table 2.2: scVI ablation study. Differences in MAE with Table 2.1 are attributable to weight initialization randomness.

	FDR MAE	LFC MSE	PSIS
Ref.	0.0504	0.1880	0.5988
Ref. + ELBO	0.0243	0.1306	0.4406

Table 2.3: scPhere ablation study

	FDR MAE	LFC MSE	PSIS
Ref.	0.0572	0.1117	0.0898
Ref. + ELBO	0.3185	0.7282	-0.1042
Ref. - cst overdispersion	0.0300	0.2688	0.1307
Ref. - random library	0.1672	0.3045	0.1104

which often is the case when A and B are cell-types. As the obtained frequencies can be arbitrarily small, the LFC distribution can reach extreme values when the gene is not expressed in both conditions.

As a solution, the offset ϵ (Equation 8) can serve a threshold frequency of interest that mitigates numerical artifacts. It also filters out low levels of expression, further ensuring that tagged DE genes have a practical interest. It is difficult to set the offset manually. A meaningful value depends on various parameters, including the sequencing depths of A and B , as well as the sparsity degree and shape of the data. We automatically tune this parameter, by looking at the distributions of gene expression levels for unexpressed genes in $C \in \{A, B\}$. Let $\mathcal{G}_C^0$ be the set of genes with zero counts in C , and $\hat{h}_g^C$ the maximum expression levels in gene g . If $g \in \mathcal{G}_C^0$, we can expect $\hat{h}_g^C$ to correspond to artifacts coming from the softmax operation. Yet, it is possible that some genes were expressed, but were not observed in the counts due to the low sensitivity inherent to scRNA sequencing. To ensure the robustness of the procedure to such events, we estimate the threshold offset as the empirical α lower quantile of $\hat{h}_g^C, g \in \mathcal{G}_C^0$, denoted as ϵ_C^α . In the experiments, we use $\alpha = 0.90$. In addition to the pseudo-count procedure, we design a robust LFC estimator that will mitigate instability when both populations do not express the gene.

Automatic effect-size thresholding

It is not always straightforward how δ should be set. The order of magnitude of the critical effect-size value that will determine if a gene is called differentially expressed depends on the two compared populations. For this reason, we suggest an automatic procedure to automatically set the value δ . To do so, we compute the posterior LFC medians $v^g := \text{median}(r_{A,B}^g)$ for each gene. We assume that a significant amount of LFCs will concentrate around 0 (which correspond to the genes that are equally expressed), while DE genes will concentrate around other modes. Based on this assumption, we fit a three-components Gaussian Mixture Model (GMM) based on medians v^g . We denote $m^i, i \in \{1, 2, 3\}$ the three obtained modes. Our δ candidate corresponds to the mean of the largest modes (in absolute value), whose associated distributions should contain differentially expressed genes.

Methods detail

We provide a scaled count matrix as input (following [110]) to MAST and edgeR. DESeq2 uses the original count matrix, which is then subject to internal normalization, using the *poscounts* estimation, which seemed to improve the normalization quality. We included the batch information as covariates of the linear models included in the design matrices. Obtained p-values are corrected for multiple hypotheses testing using the Benjamini-Hochberg procedure [37] for evaluation of false discovery rate. Similar to our approach, DESeq2 formulates composite null hypotheses in which the LFC absolute value is below a certain threshold δ . We set the value of the δ parameter of DESeq2 to 0.5. To distinguish DESeq2’s predictions using composite from the point null hypotheses, we precise $\delta = 0$ in the latter scenario. For other frequentist methods, which did not provide composite null hypotheses options, a subsequent filtering step of DE gene sets consisted in discarding genes with predicted LFC below 0.5 in absolute value.

SymSim extension to differential analysis

SymSim provides counts of the form x_{ng} along with labels s_{ng} . We focus on the comparison of cell types A and B .

We now describe SymSim-DA, a slight modification of the employed SymSim dataset that induces subtle differences in expression. Indeed, gene expression within a cluster can be multimodal. Better understanding how differential analysis tools handle such specificity is vital. The procedure we use consists in modifying the data to introduce multi-modalities in count distributions, from the raw counts. Let $\mathcal{G}_d$ be

the set of differentially expressed genes between A and B . We randomly split this set in gene sets, that we denote $\mathcal{G}_{DE}$, $\mathcal{G}_{EE}$, $\mathcal{G}_{DP}$, $\mathcal{G}_{DM}$, and $\mathcal{G}_{DB}$. The first set coincides to differential expression, and hence does not require any modification of the original data. The other sets correspond to equal expression (EE), differences in proportions (DP), differences in mixture (DM), and differences in both (DB) (Figure 2.7A). Each set relies on a pool of held-out counts for the two cell-types to artificially introduce other modes.

Both scVI-lvm and scPhere-lvm compare favorably to the other algorithms (Figure 2.7B.).

Differential expression on MUSCAT

We proceed to the same analysis than the one conducted in SymSim for a synthetic dataset generated using MUSCAT (Figure 2.8). In this framework, an external dataset helps to fit sample-specific negative binomials (with shared, gene specific overdispersion parameters). These reference distributions are used to construct a synthetic dataset consisting of several subpopulations. A first simple scenario consists in only focusing on differences of expression. In this case, each subpopulation will follow negative binomial count distributions, with subpopulation-specific mean parameters. Here, the presence of other cell-types cannot be leveraged to improve differential expression performance between A and B (Figure 2.8C). While lvm-DE’s FDR control is acceptable, linear models also work well in this scenario (Figure 2.8D).

Individual scatter plots

We here show the complete inter cell-types analysis for the PBMC (Figure 2.9) and Blish (Figure 2.10) datasets, as well as the intra cell-types analysis (Figure 2.11).

Zero-inflation effects

We here discuss to the flexibility of the lvm-DE framework to changes in the noise model. In particular, we investigate if zero-inflated models still control the FDR in the lvm-DE framework (Figure 2.14). Originally designed to better account for the low sensitivity of droplet-based sequencing procedures, its pertinence to properly model scRNA has been challenged[75]. In this dataset, we find that a zero-inflated negative binomial likelihood does not seem to better capture differentially expressed genes than their negative binomial counterparts. However, zero-inflated models may still be used by our model.

Running times

Running times of the different methods are provided in Table 2.4. They correspond for the inter cell-type comparisons performed on the Blish datasets. All experiments were run on a remote server with one Tesla V100 GPU, 40 virtual processors and 120 GB of RAM. We observed that the GLMs scaled poorly with higher numbers of batches, with the exception of Voom, which however provided poor correlation with bulk-RNA. While both scVI and scVI-DE require additional time to fit their generative model on the entire dataset, the lvm framework also provides other functionalities (clustering, normalization, imputation tasks) often required for differential expression analysis that require the use of external packages for the other DE protocols. In addition, scVI-DE differential expression execution time is constant with respect to the number of cells analyzed for differential expression.

Table 2.4: Mean Running time on the Blish dataset for the comparison of the six pairs of cell types. Training times correspond to the required time to fit latent variable models on the whole dataset for 250 epochs.

Blish	DESeq2	MAST	edgeR	Voom	scVI-lvm	scPhere-lvm
Pre-training (<i>min</i>)	N/A	N/A	N/A	N/A	21.4	23.4
DE (<i>s</i>)	266.7	139.0	102.7	1105.3	35.1	41.4

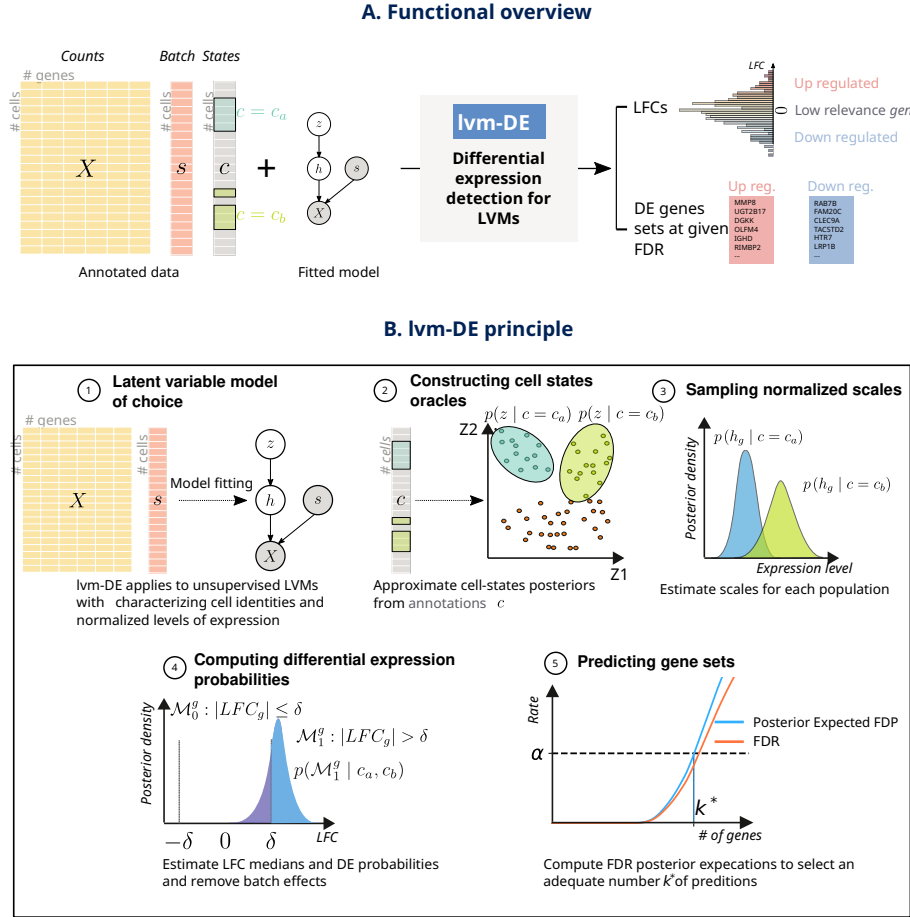


Figure 2.1: Differential expression model for deep generative models. **A.** lvm-DE takes annotated data (from clustering, metadata, or transfer learning), a latent variable model, and a target FDR level as inputs and returns log fold change (LFC) estimates as well as calibrated DE predictions. **B.** lvm-DE works as follows. 1. A preliminary step consists in fitting the latent variable model of choice of the data from the collection of available scRNA-seq data. 2. lvm-DE uses existing cell states annotations to approximate the distributions of c conditioned on the cell states. 3. These distributions help determine the normalized expression level distributions of the compared populations. 4. The associated LFC distribution helps to determine posterior DE probabilities that correspond to the model in which the LFC is higher than a given threshold. 5. To tag DE genes of interpretable interest, we estimate the maximum number of genes for which the posterior expected FDR is below the desired FDR level.

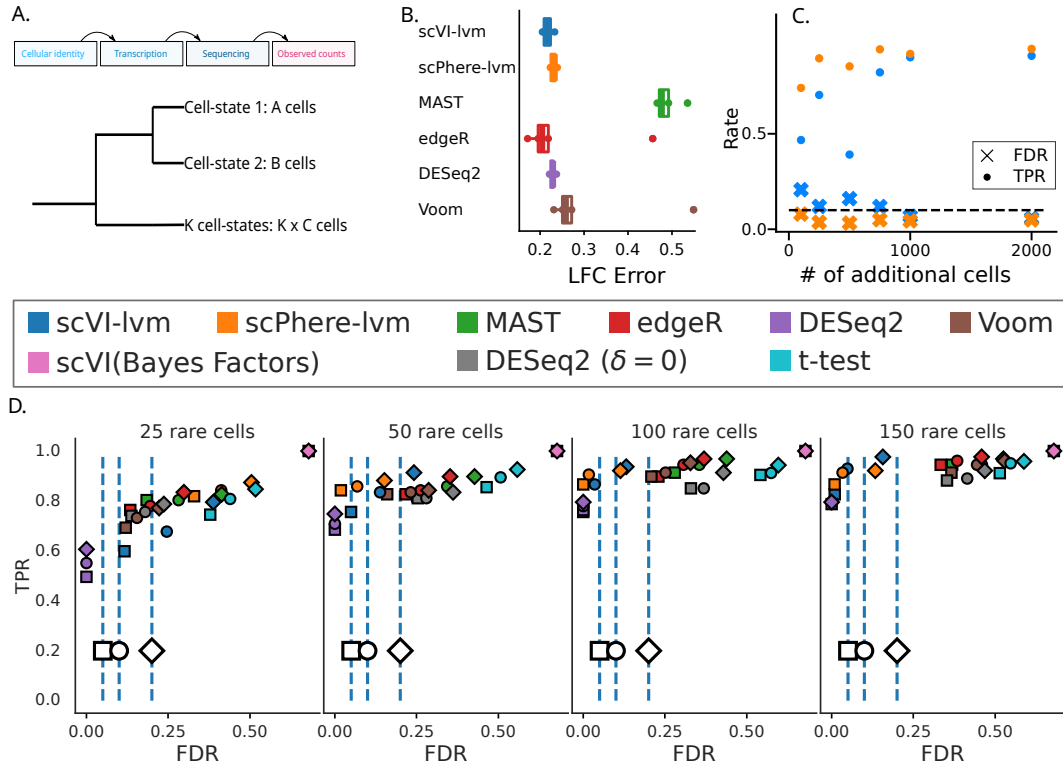


Figure 2.2: SymSim results. **A.** Dataset presentation. *Top*: SymSim is a simulation framework modeling biological and technical effects to provide realistic simulations. *Bottom*: We consider a two cell-type DE analysis scenario. We subsample population A to compare the different algorithms for rare cell-type detection. For this sub-panel and all the experiments, we refit the models for each scenario such that all of the algorithms use the same number of observations from A and B for model fitting and DE. **B.** LFC point estimation error when comparing two populations of $A = B = 200$ cells. For Bayesian techniques, we summarize the posterior LFC distribution by its median. For this figure and in the remainder of the article, boxplots represent medians (line), interquartile range (box), and distribution range (whiskers) estimates. **C.** TPR (dots) and FDR (crosses) changes for an increasing number of external cells for the different latent variable models. **D.** FDR and TPR of decisions for the detection of DE genes when comparing varying $A \in \{25, 50, 100, 150\}$ cells to $B = 500$ cells (for $C = 2000$) (We refit each model for each configuration, such that all algorithms use the same data). Squares, circles, and diamonds correspond to decisions controlling FDR at targets 0.05, 0.1, and 0.2. For scVI's original DE procedure, we reject the null when Bayes factors are greater than three in absolute value.

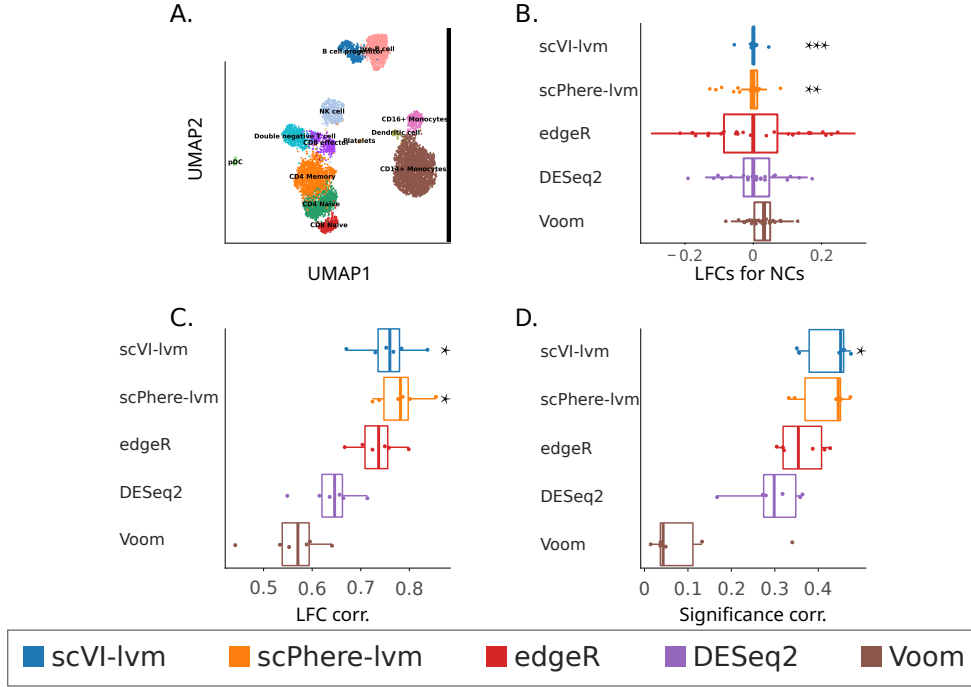


Figure 2.3: PBMCs results. **A.** UMAP from scVI's embedding **B.** Negative controls (among B cells), corresponding to LFC range study for the different methods. For this experiment, the lvm-DE outlier removal procedure was not employed. **C.** Positive controls. Distribution of Pearson correlation between the reference LFC (bulk-RNA) and estimated LFC for pairwise comparisons of B cells, mDC, pDC and monocytes. Each point in these graphs corresponds to one of the six possible cell-type comparison. For lvm-DE, we use the custom LFC median estimator. Individual scatter plots can be found in the annex. **D.** Distribution of Spearman correlations between the reference p-values (bulk-RNA) and estimated significance scores for pairwise comparisons of B cells, mDC, pDC and monocytes. GLMs and lvm-DE respectively used p-values and posterior DE probabilities as significance scores. Stars represent significant differences with all the GLMs at various significant levels (*, **, and *** denote respectively significance levels $< 0.05, 0.01, 0.005$), under a two-sample F test for the negative control and a one-sided two-sample t test for the positive control experiments.

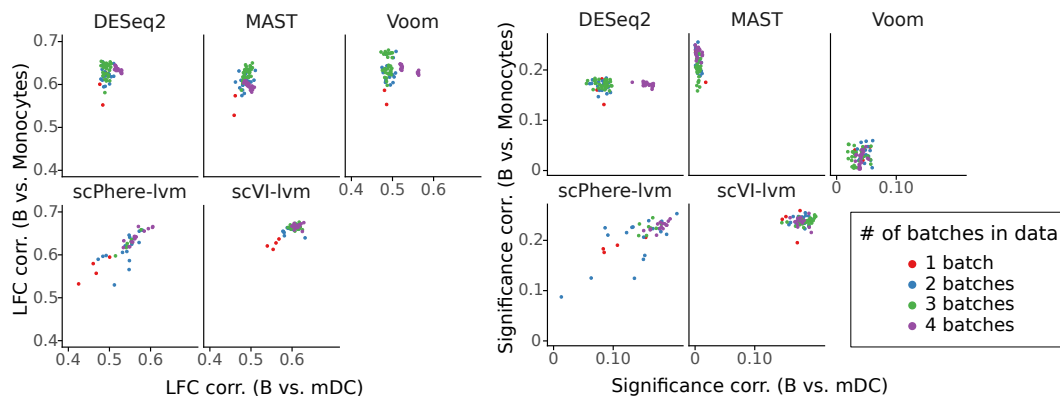


Figure 2.4: Batch harmonization on *Pbmcbench*. Pooled information from several batches improves the match of the prediction with bulk for scPhere and scVI-lvm. *Left*: Pearson correlation of the predicted and the reference LFCs (from bulk-RNA) for two cell-type pairs. *Right*: Spearman correlation of the predicted significance scores (posterior differential expression probabilities for scPhere and scVI-lvm, pvalues for other algorithms) and the reference pvalues (from bulk-RNA) for two cell-type pairs. In both graphs, points correspond to a given training on a subset of *Pbmcbench* containing a varying number of batches (color). As GLMs struggled to scale to large sample sizes, these algorithms used a maximum of 500 cells per dataset.

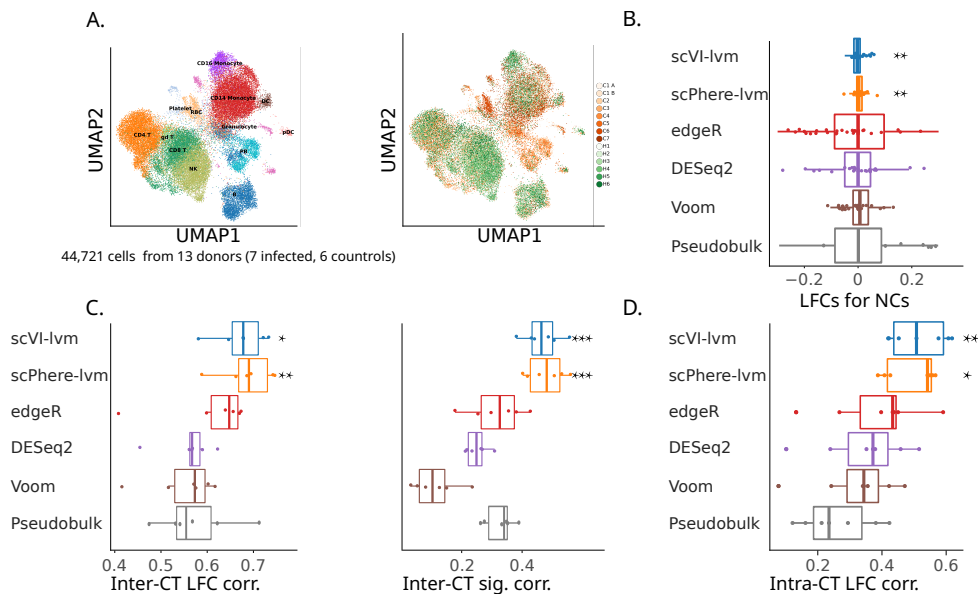


Figure 2.5: SARS-CoV2 dataset results. **A.** Dataset presentation. UMAP from scVI’s embeddings colored by cell type (*left*) and batch (*right*). Counts were obtained from six healthy donors (H1 to H6) and seven SARS-CoV-2 infected patients (C1 to C7). **B.** Negative controls (among DC cells), corresponding the study of the range of the LFC parameter (LFC) for the different methods. **C.** Positive controls for inter cell-type analysis. *Left*: Distribution of Pearson correlations between the reference and estimated LFC for pairwise comparisons of B cells, mDC, pDC and monocytes. *Right*: Distribution of Spearman correlations between the reference pvalues and estimated significance scores for pairwise comparisons. Each point in these graphs corresponds to one of the six possible cell-type comparison. **D.** Positive controls for within cell-type analysis. Distribution for different cell types of Pearson correlations between the reference and estimated LFC. The reference corresponds to cell-type-specific cytokine signature genes LFC independently computed on microarray, but that unfortunately did not contain significance assessments. The considered cell types are dendritic cells, NK, neutrophils, gd T, B, CD4T, and CD8T cells.

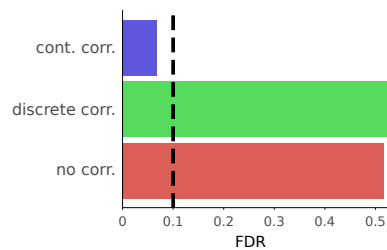


Figure 2.6: Continuous covariate synthetic experiment. Comparison of the FDR control (target level $\alpha = 0.1$) of scVI-lvm using *continuous*, *discrete*, or no covariate correction. The discrete covariate correction uses discretized covariates from empirical quintiles of the batch for both model fitting and DE analysis. The continuous covariate correction works on continuous batch values for model fitting, and uses the batch correction described in Supplement 2.6.

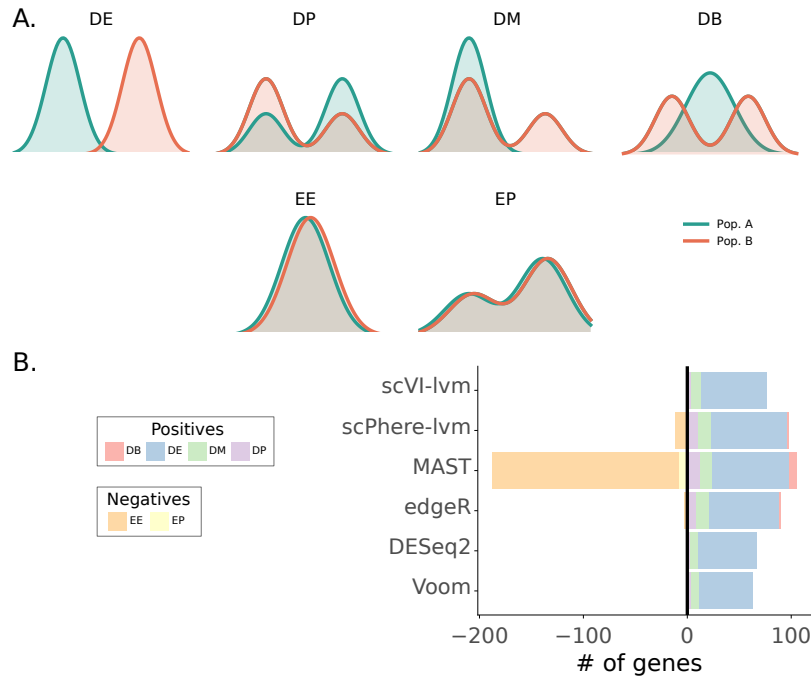


Figure 2.7: SymSim-DA results, extending the dataset to unlock a broader differential analysis scenario. **A.** top: Overview of the different differential distributions for differential analysis. From left to right: 1. Differential expression, in which the modes of gene expressions are different. 2. Differences in proportion, in which both populations express the gene according to two modes, but with different weights 3. Differences in mixture, in which the two populations share the same mode but one of them also has another mode 4. Differences in both, a combination of the last two scenarios. bottom: differential distributions that do not induce differences between the populations. In both cases, both populations have the same gene expression, but in EP the gene expression is multimodal. **B.** Types of genes detected by each method (at FDR level 0.05). Detections corresponding to DB, DE, DM, or DP correspond to true positives, while EE or EP genes denote false positives.

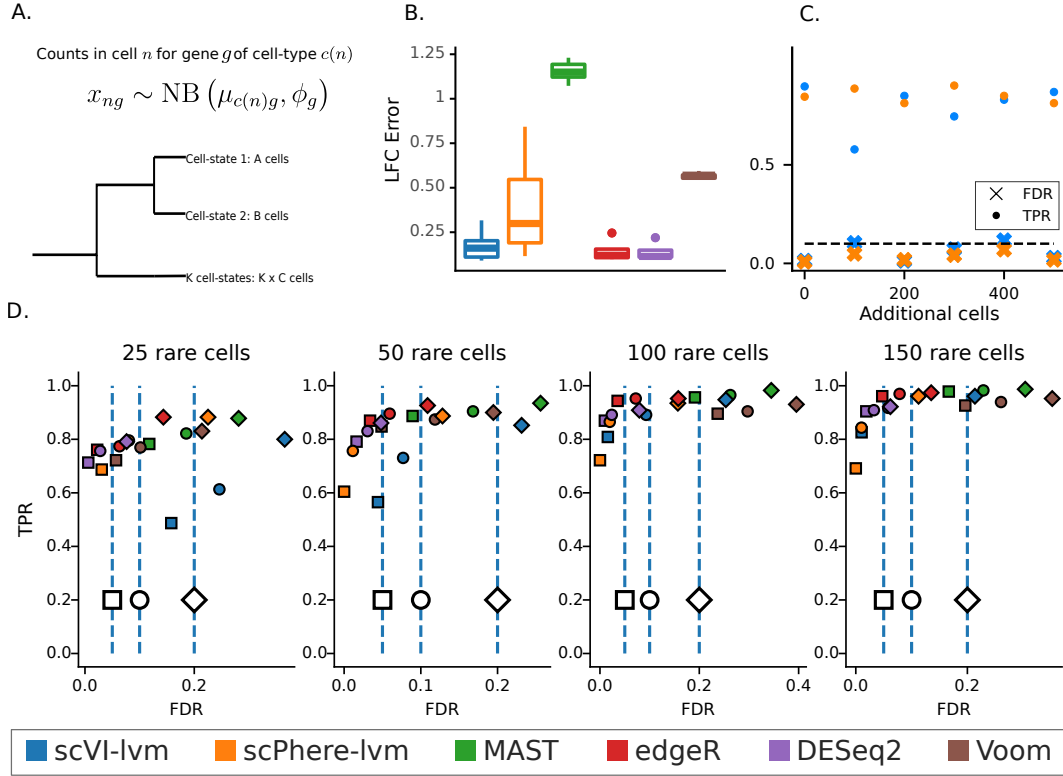


Figure 2.8: MUSCAT results. **A.** Dataset presentation. **B.** LFC point estimation error when comparing two populations of 200 cells. For Bayesian techniques we summarize the posterior LFC distribution by its median. **C.** TPR (dots) and FDR (crosses) changes for an increasing number of external cells for the different latent variable models. **D.** FDR and TPR of decisions for the detection of DE genes when comparing 25, 50, 100, and 150 cells to 500 cells. Squares, circles and losanges correspond to decision controlling FDR at targets 0.05, 0.1, and 0.2. Because the original scVI model does not provide explicit thresholds to tag DE genes, we reject the null when Bayes factors are greater than 3 in absolute value.

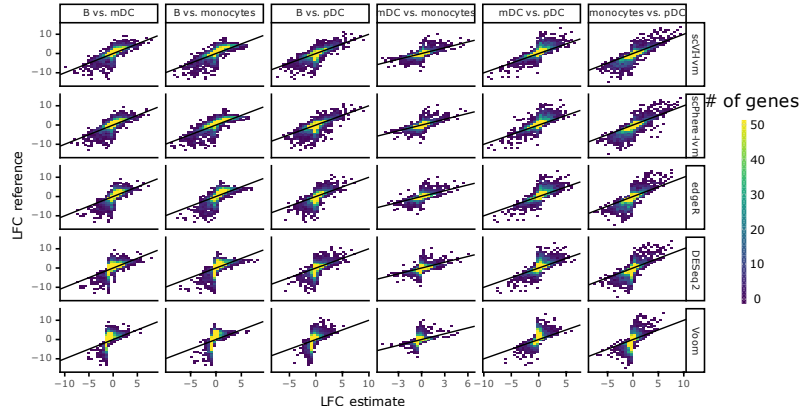


Figure 2.9: Inter cell-types analysis for the PBMC dataset, through the comparison of predicted LFC with reference bulk for pairwise comparisons of B cells, mDC, pDC and monocytes.

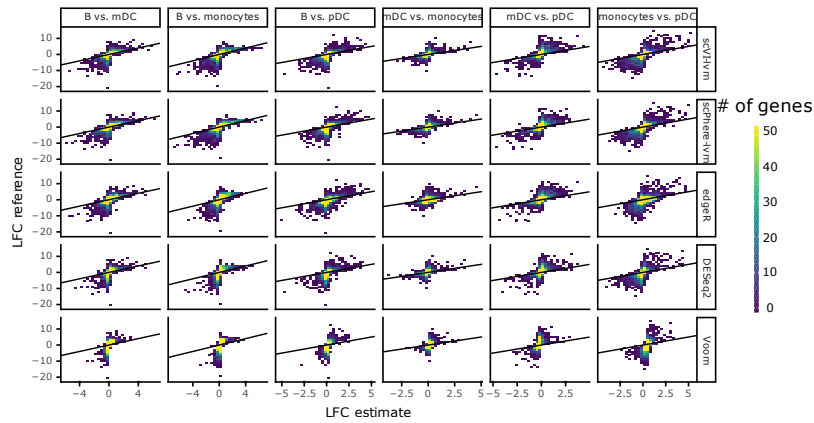


Figure 2.10: Inter cell-types analysis for the Blish dataset, through the comparison of predicted LFC with reference bulk for pairwise comparisons of B cells, mDC, pDC and monocytes.

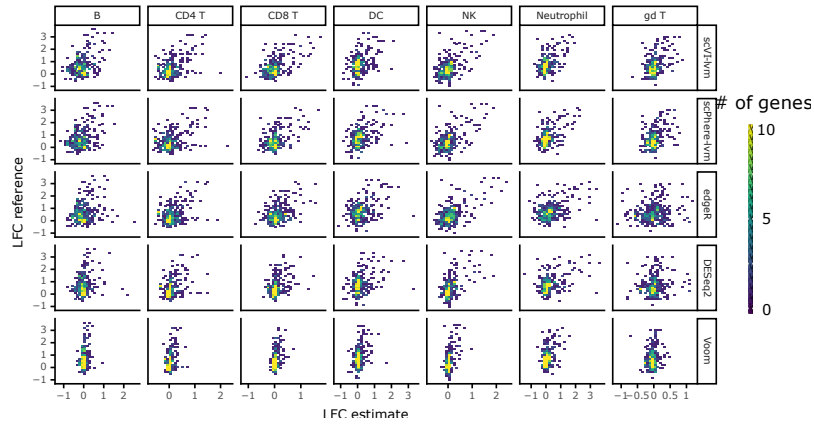


Figure 2.11: Within cell-types analysis for the Blish dataset, through the comparison of predicted LFC with cytokine genes microarray.

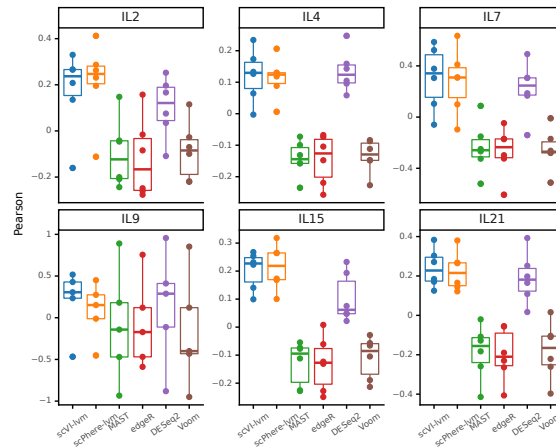


Figure 2.12: Predicted LFC correlations with reference RNA data for several cell-type-specific cytokine stimulation markers. In each subgraph, points correspond to the correlation for a given cell-type (B, Dendritic, NK, CD4, CD8, or gamma-deltas.)

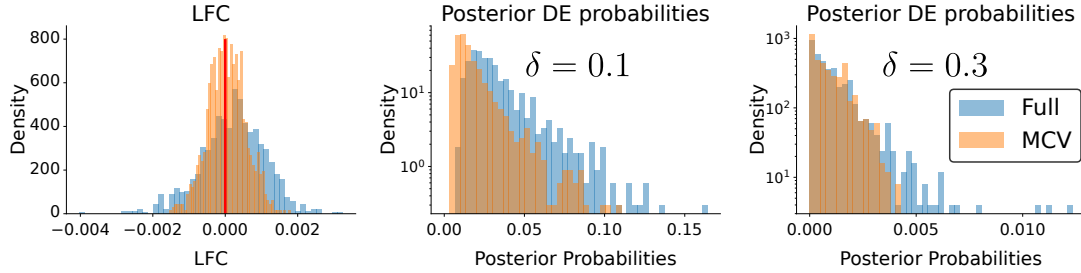


Figure 2.13: Effect of molecular cross-validation (MCV) on LFC estimation and posterior DE probabilities on Poisson negative-control data using scVI-lvm. The model either used the same transcripts to train the model and to detect DE genes (Full) or used different molecules for model fitting and inference (MCV), according to a 80%-20% split. *Left*: Distribution of obtained LFC estimates. *Middle* and *right*: Distribution of posterior DE probabilities for two LFC thresholds ($\delta = 0.1$ and $\delta = 0.3$).

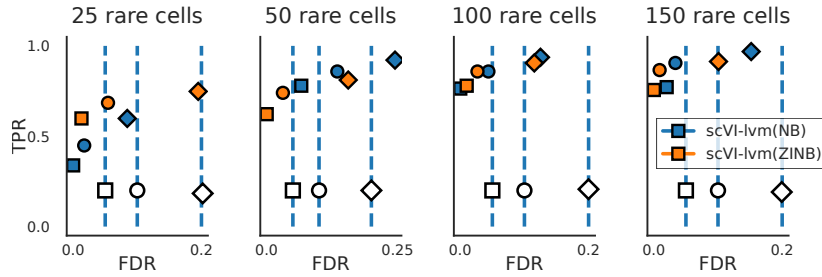


Figure 2.14: FDR-TPR curves on the Symsim dataset for varying A sizes, comparing different likelihood models for scVI-lvm. In this experiment, $B = 500$, $C = 2000$.

Chapter 3

Variable Inference for Variable Selection

3.1 Introduction

Single-cell transcriptomics offers an unprecedented opportunity for probing the function of individual cells and for characterizing the cellular composition of entire samples, thus shedding new light on processes in immunity, development, and pathogenesis of various diseases [12, 111–113]. The emergence of spatial and multiomic technologies further adds the ability to simultaneously profile the surface proteome, epigenome, or location of each cell, on top of its transcriptome. In addition to providing a more comprehensive view of each cell, these technologies open the way for a better understanding of the interplay between molecular or cellular properties. For instance, assessing the dependency between protein abundance on the cell surface and the expression of genes can help identify signaling cascades that help propagate extracellular cues and induce a transcriptional response [114]. Identifying associations between gene expression and the cell’s epigenetic landscape [111] may further help with our understanding of how gene expression is regulated. In spatial transcriptomics, an examination of gene expression patterns across tissue localizations may reveal how the microenvironment affects the function of its residing cells [115]. All of these opportunities require statistical procedures to help detect the most relevant relationships between the observed molecular or cellular features (genes, proteins, chromatin regions, cellularity of the microenvironment, and more).

In single-cell genomics and bulk settings, efforts to detect relationships between such features fall into two broad categories. The simplest methods identify *marginal* associations, which quantify statistical dependencies between pairs of features without

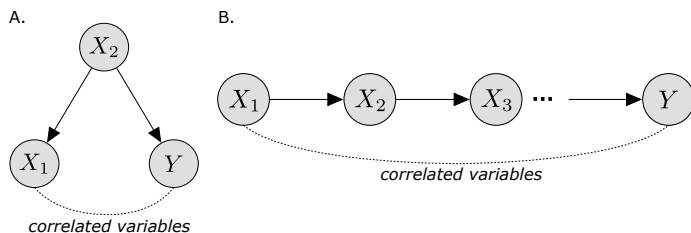


Figure 3.1: ***Correlated variables may be functionally unrelated.*** Here, X_1, X_2, Y are random variables characterizing the expression of three genes. **A.** X_2 is directly and causally linked to Y and X_1 . Here, X_1 and Y might be highly correlated, but X_1 does not causally affect Y . **B.** X_1 is directly and causally linked to X_2 , and similarly, X_2 is connected to Y . Here, X_1 and Y might be highly correlated, but their association is indirect.

considering the other observed features. While these were broadly used for studying gene co-expression networks [116–119], marginal associations suffer from key limitations for single-cell genomics. Practically any technology in this field is impacted by technical factors such as batch effects or variation in sequencing depth, as well as ‘nuisance’ biological factors that are less relevant to the question in hand, e.g., the cell cycle. These factors may inflate marginal correlations, resulting in associations that do not carry the intended biological meaning [120]. More fundamentally, a marginal correlation between two variables in any arbitrary system does not imply causation [121, 122]. For instance, two genes that are regulated by a common set of transcription factors can be highly correlated without being functionally related (Figure 3.1A). Even when they are functionally related, marginal dependencies may not inform on the proximity of this relationship when two highly correlated genes are indirectly linked through a series of mediator genes (Figure 3.1B). Marginal approaches hence tend to detect many spurious or indirect associations, which requires further filtering to identify the most relevant relationships [116, 123, 124].

Conditional associations are a second category of relationships that address these issues by accounting for the overall dependency structure of the data when assessing the dependency between a pair of variables [125–129]. Specifically, detecting conditional dependencies between a response variable Y and individual features in a feature matrix X often starts by learning a predictor function, $f(X) \approx Y$, which is then scrutinized to identify variables in X that are most associated with Y . The simplest example for this approach is the generalized linear model [130–132], in which learned regression coefficients are used to quantify conditional associations. While limited to linear relationships and simple noise models, linear approaches are relatively

scalable. In some cases, these models come with statistical guarantees for the inferred coefficients and are thus easily interpretable. Nonlinear predictors [125, 126] have also been introduced to capture more complex relationships, with tree ensembles being the most prevalent approach. Ensemble approaches have been demonstrated to reach state-of-the-art performance in a variety of tasks such as inference of regulatory interactions between genes [133]. Conditional dependencies provide a more stringent notion of association than marginal dependencies and are more likely to reflect causal relationships. Indeed, pairwise dependencies that persist after conditioning on all other variables imply causal relationships in cases where the causal direction is known, there are no unobserved causal variables, and there is no feedback loops [134]. As such, conditional dependencies are a promising avenue for uncovering relevant interactions in single-cell multiomic data.

In practice, however, algorithms for identifying conditional relationships often need to compromise on (i) *scalability*, e.g., requiring heavy pre-processing to ensure that inference can be completed in a reasonable timeframe [133], (ii) *modeling assumptions*, using often mis-specified view of the underlying process, e.g., with simplified noise models or by assuming linear relationships between variables [135], and importantly: (iii) *interpretability and calibration*, by relying on heuristics to evaluate which of the interactions under consideration are indeed relevant [136–138]. Given these challenges, analyzing dependencies in single multiomics, where millions of measurements (possibly from different batches or studies) are available, requires the use of scalable and rigorous statistical methods. These methods should be able to handle count data distributions, account for technical and biological noise and bias, and allow for nonlinear relationships between variables.

To address these three challenges, we introduce **VI-VS** (Variational Inference for Variable Selection), a general framework for conditional independence testing with multiomic data. **VI-VS** is based on the conditional randomization test (CRT) [39], which quantifies the credibility of pairwise interactions by measuring the effect of exchanging observed features with synthetic ones. We demonstrate and theoretically prove that our procedure provides a calibrated estimation of the false discovery rate. This is achieved without making any assumptions about the distribution of the response variable Y or the nature of its interactions with the features in X , such as linearity. **VI-VS** harnesses the distributional expressivity of latent variable models, allowing for a variety of noise models for X , including count distributions commonly used in single-cell genomics. Finally, **VI-VS** relies on deep neural networks for testing, allowing it to scale to large single-cell genomic datasets as well as capture complex nonlinear relationships between variables.

In the following, we demonstrate the accuracy and calibration of **VI-VS** with several simulation and multi-ome case studies. We also showcase that our procedure provides

a theoretically grounded “wrapper” framework that can take existing algorithms for detecting pairwise relationships, and use them to output calibrated decisions. We demonstrate this using the popular GENIE-3 algorithm for inference of regulatory networks.

3.2 Methodology and Statistical Framework

Overview of the problem

Before describing our approach in detail, we first provide an overview of the problem we address, and the main challenges we aim to address.

VI-VS is a conditional independence testing framework for single cell genomics. We observe, for multiple cells, G features $X_1, \dots, X_G$, e.g., genes, as well as a response variable Y , e.g., measured protein expression levels, or another type of cellular property. This section omits the confounding factor adjustment for simplicity. We refer to the Methods for a detailed description of the full algorithm in the presence of confounding factors.

Conditional independence tests identify features for which the null hypothesis of conditional independence $\mathcal{H}_{0,g} : X_g \perp\!\!\!\perp Y \mid X_{-g}$ can be rejected. Here, X_{-g} denotes the set of features $X_1, \dots, X_G$ excluding X_g . Conditionally dependent features, for which the null can be rejected, are associated with the response variable in a way that cannot be explained by the other features. This allows for the identification of features that have a distinct and significant association with the response variable, and that are less likely to be the consequence of spurious correlations among features.

VI-VS compares observed data statistics with statistics of synthetic data generated under the null hypothesis of conditional independence (Figure 3.2A). It requires two core components: (i) a way to generate synthetic data, and (ii) a procedure to compare observed with synthetic data. To generate synthetic data, VI-VS employs a *generative model*, more particularly a latent variable model, trained on a subset of the available data. The generative model can be seen as a simulator that generates synthetic data in the scenario where $\mathcal{H}_{0,g}$ holds. More specifically, the generative model produces K synthetic measurements for feature g , $\tilde{X}_g^{(1)}, \dots, \tilde{X}_g^{(K)}$ that are consistent with $\mathcal{H}_{0,g}$. In other words, no matter if the observed X_g is conditionally dependent on Y , the synthetic samples $\tilde{X}_g^{(k)}$ will be conditionally independent on Y by construction.

The next step is to compare the observed and synthetic data to assess the credibility of $\mathcal{H}_{0,g}$ via *importance scores*. Importance scores are data statistics that concisely summarize the relationship between the features and the response variable and that we can use to compare the observed and synthetic data. A simple example of an

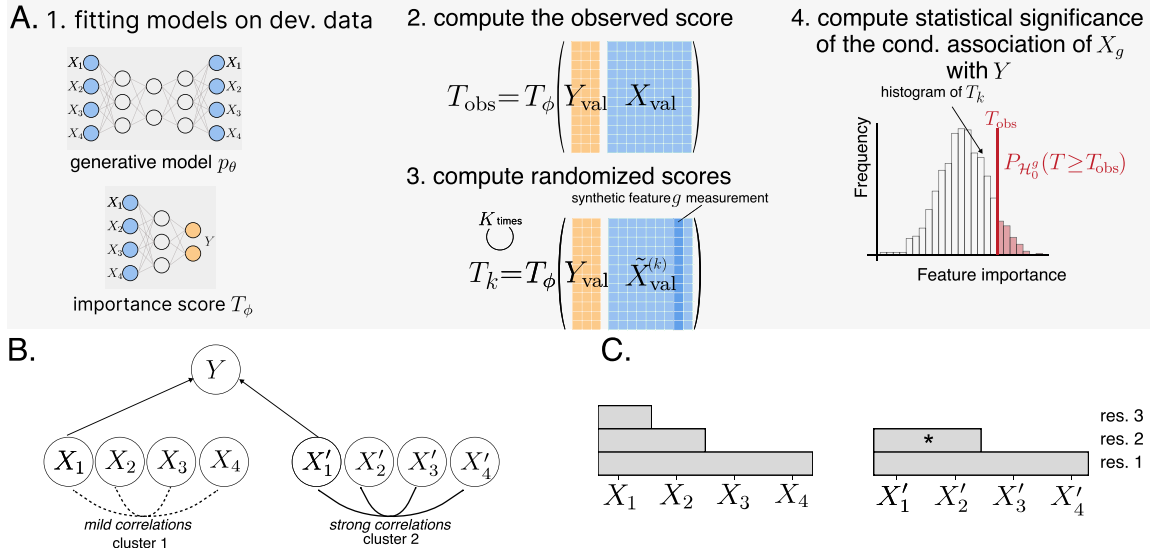


Figure 3.2: **A. VI-VS overview.** VI-VS identifies conditional dependencies between molecular features X and a response variable Y in single-cell genomics. **1.** We split data into validation and development sets, fitting a generative model p_θ and importance score T_ϕ on the development set. **2.** We compute the importance score on the validation set. **3.** We sample K synthetic feature samples for gene g using the generative model, computing synthetic importance scores based on Y and the modified feature matrix X . **4.** We compare observed and synthetic importance scores to compute a p -value. **B. Power limitation of conditional approaches.** Features X_1, X_2, X_3, X_4 are mildly correlated, and form a first cluster. Features X'_1, X'_2, X'_3, X'_4 are strongly correlated and form a second cluster. If the target response causally depends on X_1 and X'_1 , then a conditional independence test may fail to detect X'_1 due to its strong correlations with features of the same cluster. **C. Illustrative example of multi-resolution testing on B** in the case where conditional dependencies at assessed at three resolutions (res. 3 being the finest at the feature level). VI-VS can detect groups of features that are conditionally dependent on the response variable, even if no individual gene can be identified as conditionally dependent, as well as individual features, if the sample size allows. For instance, VI-VS could detect, in addition to individual feature X_1 , that group $\{X'_1, X'_2\}$ (marked as a star in the figure) is conditionally dependent on the response without being able which of the two features is responsible for the association.

importance score is the prediction error of a regression model of Y given X , e.g., a

neural network or a linear model trained on a subset of the observed data. If $\mathcal{H}_{0,g}$ were true, then the value of feature g should not be informative for predicting the response. In particular, replacing X_g with a synthetic counterpart $\tilde{X}_g^{(k)}$ in the input of the regression model should not significantly change the prediction error. On the other hand, if X_g is conditionally dependent on Y , then X_g should be informative for prediction, and the replacement operation should lead to a significant increase in the prediction error.

Following this logic, and in the general setting, we let $T(X, Y)$ denote the importance computed on the observed data, and $T(\tilde{X}^{(k)}, Y)$ as the one computed from inputs $\tilde{X}^{(k)}$, where X_g was replaced $\tilde{X}_g^{(k)}$. VI-VS compares the observed importance score to the histogram of the synthetic importance scores to compute a p-value for $\mathcal{H}_{0,g}$ as

$$p_g = \frac{1}{K+1} \left(1 + \sum_{k=1}^K \mathbb{I} \left(T(\tilde{X}^{(k)}, Y) \leq T(X, Y) \right) \right).$$

Theoretical guarantees The p-values computed by VI-VS are calibrated; in particular, they control the false discovery rate (FDR) at the desired level, regardless of the complexity of the relationship between the response variable and the features. The core assumption of VI-VS is that the generative model can describe the statistical relationship between features $X_1, \dots, X_g$. While not described in this section, VI-VS adjusts for confounding factors, e.g., batch effects, that may affect the relationship between X and Y . We refer to the Methods for a detailed description of the full algorithm.

Importantly, *any* importance score can be used: while poor importance scores will lead to low power, the FDR will remain controlled. This property has strong consequences for the method’s versatility and validity. First, the relationship between Y and X does not need to be understood or modeled properly. For instance, an importance score built on a linear model will still provide calibrated p-values when the relationship between Y and X is non-linear. Second, importance scores can be built from the output of existing feature selection algorithms that do not inherently provide p-values, allowing VI-VS to act as a meta-algorithm that makes interpretable decisions on top of these algorithms.

Multi-resolution approach to feature detection In practice, conditional independence tests may not detect many features, e.g., due to limited sample sizes when some features are highly correlated. Consider a hypothetical experiment where features form two clusters and there is a function of one feature in each cluster (Fig-

ure 3.2B). A conditional independence test may fail to detect either of these features if there are strong correlations within the clusters (see, for instance, [139]). VI-VS includes a multi-resolution testing procedure that identifies conditionally dependent feature groups in addition to individual features to address the power issue of conditional independence tests. This allows for the detection of feature groups for which no individual feature can be identified as conditionally dependent on the response variable, allowing to avoid missing relevant associations. Figure 3.2C provides a preview of the output of this approach in the illustrative example above.

Implementation We implemented VI-VS in a fast and scalable fashion, parallelizing synthetic data generation across genes and samples via GPU acceleration (Additional file 1: Algorithm S1). The algorithm is implemented in Jax and is available as a Python package at <https://github.com/YosefLab/VIVS>.

Conditional randomization tests for single-cell genomics

As an input, VI-VS receives a matrix of features $X \in \mathbb{R}^{N \times G}$ and a vector, representing a response variable $y \in \mathbb{R}^N$ where G is the number of features and N is the number of cells. We also assume that observed nuisance factors $S \in \mathbb{R}^{N \times T}$, e.g., batch assignments, sequencing depths, or cell cycle events, affect these experiments and need to be accounted for. Our goal is to detect features in X that are associated with the response variable y while controlling for the nuisance factors.

In the following, we assume that X consists of observed molecular expressions of G genes in N cells, although the method is general and applies to other modalities. The choice of y varies depending on the assay considered and the problem of interest. Specifically, y can characterize molecular quantities, such as protein counts in CITE-seq experiments or chromatin accessibility in ATAC-seq data. It can also represent other, more abstract cell-level properties, e.g., characterizing the tissue environment of a cell in spatial transcriptomic assays. y may also correspond to a singled-out gene of interest, for which we wish to identify the interacting genes.

When referring to observations from an individual cell, we will employ lowercase letters, reserving uppercase letters for the entire array of observations. In addition, $x_g \in \mathbb{N}$ and $x_{-g} \in \mathbb{N}^{G-1}$ will respectively denote gene expressions for gene g and the vector of remaining genes. When needed, superscripts will index cells, such that x^n denotes the gene expression vector $[x_1^n, \dots, x_G^n]^T$ of cell n . When A is a set of features, x_A will denote the vector of features contained in A . We make the assumption that the samples (x^n, y^n, s^n) are independent and identically distributed (i.i.d.).

To detect genes in X that are associated with the response variable y , VI-VS employs a *conditional* independence test, which estimates, for each gene, the plausibility

of the null:

$$\mathcal{H}_{0,g} : x_g \perp\!\!\!\perp y \mid x_{-g}, s. \quad (3.1)$$

We rely on the conditional randomization test (CRT) approach [39] to test these hypotheses. The premise of CRT is that while it is difficult to directly assess how the distribution of the response variable y depends on X , it is easier to describe how the features of X depend on each other. VI-VS requires two ingredients: a *generative model* for X to capture the dependencies between features, and an *importance score* to evaluate their association with y .

Importance score The importance score is a function $T : X, Y, S \rightarrow \mathbb{R}$, which summarizes the observed data. To make decisions, the CRT compares this summary $T(X, Y, S)$, with $T(\tilde{X}, Y, S)$, where $\tilde{X}$ denotes partially synthetic data in which one or few of the features are replaced with values that are generated with the generative model. Here, we propose to *learn* the importance scores from the data. In particular, we consider importance scores corresponding to the prediction error of a regression model of Y on X and s ,

$$T_\phi(X, Y, S) = \frac{1}{N} \sum_{n=1}^N -\log p_\phi(y^n \mid x^n, s^n), \quad (3.2)$$

where (y^n, x^n, s^n) respectively denote responses, gene expression, and nuisance factors for cell n . Here, $p_\phi(y^n \mid x^n, s^n)$ is a likelihood for y based on a model p_ϕ trained on held-out data. For instance, p_ϕ may be based on a linear regression, or more complex models such as random forest or a multi-layer perceptron (MLP). Importantly, this predictive model does not need to perfectly capture the conditional distribution of y given X, s . The CRT will indeed control the false positive rate irrespective of the choice of the predictor model and its assumptions on the nature of the interaction between X and y or on the distribution of y [39]. However, the more adequate the model, the more powerful we can expect the test to be.

Generative model The other required component is a generative model p_θ that (i) can be used to sample *synthetic* expression profiles for a given gene, and (ii) does not depend on the response variable y . Due to their scalability, ability to capture nonlinear effects, and flexible likelihood assumptions, latent variable models are a useful choice to model gene expression in this context. In these models, an unobserved low-dimensional variable z is assumed to capture the state of each cell and provide a

concise summary of the biological variation among cells. We assume that the model factorizes, for each individual cell and under i.i.d. assumptions, as

$$p(x, z \mid s) = p(z) \left(\prod_{g=1}^G p(x_g \mid z, s) \right), \quad (3.3)$$

where $p(z)$ is the latent variable prior, and $p(x_g \mid z, s)$ is the likelihood for gene g . We rely on variational autoencoders (VAEs) to define the latent variable model. In this model, the prior is usually the standard normal, and the posterior distribution is approximated using a variational approach, with the approximation parameterized by neural networks [22, 25, 27]. Assuming access to such a model, testing $\mathcal{H}_{0,g}$ requires replacing the measurements for the feature g , with synthetic measurements that are conditionally independent of y . To this end, we use the generative model to obtain K vectors $\tilde{X}_g^{(k)}, k \leq K$, containing synthetic counts for gene g for all the cells in a manner independent of y . Here, superscripts in parentheses denote Monte Carlo samples. We then construct the overall gene expression for which gene g was randomized, as

$$\tilde{\mathbf{X}}^{(k)} := [X_1 \dots X_{g-1}, \tilde{\mathbf{X}}_g^{(k)}, X_{g+1}, \dots X_G]^T, \quad 1 \leq k \leq K. \quad (3.4)$$

With these two components, a p-value for $\mathcal{H}_{0,g}$ with the CRT corresponds to the proportion of random trials in which the importance score, when gene g is replaced with synthetic data, is not worse than the score obtained with the original data. It writes as

$$p_g = \frac{1}{K+1} \left(1 + \sum_{k=1}^K \mathbb{I} \left(T(\tilde{\mathbf{X}}^{(k)}, Y, S) \leq T(\mathbf{X}, Y, S) \right) \right). \quad (3.5)$$

Valid inference for CRTs with latent variable models

Given a latent variable model, an intuitive way to generate synthetic samples $\tilde{X}_g$ is by independent draws from the Gibbs distribution:

$$\tilde{x}_g^n \sim p_\theta(x_g^n \mid x_{-g}^n, s) = \int p(x_g \mid z) p(z \mid x_{-g}^n, s) dz,$$

This choice, however, requires sampling from $p(z \mid x_{-g}, s)$. In the context of VAEs, this requires training a separate model for every feature g , which is in most cases computationally prohibitive. Instead, VI-VS provides a fast and valid sampling alternative that still provides valid p-values. This is done by drawing *fixed* posterior

sample of z . Here, for each cell n , we first sample one particle from $\bar{z} \sim q(z \mid x^n)$ where q is the encoder network of the VAE. We then rely on the decoder network of the VAE to obtain synthetic samples:

$$\tilde{x}_g^{(k)} \stackrel{\text{iid}}{\sim} p_\theta(x_g \mid z = \bar{z}, s), \quad k \leq K. \quad (3.6)$$

Note that in both cases the generative model does not have access to the value of y during sampling, The samples $\tilde{\mathbf{X}}^{(k)}$ therefore reflect a hypothetical reality in which x_g and y are conditionally independent. In Proposition 1 we demonstrate that both sampling schemes provide valid p-values for the CRT (proof in Additional file 1).

Proposition 1 (Valid sampling distributions for CRTs with latent variable models). *Assume a latent variable model $p_\theta(x, z \mid s)$, factorizing as (3.3). Let $\tilde{X}_g = [\tilde{x}_g^1, \dots, \tilde{x}_g^K]^T$ be a vector of synthetic gene expression profiles generated using the latent variable model for gene g obtained using either of the two sampling schemes described above. Then, the p-values p_g in Equation (3.5) have a distribution that stochastically dominates the uniform distribution when the null hypothesis $\mathcal{H}_{0,g}$ holds. That is, p_g is a valid p-value.*

The entire procedure can therefore be summarized as follows:

Algorithm 1 Conditional randomization tests with VI-VS

Require: Dataset $\mathcal{D} = \{X, Y, S\}$, importance score function $T_\phi(X, Y, S)$, variational autoencoder p_θ (both trained on held-out data), sample budget K .

for gene $g \leq G$ **do**

$\bar{Z} \sim p_\theta(Z \mid \mathbf{X}, S)$

for sample $k \leq K$ **do**

$\tilde{X}_g^{(k)} \sim p_\theta(X_g \mid \bar{Z}, S)$

$\tilde{\mathbf{X}}^k \leftarrow [X_1 \dots X_{g-1}, \tilde{X}_g^k, X_{g+1}, \dots, X_G]^T$

end for

$p_g = \frac{1}{K+1} \left(1 + \sum_{k=1}^K \mathbb{I} \left(T_\phi(\tilde{\mathbf{X}}^k, Y, S) \leq T_\phi(\mathbf{X}, Y, S) \right) \right)$

end for

Return p-values $\{p_1, \dots, p_G\}$

VI-VS further corrects the obtained p-values p_g using the Benjamini-Hochberg procedure [37] to control the false discovery rate (FDR).

Cell-specific scores Equation (3.5) quantifies the significance of the association between gene g and protein p ; it does not, however, inform on which cell subpopulations

may be most responsible for this association. For this purpose, we introduce the cell-specific score,

$$s_g(x, y, s) := \frac{1}{K} \left[\sum_{k=1}^K T(\tilde{x}^{(k)}, y, s) \right] - T(x, y, s), \quad (3.7)$$

where $\tilde{x}^{(k)}$ denotes a randomized sample for the CRT. In other words, positive scores will highlight that randomizing the gene g in cell x increases the predictive loss, which may mean that this gene plays a relevant role in the prediction y for the considered cell.

Multi-resolution hypothesis testing

Conditional dependence is a more stringent statistical notion than marginal dependence. Consequently, applying conditional independence tests at the gene level may not yield many significant genes. This could be due to several factors, such as limited sample sizes or strong correlations between genes that make it challenging to reject the conditional null.

In scenarios with small sample sizes it might prove challenging to detect a true positive gene if it heavily correlates with other genes in its cluster. However, it is easier to detect that one or more genes in the cluster are conditionally associated with the response, even if we cannot definitively identify which genes are responsible.

Therefore, following an approach introduced for genome-wide association studies [140], we test for conditional independence at multiple resolutions, ranging from broad groups of genes to the individual gene-level resolution, to avoid overlooking genes of interest.

To further illustrate how VI-VS behaves as correlation between features increases, we devised a simple simulation study, where a synthetic response Y is conditionally dependent with one feature g , which itself correlates with another feature g' (Additional file 1: Figure 3.14). Briefly, as the correlation between features increases, it is not possible to detect the conditionally dependent relationship between Y and g anymore. However, at a coarser resolution, the group of genes g and g' is still detected as conditionally dependent with Y .

We will now explain how we: (i) group genes together, and (ii) test for conditional independence of a group of genes.

Determining relevant clusters of genes Our goal is to group together features associated with the same biological functions. We assume that high correlations between features may indicate that they are associated with the same biological

function. Consequently, we cluster genes based on their empirical correlation matrix. Any gene clustering algorithm can be used in principle, e.g. [141]. We propose using a fast hierarchical clustering approach that is scalable to large datasets. This approach performs agglomerative clustering based on the gene-by-gene empirical correlation matrix computed on the normalized gene expression of the generative model, e.g., scVI [22]. More details about this procedure can be found in Additional file 1: Supplement 3.5. At a specified resolution K , the clustering provides a partition M of all genes into K groups of genes $A_1, \dots, A_K$.

Group conditional independence Next, we aim to determine whether a cluster of genes is significant. To formalize this, let $A \in M$ denote a group of genes. We are interested in interactions of the form

$$\mathcal{H}_{0,A}^M : x_A \perp\!\!\!\perp y \mid x_{A^C}, \quad (3.8)$$

where x_A , x_{A^C} denote the gene expression vectors for the genes in sets A and its complement A^C , respectively. We test this null hypothesis using the same procedure as described above, with more details provided in Supplement 3.5 and illustrated in Additional file 1: Figure 3.15. Specifically, we can test Equation 3.8 by sampling from the same distribution as in Algorithm 1.

Faster inference using parallel computing

We implemented VI-VS in a fast and scalable way that is available as an open-source Python package. The scalability of this solution relies on two components. Our implementation first relies on parallel computing and just-in-time compilation components of Jax to speed up the inference, allowing us to efficiently compute the p-values for all genes. This practical choice offered a two-fold speedup compared to a Pytorch backend (Additional file 1: Figure 3.16). The second key component is the fact that we have set up our algorithm to avoid fitting a model for each Monte Carlo sample, and instead, we only have to perform a forward pass of the pre-fit feature statistic. This computational ingredient improves the run time by orders of magnitude, thus improving scalability.

3.3 Results

VI-VS provides calibrated decisions in a semi-synthetic experiment

We considered a scRNA-seq dataset of 6,855 peripheral blood mono-nuclear cells (PBMC) [90] from a healthy human donor, with 500 genes. We generated five synthetic response variables, each corresponding to the expression of a surface protein measured in the observed cells. The expectations of each response variable were calculated as a linear combination of the *squared* values of mean-centered log count per million (CPM) expression levels of 150 randomly selected genes in X . These simulated response variables were further corrupted by the addition of Gaussian noise. This simulation represents a case where simple linear assumptions do not hold, but there is still a relatively simple model that connects y to a subset of features in X . More details on data generation can be found in Additional file 1: Supplement 3.5.

We compared VI-VS to two baselines. Ordinary Least Squares (OLS) is a canonical method for conditional independence testing. For OLS, we regressed the response variables y on X , under linear and Gaussian assumptions, and used a t-test to estimate significance of each coefficient. We also considered a simpler (marginal) independence test baseline. For each gene g , we regressed y on the expression of g only and used a t-test to estimate significance. These two baselines used all available data, i.e., both the development and validation splits, to fit the regression models, thus lending a natural advantage over the way VI-VS was fit.

We first evaluated the extent of type I error of the different algorithms (Figure 3.3A). We found that the FDR estimates of the marginal independence test exceeded target levels, leading to many false positives. These false positives likely reflect indirect correlations, that is, genes that were not used to generate the synthetic response variable but strongly correlate with other genes used for data generation.

The OLS performed better but still overestimated the FDR, possibly due to the violated linearity assumptions. In addition, likelihood misspecification, i.e., invalid Gaussian assumptions on the data, can cause FDR miscalibration for OLS. As an illustration, we repeated the simulation, this time generating the response variable counts from a Poisson distribution (Additional file 1: Figure 3.8), in which case OLS performs worse. On the other hand, the application of VI-VS with an MLP for the importance score controlled the FDR in both these scenarios. We also evaluated the robustness of our approach to key characteristics of the simulated data, including sparsity and the number of genes in the assay, and show that our approach compares favorably to OLS, and provides consistent FDR control across all parameter settings (Additional file 1: Figure 3.9). Finally, we confirmed the robustness of VI-VS to

the choice of development set size in Additional file 1: Figure 3.10, where we also suggested strategies to choose the development set size and assess calibration when necessary.

Increased power using more complex importance score functions Using better importance scores can increase the power of the CRT framework (i.e., lower type II error), while still maintaining calibration of the type I error estimates (as stated by Proposition 1 in the Methods and [39]). To illustrate this, we considered a range of increasingly complex model choices to compute the importance scores and repeated our simulation analysis (Figure 3.3B). We found that all importance scores controlled the FDR at different levels. For instance, using linear regression with an OLS objective to compute importance scores still controlled the FDR, although this model is a poor predictor of the response variable. Conversely, higher-capacity models relying on complex MLP architectures led to increased power, indicating that more pertinent importance scores may lead to more discoveries. We also found that the models with the best held-out predictive performance also detected more true positives (Figure 3.3C), providing an empirical strategy to design importance scores with VI-VS. Therefore, we advise using predictive performance as a criterion to select the importance score to use with VI-VS.

Using VI-VS to calibrate existing algorithms for the identification of feature interactions in single-cell genomics Our framework can build interpretable decisions on top of existing algorithms that lack a scalable or otherwise principled way to define calibrated decision rules. An example of such a model is GENIE3, which uses an ensemble of regression trees to produce scores that quantify the importance of each gene in predicting a held-out “response” gene, thus identifying putative interactions between genes. These scores were shown to have state-of-art performance in ranking putative interactions from the most to the least relevant. They, unfortunately, do not easily inform which interactions should be considered relevant for decision-making. Consequently, we used VI-VS to construct interpretable decisions on top of GENIE3. To do so, it sufficed to plug in the GENIE3 model as importance score. Specifically, given a response variable y , we trained GENIE3 regression trees once using the development part of the data. We then used the respective prediction errors on the validation data as importance scores for VI-VS (Figure 3.3D; Additional file 1: Supplement 3.5 for details). Application to our simulated data demonstrates that this wrapper procedure provides decision rules that control the FDR at several levels (Figure 3.3E), while still providing large true positive rates. The CRT framework can therefore be used to better utilize a large family of algorithms in this area, as

long as they produce an estimate of interaction “strength” that considers all features in X serving as plugin importance score.

Multi-resolution testing as a way to increase power The limited true positive rate in our simulation results can be explained to some extent by gene correlations that could not be resolved because of the limited size of the data. We next tested whether multiresolution could mitigate this. To this end, we applied our hierarchical procedure with different gene cluster granularities (here, 200 clusters, 300 clusters, or a per-gene analysis; Figure 3.4A). We generally observed that the detections were consistent across the different resolutions, i.e., if a group of genes was identified at a given resolution, groups containing these genes were detected at coarser resolutions. Testing at multiple resolutions is useful to identify clusters of genes that were not detected at the gene level due to sample size limitations. Clusters 5 and 6 are such examples, illustrating cases where relevant genes might not be detected at the gene level, but could be detected at coarser resolutions.

For a more quantitative evaluation of the merits of multi-resolution testing, we computed FDR and power, where a selected group of genes was considered true positive if it contained at least one gene that was a true positive (at the gene level) and false positive otherwise (Figure 3.4B). Our results show that testing at coarser resolutions, i.e., grouping genes together, yielded more discoveries while maintaining calibrated FDR. More generally, multi-resolution testing provides more insight into the statistical relationships between genes and responses. When a coarse gene group is detected, the identification of finer-grained gene groups can be used to identify the individual genes that are responsible for the association [142]. Tying individual genes with coarser gene groups can also help identify and annotate genes with additional sources of information [143].

VI-VS identifies causal interactions in perturb-seq data

We also considered a perturbational screen assay [144] to assess the relevance of VI-VS in identifying causal gene-gene interactions in single-cell genomics. As they can identify true causal interactions between genes, perturbational assays constitute a natural choice to benchmark the different models to produce causal candidates. To benchmark our approach, we restricted the analysis to the ten target genes with the largest number of guides. We first identified for each target gene true positive genes as differentially expressed genes using a t-test comparing gene expression between cells with and without guide RNA. We then fit the different models on unperturbed cells using the measured target gene expression as the response variable.

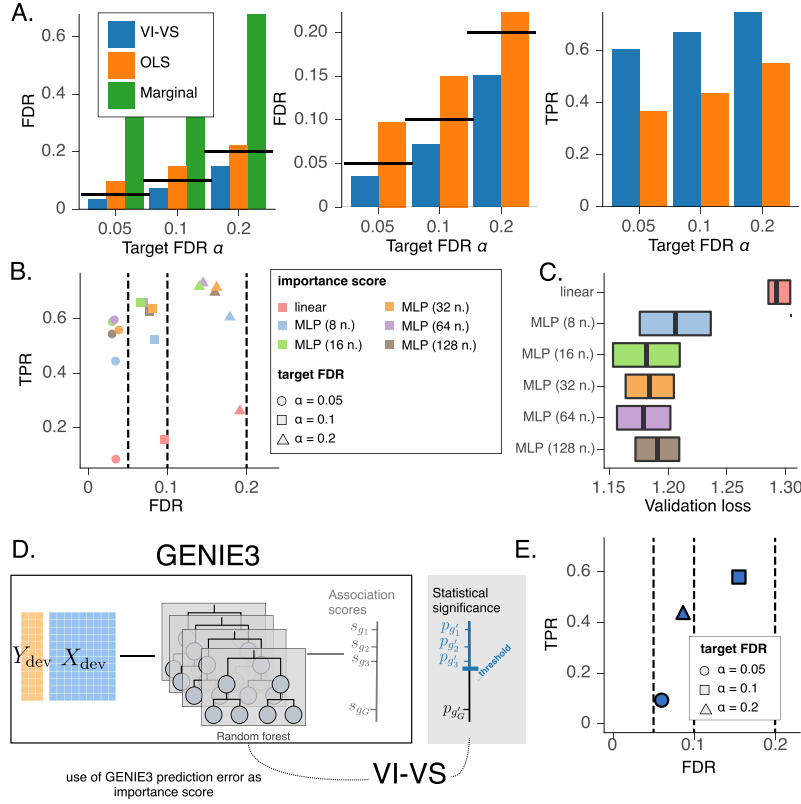


Figure 3.3: Semi-synthetic experiment. **A.** Comparison of FDR control and power for conditional independence testing at the *gene* level, averaged over five random weights initializations and the five surface proteins. Here, VI-VS uses a neural network with 64 units to compute importance scores. *Left*: FDR control comparison for the CRT, Ordinary Least Squares (OLS) under t-tests, and marginal independence tests. Because the marginal test did not control the FDR, it was removed from the rest of the experiments. *Center*: Zoom on the previous figure. *Right*: Associated TPR. **B.** FDR-TPR curves for different importance scores averaged over five random weights initializations and the surface proteins. **C.** Associated held-out mean squared error of the different regression models used as importance scores. **D.** Use of VI-VS as a calibration tool for GENIE3. After fitting the regression tree ensemble of GENIE3, we used their prediction error as importance scores for VI-VS, allowing one to detect conditionally dependent genes with statistical significance. **E.** FDR/TPR levels of VI-VS using GENIE3 reconstruction losses as importance scores.

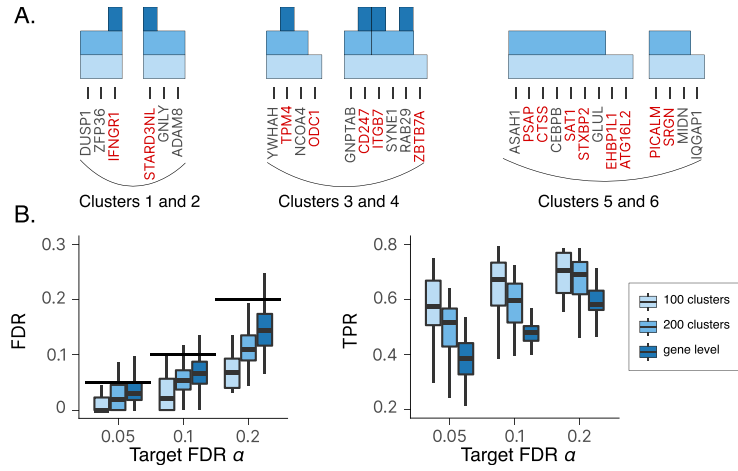


Figure 3.4: **A.** Examples of gene groups identified by VI-VS in the semi-synthetic experiment. Clusters 1 and 2 show examples where all genes affecting the surface protein expression in the simulation are detected at the gene level. Clusters 3 and 4 show examples where some of the genes are not detected at the gene level but are detected at coarser resolutions. Clusters 5 and 6 show examples where none of the genes are detected at the gene level, but are detected at coarser resolutions. **B.** FDR (*left*) and power (*right*) for VI-VS applied at different resolutions. When testing at the group level, a group of genes was considered a true positive if it contained at least one gene that was a true positive at the gene level.

Figure 3.5 shows the estimated FDR and TPR for the different models evaluating the match between t-test of perturbed and unperturbed cells and genes significant predictive genes estimated by the different models. As expected, the marginal approach had the highest TPR, but also the highest FDR. VI-VS obtained a significantly lower FDR than both OLS and the marginal approach (70% for VI-VS; 95% for OLS). In other words, almost a third of the associations identified by VI-VS were true positives, which is a major increase over the other models. The improved precision of VI-VS did not affect its power. VI-VS identified a similar number of true positives as OLS, but from a smaller pool of detected associations (Additional file 1: Figure 3.11). In other words, our approach produced a smaller set of associations with high precision, which contained a similar number of true positives as the larger set of associations produced by OLS. The superiority of VI-VS over OLS for causal candidate identification was further supported by significantly higher F1 scores (Additional file 1: Figure 3.11).

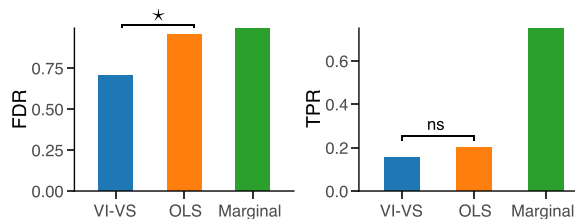


Figure 3.5: Perturb-seq experiment. *Left*: FDR and *Right*: TPR for the different models for the identification of causal interactions identified from the perturbational assay estimated from the ten target genes with the largest number of guides in the assay (target FDR level: $\alpha = 0.1$). A star (*) indicates a significant difference between the metrics based on a paired t-test with a significance level of 0.05; nc indicates that the difference is not significant.

VI-VS identifies links between surface proteins and gene expression programs with CITE-seq

Next, we considered a CITE-seq dataset of PBMCs, obtained from eight healthy human donors [145]. We applied VI-VS at the single gene level, as well as the OLS baseline to a subset of this data with a total of 50,000 cells, 2,000 genes, and 224 surface proteins (Figure 3.6A). Notably, this dataset includes information from 13 batches, which could be accommodated by VI-VS due to the batch correction capacity built into its generative model [22]. Application of OLS to this case study (using an FDR cutoff of 10%) returned a very large number of associations for all 224 proteins (Additional file 1: Figure 3.12A). In contrast, VI-VS (with the same target FDR) only predicted associations for 51 of the proteins, with a smaller number of interactions per protein (52 interactions on average, compared to 140 interactions with OLS). To understand this, we first considered the set of proteins that had no associations with VI-VS. Using TotalVI [30], we estimated for each protein the percentage of cells that plausibly express it on their surface (accounting for background signal, which is often observed in protein quantification with CITE-seq). We found that those proteins with no detection by VI-VS tend to have a much lower signal, compared to the ones that have been associated with gene expression (Additional file 1: Figure 3.12B). Conversely, the OLS analysis identified associations for proteins that are likely not well captured or not expressed in these settings. OLS detected numerous associations (21, 41, 121, and 123 gene-protein pairs respectively) for four negative control proteins (Rat-IgG1-1, Rat-IgG2b, Rat-IgG1-2, and Rat-IgG2c) that are not expressed by human cells. In contrast, VI-VS detected none of these associations.

To further compare the validity of the associations made by VI-VS and OLS, we used StringDB [146] to evaluate the a priori support for each interaction. Specifically, we assigned StringDB’s protein-protein “combined score” to each protein-gene pair. This combined score is a composite measure that integrates the scores of protein-protein associations computed across several modalities. We first compared the distribution of these scores for predicted gene-protein interactions across all proteins (Figure 3.6B). We found that the scores of the interactions predicted by VI-VS were significantly higher than the ones identified by OLS (Kolmogorov-Smirnov test, $P \leq 10^{-6}$). A similar trend was observed when comparing these scores for proteins for which both methods made predictions (Kolmogorov-Smirnov test, $P \leq 0.05$). This, combined with the fact that OLS detected associations for several negative control proteins, suggests that OLS is likely misspecified and may return many false positives. Conversely, VI-VS provides a more conservative and accurate way to identify biologically meaningful associations.

Locating protein-gene associations to the relevant cell subsets A core feature of VI-VS is the ability to not only identify the association of genes with the response variable but also highlight the set of cells in which this interaction is more likely to be relevant. As a first example of this, we consider an association detected between MS4A1 (encoding the B cell marker CD20) and the HLA-DR receptor. Using the cell-specific importance scores from Equation 3.7 of the Methods, we identified B cells as the most relevant cells for this association (Figure 3.6C). This agrees with previous findings on the physical and functional association between CD20 and MHC-II in activated B cells [147], and the use of these two molecules as joint targets for combination therapy in lymphomas [148]. VI-VS also identified an association between the presentation of CD48 on the cell membrane and the expression of 2B4, which encodes the activating NK cell receptor CD244. The cell-specific importance scores suggest that this dependency is primarily driven by natural killer (NK) cells (Figure 3.6D). This result agrees with reports on the functional association between CD48 and CD244 in NK cells, where direct binding of these molecules is important to drive the surface expression and phosphorylation of CD244 in NK cells, consequently affecting their effector function [149].

When the practitioner has prior knowledge about the cell types of interest for the analysis, it is advantageous to fit the model only on these specific cells rather than the entire dataset. This choice reduces the computational cost of the algorithm and yields clean type-specific associations, eliminating the need for post-processing based on cell-specific importance scores. To illustrate how VI-VS can unveil biologically relevant associations at multiple resolutions, we searched for associations between

genes and proteins in T cells specified before testing. We focused on the CD25 surface protein (IL2RA) and identified conditionally dependent genes and gene groups at different resolutions. For a coarse resolution ($K = 100$), VI-VS detected 26 groups of genes, seven of which contained genes known to be involved in the regulation of IL2RA [150], which we visualized in Figure 3.6E. In addition to IL2RA and IL2RB, detections at the gene level included CCR5, which encodes a chemokine receptor that influences IL2 production in T cells [151]. Testing at several resolutions simultaneously identified causal genes that were not detected at the gene level, presumably due to sample size limitations or strong correlations. For instance, STAT3 and STAT5B, two transcription factors involved in the regulation of IL2RA, were not detected by VI-VS at the gene level but detected at a coarser resolution. STAT3 promotes T cell survival and is known to inhibit T cell proliferation and IL2 production [152]. The activation of STAT5B by IL2 cytokines is a critical signaling pathway associated with regulatory T cell differentiation and function [153]. We generalized this analysis to other proteins, and compared the number of detected genes contained in known pathways for VI-VS and OLS more quantitatively (Figure 3.6F). VI-VS detections at coarser resolutions detected more overlapping genes contained in pathways, while tests at finer resolutions provided more precise gene-level associations and overall more overlapping genes than OLS.

An important observation is that contrary to marginal approaches, VI-VS automatically controls for cell-type variation. A gene and a protein may be marginally dependent if they are expressed by the same cell types, even if they do not correlate within these types. Additional file 1: Figure 3.13 compares, for every gene, significance scores for its association with the surface protein CD4, using a marginal test and VI-VS with a significance score of DE between CD4+ T cells and the rest of the cells. The marginal approach has a very strong correlation with the cell-type variation, while VI-VS does not. Consequently, a marginal test for gene-protein association may reflect cell-type variation rather than molecular interactions. Conversely, since gene expression data from all genes except one are sufficient to identify cell types, VI-VS conditions the variation between cell types and finds associations that are not explained by cell-type variation.

VI-VS identifies spatially-dependent gene expression programs in lymphocytes using ST

We showcased how VI-VS can be applied for spatial transcriptomic (ST) analyses. In particular, we studied an ST dataset consisting of one lung biopsy from a non-small cell lung cancer (NSCLC) patient, containing 960 genes and 200,000 cells, sequenced

using the CosMx platform [154] and segmented with Baysor [155]. In this case, our objective was to link gene expression to spatial contexts that reflect cell localization in the tissue or its proximity to other cells.

Characterizing spatial differential expression patterns for T cells We first aimed to identify differences in gene signature between T cells located in the tumor and lymphoid aggregates (Figure 3.7A). We trained the considered models on these cells using a binary response variable indicating the cell location ($y = 1$ for tumor cells, 0 for lymphoid aggregate cells); the importance score used by VI-VS was modified to the log-likelihood of a neural network binary classification model. We first compared the number of genes predicted by VI-VS to simple parametric and nonparametric differential expression tests (Additional file 1: Table 3.2). The latter approaches detected almost all genes in the dataset. As the number of cells increases, negligible differences in gene expression are likely to be detected as significant, even after multiplicity correction. This is a known problem for point null hypothesis tests applied to single-cells [33, 156], which would require further filtering of the results to obtain a reasonable number of discoveries that can be interpreted. In contrast, VI-VS detected a much smaller number of genes at the gene level. Indeed, only five genes were detected by VI-VS at the gene level, including ITGAE and IL7R (Figure 3.7B). ITGAE, encoding CD103, is a canonical marker of tissue-resident memory CD8⁺ T cells (Trm). Its expression characterizes T cell infiltration in the tumor microenvironment (TME) [157]. This result highlights that tissue-resident memory T cells preferentially infiltrate the tumor. IL7R is a general marker for memory CD4 T cells. The multiresolution approach detected a larger set of genes known to capture known tumor-specific T cell signatures. ITGAE is associated with a group of genes that have a cytotoxic function in CD8 cytotoxic T cells (CTSW, GNLY, NKG7, PRF1, GZMB, KLRK1) up-regulated in the tumor. Studying this gene module therefore identifies resident CD8 T cells in the tumor to have a highly cytotoxic and activated phenotype. IL7R on the contrary shows up in a module with CCR7 and KLF2, genes that mark central-memory CD4 T cells. This module therefore identifies central-memory CD4 T cells to be enriched outside of the tumor. These genes were mainly located in spatial clusters of lymphocytes which we identified as lymphoid aggregates. In general, the genes and modules detected reflect a diverse set of biological processes varying across lymphoid aggregates and tumor regions.

Identifying T cell genes associated with tumor proximity Last, we characterized the associations between gene expression and tumor proximity by defining y as the local density of tumor cells surrounding T cells. The first step was to define the

range of tumor proximity we considered relevant. To do so, we constructed several responses y , corresponding to the predicted tumor density for each T cell predicted by a Gaussian kernel density estimator of different bandwidths, taking values in $h \in [10\mu m, 50\mu m, 100\mu m, 200\mu m, 500\mu m, 1000\mu m]$. Of these, only $h = 500\mu m$ and $h = 1000\mu m$ detected associations at the gene resolution, suggesting that gene expression interactions with immediate and close tumor cells are more difficult to detect and would require more data to reach significance. We focused on the discoveries made by VI-VS for $h = 500\mu m$ (Figure 3.7C), and visualized the detections, corresponding to six gene groups (Figure 3.7D). Our approach provided a significant number of genes related to T cell function in the tumor microenvironment, including HSP90AA1, ETS1, CXCR4, RGS1, and FYN, all detected at the gene level. HSP90AA1, for instance, encodes a heat shock protein, whose overexpression correlates with tumor progression and a poor prognosis in NSCLC [158, 159], and has been shown to correlate with an exhausted phenotype of CD8 T cells in tumors [160]. CXCR4 encodes a chemokine receptor whose expression is associated with the formation of lymphoid follicles that we detected outside of the tumor [161]. TGFBR2 is associated with this gene at the module level and has been shown to induce the residency of T cells in lymphoid tissue [162]. A correlation of RGS1 with T cell exhaustion has been observed in various cancers, including NSCLC [163]. We emphasize here that the location of T cells outside of the tumor is related to specific chemokine signals, specific cell states, and markers of exhaustion, whereas the location inside the tumor is related to an increase in heat shock protein signatures associated with T cell exhaustion and cellular stress. These results suggest that VI-VS is flexible to be applied to continuous descriptions of spatial localization and then helps to dissect function without prior knowledge of important tissue niches.

3.4 Discussion

Detecting conditional dependencies requires more data than identifying marginal dependencies [164]. This requirement may cause conditional approaches to miss potentially relevant associations due to limited statistical power. To address this risk, we proposed a multiresolution testing procedure. This procedure not only identifies individual genes with conditionally dependent features, but also recognizes feature groups that may contain them, providing a comprehensive characterization of the statistical dependencies between features and responses. We highlight throughout the manuscript that these modules can help in identifying the functional role of an identified molecule and thereby help in interpreting the results.

In the general case, VI-VS discoveries have no guarantees to be functional. Feed-

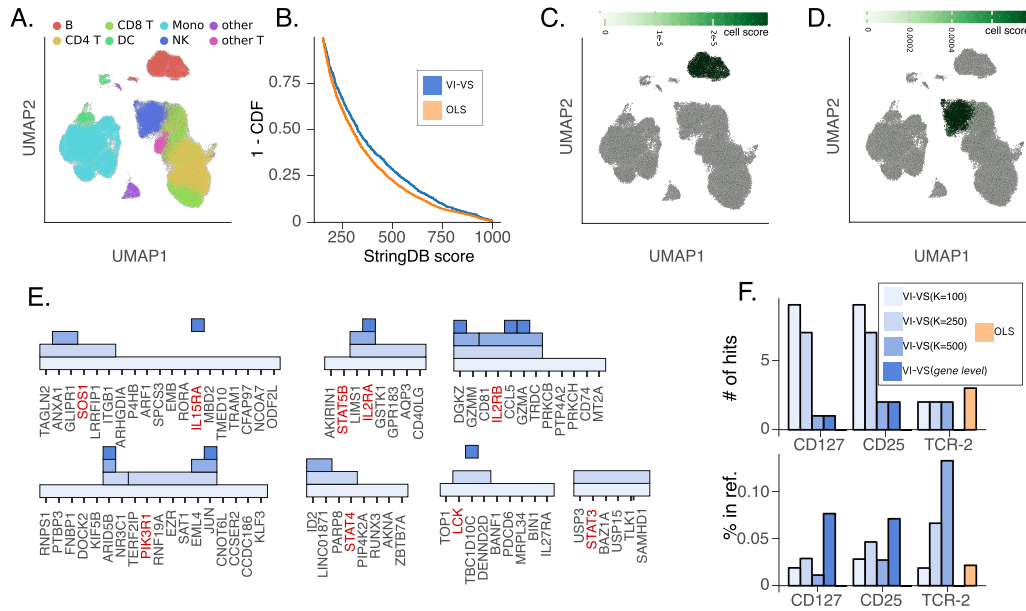


Figure 3.6: CiteSeq experiment. **A.** UMAP of the dataset. **B.** Distribution of stringDB scores of gene-protein discoveries made by VI-VS and least-squares (higher is better). **C.** and **D.** cell scores (averaged per cell-type) for the CD86-HLA and CD48-2b4 gene-protein pairs detected by VI-VS. High scores identify cells where the dependency is most expressed. **E.** Visualization of VI-VS detections at several resolutions for surface protein CD25 and T cells. Each filled rectangle characterizes a gene group detected as significant by VI-VS when testing for conditional independence at several resolutions ($K \in \{100, 250, 500\}$ and at gene level). Genes in red correspond to genes contained in *Interleukin-2 Family Signaling R-HSA-451927* or *Interleukin-2 Signaling R-HSA-9020558* pathways. **F.** Agreement of the predictions with the Reactome pathway database, focusing on T cells for three surface proteins. *Top*: Number of predicted genes contained in each pathway, and *bottom*: proportion of predicted genes contained in each pathway over the total number of detections. The following pathways were considered: *Interleukin-7 Signaling R-HSA-1266695* for CD127, *Interleukin-2 Family Signaling R-HSA-451927* and *Interleukin-2 Signaling R-HSA-9020558* for CD25, and *TCR Signaling R-HSA-202403* for TCR-2. For **E.** and **F.**, models were fit on T cells only.

back loops prevalent in molecular interactions, cell communication, or unobserved molecular species are just a few examples of phenomena that lead to spurious discover-

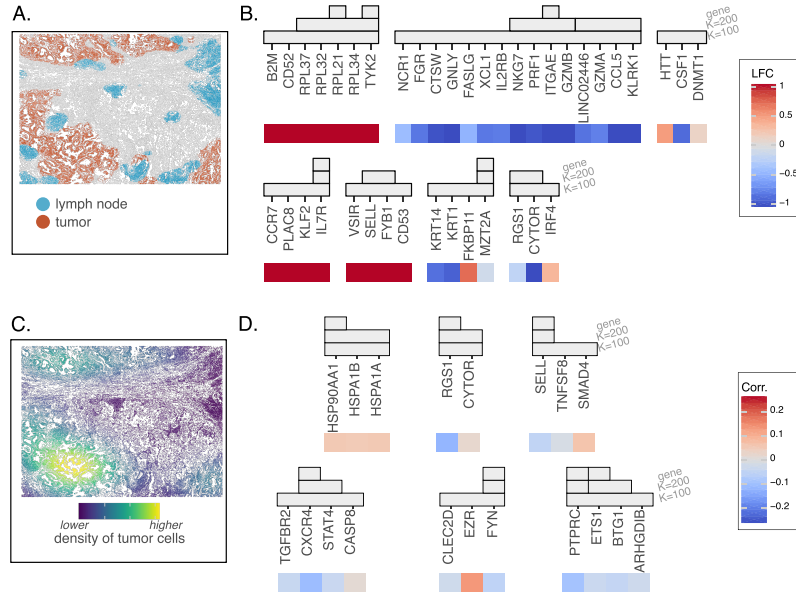


Figure 3.7: Application of VI-VS to spatial transcriptomic data. **A.** Tissue segmentation into lymph nodes and tumor regions. **B.** Identified spatial DE genes by VI-VS in T cells, along with spatial LFCs (positive values denote gene upregulation in lymph nodes compared to tumors) against the significance scores of VI-VS. **C.** Local density of tumor cells in the tissue. Density is estimated using kernel density estimation (bandwidth of $500\mu m$). **D.** Identified T cell genes conditionally associated with local density of tumor cells, along with marginal Spearman scores between gene expression and local tumor density.

ies. However, certain multiomic setups already offer promising avenues for identifying causal relationships with VI-VS. Identifying reproducible and robust discoveries across biologically diverse environments could help mitigate the effect of unobserved confounders [165, 166].

Making no assumption on the distribution of the response, our approach can readily be applied to other multiomic setups. A first application could be the identification of gene associations with metabolites [167]. VI-VS could also be applied more broadly to spatial transcriptomics to create complex characterizations of cell phenotypes and their environments. It could, for instance, pinpoint genes involved in receptor-ligand interactions [168], or in determining cellular morphologies [169]. VI-VS is also relevant for the identification of potential transcription factor binding sites, using paired gene

expression and chromatin accessibility data. A major advantage of VI-VS in this scenario is its ability to identify broader regulatory regions associated with gene expression, even when there is not enough data to pinpoint individual peaks.

The primary assumption underpinning VI-VS is the availability of a valid generative model of the feature data. The generative model should be capable of generating synthetic data that is statistically indistinguishable from the observed data. The generative model considered in this manuscript has undergone rigorous stress-testing for single-cell RNA data generation and imputation [60, 70, 170]. To show that VI-VS is not tied to this choice, however, we also showed that VI-VS also produced calibrated p-values using an alternative generative model for single-cell RNA-seq data (Additional file 1: Table 3.1). Other types of features may require the use of a generative model that better approximates the data generating process. For instance, chromatin accessibility data may require the use of a generative model that properly account for ATAC-seq sparsity [28, 171] to ensure that VI-VS p-values remain valid. As illustrated in this work, more expressive importance scores, on the other hand, do not affect calibration but can improve power.

Technical data variations, such as differences in sample preparation and sequencing technologies, pose significant challenges for large-scale multiomic analysis [20, 172]. VI-VS effectively addresses this issue by conditioning on these nuisance factors. In single-cell studies, nonparametric tests, particularly the Conditional Randomization Test (CRT), have demonstrated the ability to produce calibrated significance scores in the presence of technical factors [173]. The generative models used by VI-VS are capable of capturing multiple nonlinear technical effects [53], enabling robust discoveries even in complex settings [174]. Therefore, we propose VI-VS as a general framework to produce calibrated significance scores of conditional associations in complex settings via integration of large datasets across multiple batches.

3.5 Supplementary Information

Supplementary Information on the experiments

Models

VI-VS We trained both scVI and the model associated with the importance scores until convergence using early stopping criteria on the held-out validation data.

OLS and marginal tests We computed OLS and marginal tests from log-CPM normalized counts. Tests relative to OLS and to the marginal baseline were fitted using the *statsmodels* python package.

Multiplicity control P-values obtained with VI-VS OLS, and the marginal test were adjusted using the Benjamini-Hochberg procedure [37].

Semi-synthetic data

We consider a PBMC scRNA-seq dataset consisting of $N = 6,855$ cells and $G = 500$ genes. We construct 5-dimensional synthetic surface protein measurements using the following scheme. Let $p \leq 5$ be some protein. We first construct the subset of conditionally dependent genes S using Binomial sampling, with sparsity rate $s = 0.3$. Mean measurements for protein p are obtained as

$$\mu_p = A_p \tilde{x}_S + \epsilon_p, \quad (3.9)$$

$A_p \in \mathbb{R}^{N \times s}$ corresponds to the ground-truth regressors, constructed as $A_p = \beta B$, where B denotes the $N \times s$ matrix of ones and β is the signal strength, parameterized as

$$\beta = \beta_0 \sqrt{\frac{2 \log G}{N}}. \quad (3.10)$$

Perturb-seq experiment

Data preprocessing We restricted the analysis to ten targeted genes with the highest number of single-guide RNA (sgRNA) perturbations. We identified true positive associations using a t-test on log-normalized counts comparing gene expression between cells without guide RNA, and cells with guides targeting the relevant gene. In particular, true positives interactions were defined as interactions for which multiplicity-adjusted p-values were below 0.1. We also restricted the analysis to 1000 pseudo-genes using the following procedure. We first clustered the genes using k-means based on their empirical correlation matrix, computed on log-CPM counts. For each obtained cluster, we summed the expression of the genes in the cluster, and used the resulting quantity as a pseudo-gene. A pseudo-gene was considered as a true positive if it contained at least one gene that was significantly differentially expressed between cells with and without guide RNA.

CITE-seq experiment

Data preprocessing The original data [145] contained gene expression and surface protein counts for eight donors at several time points, corresponding to both pre-vaccination and post-vaccination. We focused our analysis on the pre-vaccination time points, resulting in a dataset containing a total of 50,000 cells and 2,000 genes.

These genes were selected based on a procedure hoping to retain as much information about cells’ biological states as possible. To do so, we selected genes based on their joint variability with other genes instead of their marginal variability. To do so, we first clustered the genes using k-means based on their empirical correlation matrix, computed on log-CPM counts. In each cluster, we selected the gene with the highest highly variable-gene score, as predicted by Seurat [19]. The surface protein measurements y were centered and scaled to have zero mean and unit variance.

Nanostring experiment

Data segmentation and annotation Baysor [155] was used to segment the tissue into individual cells. We downloaded the lung FFPE Nanostring CosMX data. We used the position file of Lung 5 replicate 2 and split this into separate files for each FOV. Inside each FOV, we used the provided segmentation from Nanostring as a prior for Baysor segmentation and chose 0.1 as the prior segmentation weight. In short, we multiplied the z-position of each molecule by 10 to adjust for anisotropic distances. These results were concatenated across all FOV and scVI [22] was used to integrate these field of views. Leiden clustering was performed after neighbor calculation in latent space and cell-types were annotated based on known marker genes. We focused our analysis on T cells.

Identification of T cell markers in spatial analysis A number of genes not expressed in T cells might still be detected in the T cell analysis, due to contamination events, e.g., due to imperfect cellular segmentation. To circumvent this issue, we restrict the analysis of the spatial datasets to genes that are higher expressed in T cells compared to other cell-types. We identified these genes via one-vs-all differential expression analysis based on log-CPM counts, using t-tests to compare mean expression levels, and BH to correct for multiple testing. We then constructed a list of 245 genes, containing all genes with adjusted p-values < 0.05 , and with a log-fold change > 0.1 (upregulated in T cells).

Additional results

Semisynthetic Poisson experiment

We also considered a scenario where protein expressions suffer from Poisson instead of Gaussian noise, and where the relationship the protein expressions means are a linear combination of *squared* gene expressions. In such a scenario, an approach based on OLS does not produce valid p-values because of the model assumptions’

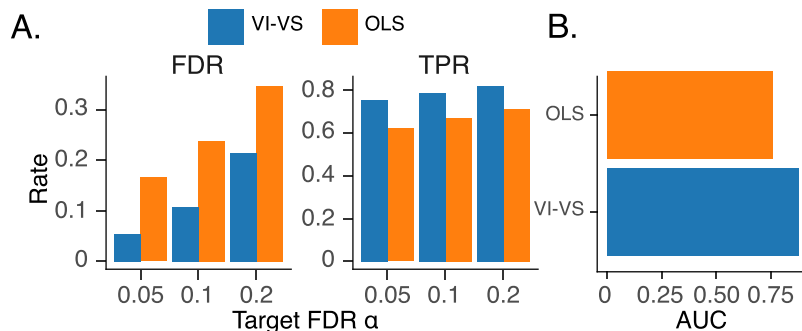


Figure 3.8: Semisynthetic experiment from a Poisson model. **A.** *Left*: FDR levels reached by VI-VS and OLS. *Right*: Power levels reached by VI-VS and OLS. **B.** Comparison of the area under the precision-recall curves of the different algorithms.

violations, while our approach still returns valid p-values (Figure 3.8). The power of our approach remains high, and has an higher area under the precision-recall curve to OLS.

Influence of key parameters on the semi-synthetic experiment

Contrary to OLS, the CRT-based approach provides consistently calibrated predictions (Figure 3.9). We however note that our approach displays intriguingly high FDR values for different sparsity levels. These scenarios correspond to regimes where the test detects a handful of discoveries.

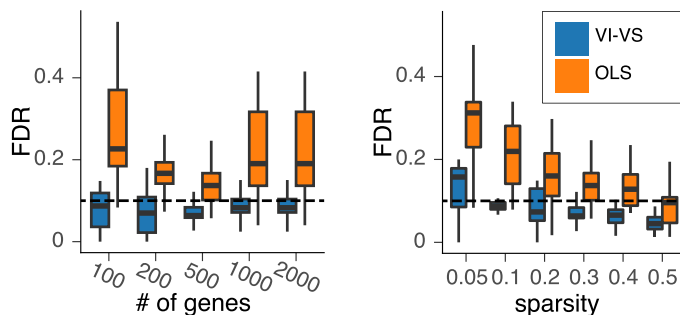


Figure 3.9: Influence of several parameters of importance in the semi-synthetic experiment. *Left*: Influence of the number of observed genes in the experiment. *Right*: Influence of the sparsity of the ground-truth parameter β .

Alternative generative model in the semi-synthetic experiment

We also considered an alternative generative model to scVI to produce synthetic gene expression counts. We considered a linear generative model, ldVAE, using a negative binomial likelihood, corresponding to [77]. The model was reimplemented in `jax`. Table 3.1 displays the obtained FDR and power for both scVI and linear generative model. These results show that the linear generative model controls the FDR, at a similar power level to scVI. This observation suggests that a simpler generative model suffices to produce valid p-values in this scenario. This experiment also shows that VI-VS is robust to the choice of generative model.

Table 3.1: Comparison of the FDR and power for the linear generative model and scVI on the semi-synthetic experiment for five random seeds, for a target FDR of $\alpha = 0.1$.

	scVI	ldVAE
FDR	0.06 ± 0.03	0.06 ± 0.03
Power	0.93 ± 0.03	0.94 ± 0.03

Effect of the development set size on the semi-synthetic experiment

We studied the effect of the development set size on the semi-synthetic experiment described in Supplement 3.5. We observed that for all development set sizes above 0.5, VI-VS provided calibrated p-values with high power, showing that VI-VS is robust to the train-test validation side. These development set sizes corresponded to splits for which the validation negative evidence lower-bound (ELBO), which is a proxy for the model’s fit, was low. These splits occur after the ELBO curve’s inflection point, where the model’s fit stabilizes and shows minimal improvement with additional data.

Consequently, we recommend, when required, the following heuristic for the choice of the development set size. When possible, a good heuristic is to fit VI-VS’s generative models for a range of development set sizes, and to pick a development set size after the ELBO curve’s inflection point.

A major benefit of VI-VS is that FDR calibration be assessed once the generative model has been fitted. For this purpose, a synthetic response variable can be generated using similar strategies as employed in Supplement 3.5, and calibration can be assessed using the synthetic response variable before considering the real response variable.

Absence of FDR control on such an experiment is a sign that the generative model does not properly capture features, and that further hyperparameter tuning of the

generative model (including the development size) might be necessary to ensure a proper fit of the generative model to the data.

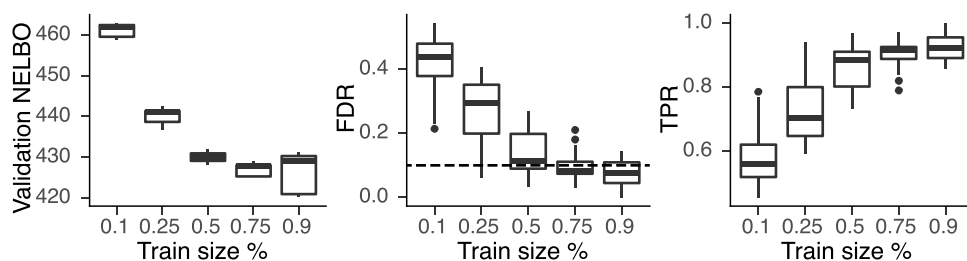


Figure 3.10: Effect of the development set size of VI-VS on the semi-synthetic experiment based on five random seeds for each size. *Left*: Validation negative evidence lower-bound (lower is better). *Middle* and *Right*: FDR and TPR for VI-VS decisions for a target FDR at level $\alpha = 0.1$.

Perturb-seq dataset

Figure 3.11 shows additional metrics for the decision obtained by the different algorithms on the perturb-seq dataset.

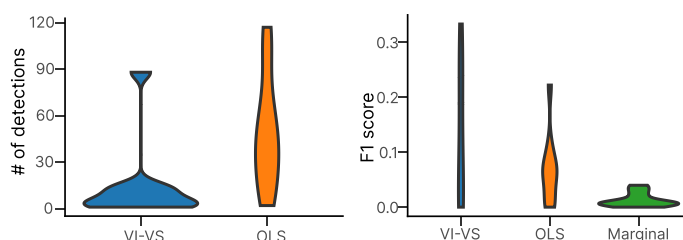


Figure 3.11: Additional metrics for the perturb-seq dataset, for the same decisions as in the main text. *Left*: Number of detections produced by the different algorithms. Marginal, similar to the rest of the experiments, detected a much higher number of hits (not shown). *Right*: F1-scores for the different algorithms.

Additional CITE-seq experiments

Figure 3.12 shows that the proteins for which VI-VS detects conditional associations have a higher percentage of cells expressing the protein.

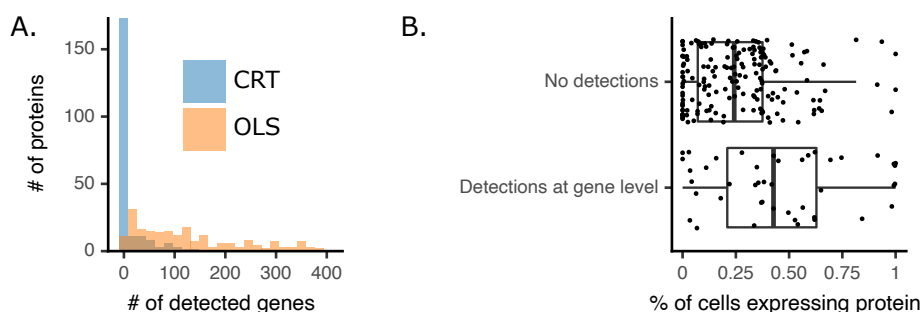


Figure 3.12: **A.** Comparison of the number of detections made by VI-VS and OLS across all proteins. **B.** Comparison of the percentage of cells expected to express the proteins for proteins for which (i). VI-VS predicts associated genes, and (ii). VI-VS does not predict associated genes. We predicted the percentage of cells expressing the proteins using TOTALVI, defining that a cell expressed a protein when the posterior probability observed protein counts comes from the foreground mode was above 0.95. Differences between the two groups are statistically significant (p -value $< 10^{-4}$ under a Kolmogorov–Smirnov test). In both these experiments, a gene-protein association is considered significant when the adjusted p -value is below 0.1.

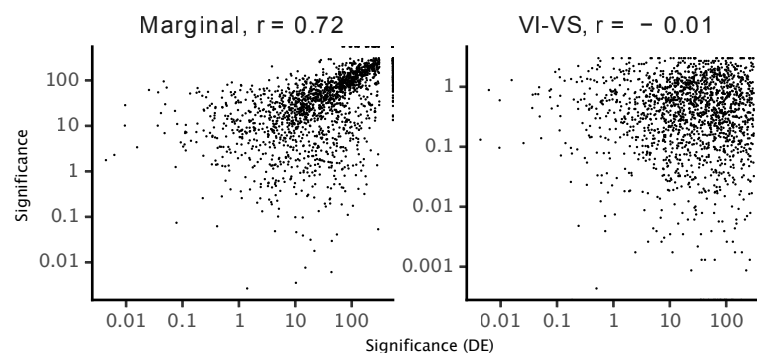


Figure 3.13: Comparison of significance scores for a Wilcoxon test between CD4+ and CD4- cells (x -axis) gene-protein significance scores (y -axis) for both a marginal test (*left*) and VI-VS (*right*). Significance scores were obtained as negative log p -values for the various tests. Correlation scores were computed using Spearman's rank correlation.

Comparison of the number of significant hits for the spatial DE analysis

Table 3.2 compares the number of significant hits for all considered methods for the spatial DE analysis.

	At cluster level	At gene level
VIVS	40	5
t-test	$\emptyset$	216
Wilcoxon	$\emptyset$	218

Table 3.2: Comparison of the number of significant hits for the spatial DE analysis for VI-VS as well as common DE approaches.

Theoretical background and guarantees

False Discovery Rate control

FDR control aims at ensuring that the rate of false positive detections made in multiple hypothesis testing scenarios is controlled in expectation. Formally, the FDR characterizes the expected proportion of erroneous discoveries:

$$FDR := \mathbb{E} \left[\frac{|S \cap N|}{\max(1, |S|)} \right], \quad (3.11)$$

where S denotes the set of detected associations, and N the set of non-existing associations in the data.

Proof of Proposition 1

Proposition 1 is a consequence of the following theorem,

Theorem 1 (CRT gives valid inference, from Lemma 4.1 in [39]). *Let $\tilde{x}_g$ denote some synthetic sample for feature g from some proposal distribution (a distribution used to generate synthetic samples). Suppose under $\mathcal{H}_{0,g}$, $\tilde{x}_g$ satisfy the following property:*

$$(\tilde{x}_g, x_1, \dots, x_g, \dots, x_G, y) \stackrel{d}{=} (x_g, x_1, \dots, \tilde{x}_g, \dots, x_G, y) \mid s, \quad (3.12)$$

for $k = 1, \dots, K$. Above, the notation $\stackrel{d}{=}$ denotes equality in distribution conditioned to s . If $\tilde{x}_g^{(1)}, \dots, \tilde{x}_g^{(K)}$ are K i.i.d. draws from this proposal distribution, then,

$$p_g := \frac{1}{K+1} \left(1 + \sum_{k=1}^K \mathbb{I}(T(\tilde{\mathbf{x}}^{(k)}, y) \leq T(\mathbf{x}, y)) \right),$$

where $\tilde{x}^{(k)} = [x_1, \dots, \tilde{x}_g^k, \dots, x_G]$, $k \leq K$, is a valid p -value for $\mathcal{H}_{0,g}$ (i.e., for the conditional independence test).

Property (3.12) ensures that the distribution of the data remains the same even when the true and synthetic samples are swapped. Equation 3.12 is a pairwise exchangeability property that ordered sets $(\tilde{x}_g, x_1, \dots, x_g, x_G, y)$ and its swapped counterpart $(x_g, x_1, \dots, \tilde{x}_g, x_G, y)$ have the same distribution. In particular, this property is a relaxation of the exchangeability property required to construct knockoffs. We refer to [39] for a more detailed discussion and interpretation of this property.

Proof. Assume that the distribution of synthetic samples $\tilde{x}_g$ satisfy Property (3.12) under the null. Then, conditional on s , under the null we have

$$T([x_1, \dots, x_g, \dots, x_G], y) \stackrel{d}{=} T([x_1, \dots, \tilde{x}_g, \dots, x_G], y).$$

Since $\tilde{x}_g^{(1)}, \dots, \tilde{x}_g^{(K)}$ are i.i.d draws from the conditional distribution and x_g is an independent sample from the same conditional distribution, we conclude

$$\frac{1}{K+1} \left(1 + \sum_{k=1}^K \mathbb{I}(T(\tilde{\mathbf{x}}^{(k)}, y) \leq T(\mathbf{x}, y)) \right) \sim \text{Uniform} \left(\frac{1}{K+1}, \frac{2}{K+1}, \dots, 1 \right). \quad (3.13)$$

□

We now use this result to prove Proposition 1, and in particular, that the two described sampling schemes verify the exchangeability property required to apply Theorem 1.

Proof. Proposition 1 states that two sampling schemes are valid for the CRT. The first sampling scheme corresponds to sampling from the distribution given by $\int p(x_g | z, s) p(z | x_{-g}, s) dz$. From our assumptions on the generative model, we have that

$$p(x_g | x_{-g}, s) = \int p(x_g | z, s) p(z | x_{-g}, s) dz.$$

Hence, (3.12) is satisfied. (See [39] for a more detailed proof.)

We now consider the second sampling scheme, which is the case presented in Algorithm 1. We let $\bar{z} \sim p(z | x, s)$ be a latent variable sampled from the posterior distribution. Let $x := [x_1, \dots, x_g, \dots, x_G]$ denote the observed gene expression profile. For clarity in the following equations, the subscript notation is used to specify which

random variable is being considered. For example, $p_{x_g|z,s}$ represents the density of random variable x_g given z and s .

Under the null, we have the following joint distribution

$$\begin{aligned} p_{\tilde{x}_g, x, y, \bar{z}|s}(\tilde{x}_g, x, y, \bar{z} | s) &= p_{x_g|z,s}(\tilde{x}_g | \bar{z}, s) p_{x|s}(x | s) p_{z|x,s}(\bar{z} | x, s) p(y | x_{-g}, s) \\ &= p_{x_g|z,s}(\tilde{x}_g | \bar{z}, s) p_{x_g|z,s}(x_g | \bar{z}, s) p_{x_{-g}|z,s}(x_{-g} | \bar{z}, s) p_z(\bar{z}) p(y | x_{-g}, s) \end{aligned} \quad (3.14)$$

These equalities follow from the assumed factorization of the generative model and from Bayes' rule. Notice the last expression is symmetric in the arguments x_g and $\tilde{x}_g$. We conclude that x_g and $\tilde{x}_g$ have the same distribution conditional on s and $\bar{Z} = \bar{z}$. We can thus apply Theorem 1 with conditioning on $(s, \bar{z})$. $\square$

Observed that exchangeability property does not hold unconditional to a fixed value of z . This can be seen in Equation 3.14, since $p(z | [x_1, \dots, \tilde{x}_g, \dots, x_G])$ and $p(z | [x_1, \dots, x_g, \dots, x_G])$ are not equal in general. This justifies why z must be fixed when sampling $\tilde{x}_g$, and cannot be sampled from the posterior distribution at every MC trial.

Supplementary Information on VI-VS

Feature-level and group-level conditional independence as a function of feature correlations

We considered a simple simulation to illustrate how VI-VS behaves as the correlation between features increases. To do so, we considered the data from Supplement 3.5. Starting from a random gene g , we constructed a synthetic gene expression g' that correlated with g ,

$$X_{ng'} = X_{ng} + \epsilon_n,$$

where n denotes the cell index. Here $\epsilon_1, \dots, \epsilon_N$ are i.i.d. Poisson draws from a Poisson distribution with parameter λ . As λ decreases, the correlation between g and g' increases. The response Y was generated using the same procedure as in Supplement 3.5, such that g was conditionally dependent with Y . Figure 3.14 displays the obtained pvalues for the conditional association between X_g and Y as a function of λ . We observed that in a regime where the correlation between X_g and X'_g is high, it was not possible to detect the conditional association between X_g and Y . At a coarser level, however, the conditional independence test was able to detect

the conditional association. These observations highlight a potential limitation of feature-level conditional independence tests, that the multiresolution component of VI-VS is able to overcome.

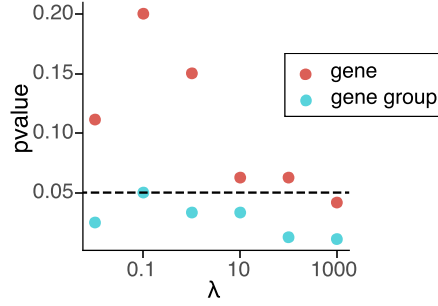


Figure 3.14: Obtained pvalues for the conditional association between X_g and Y as a function of λ . The correlation between X_g and X'_g increases as λ decreases.

Multiresolution analysis

In this section, we describe how VI-VS performs multiresolution analysis.

Feature clustering. The first step of this procedure consists in clustering features. While any clustering procedure can be used, we rely on a custom hierarchical clustering procedure when features are genes. After fitting scVI on all genes, we compute cell and gene specific denoised gene expression levels from the fitted model. We then compute the empirical correlation matrix between genes. We then used complete-linkage clustering to cluster genes at arbitrary resolution.

Multiresolution testing. Testing for conditional independence at the cluster level follows the same procedure as in the main text. In particular, let A denotes a set of genes for which we want to test

$$\mathcal{H}_{0,A}^M : x_A \perp\!\!\!\perp y \mid x_{A^C}.$$

We first randomize the expression for all genes in A , obtaining expressions $\tilde{X}_A = \{\tilde{X}_g, g \in A\}$, where $\tilde{X}_g$ is obtained from Equation 3.6. We then construct the randomized expression profile for all genes,

$$X' = \begin{cases} X & \text{if } g \notin A \\ \tilde{X}_g & \text{if } g \in A \end{cases} \quad (3.15)$$

We then compute the randomized test statistic as $T(X', Y, S)$. We then repeat this procedure K times, obtaining K randomized test statistics $T^{(1)}, \dots, T^{(K)}$, which we compare with the actual test statistic to construct the p-value for the conditional null.

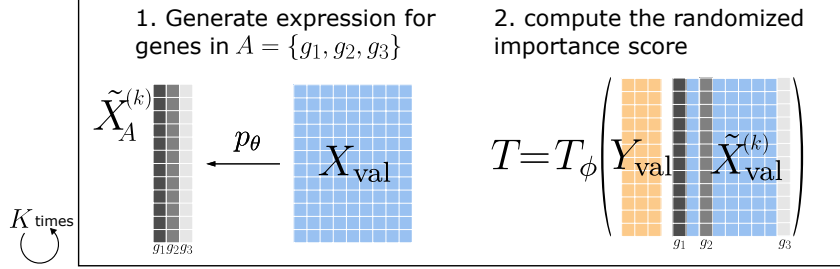


Figure 3.15: Illustration of the randomization procedure when testing at the gene group level. We rely on the generative model to produce synthetic gene expression profiles for all genes in the group A . Then, the randomized values are replaced in the original gene expression matrix to compute the randomized expression scores. This procedure is repeated K times to obtain the p-value for the conditional null.

Computational considerations. VI-VS relies on an efficient implementation that allows one to compute p-values across multiple resolutions in parallel. In particular, we use the same synthetic samples across all resolutions.

Implementation of VI-VS

We implemented VI-VS in python using Jax as backend. In particular, we relied on Algorithm 2 to compute the p-values, allowing to efficiently compute p-values using GPU acceleration. Our implementation relies on just-in-time compilation and parallelized computation to achieve high performance. This choice provides a two-fold improvement over an implementation relying on Pytorch, that benefits from a similar GPU acceleration but does not support just-in-time compilation nor parallelized computation.

Generalization to multidimensional Y

While the main text of this work assumes that Y is unidimensional, we here outline how to handle cases where the measurement is D -dimensional. To do so, we consider a multivariate regression task to fit the importance score. We train its parameters

Algorithm 2 Minibatched VI-VS

Require: Collection of B minibatches $\mathcal{D} = \{(X', Y', S')_{b \leq B}\}$, data statistic $T(X, Y, S)$, Sample budget K , trained generative model $p_\theta(X)$.

Compute $T_{\text{obs}} := T(\mathbf{X}, Y, S)$

for gene $g \leq G$ **do**

$T^k \leftarrow 0, \forall k \leq K$

for cell minibatch X', Y', S' **do**

for sample $k \leq K$ **do**

 Use p_θ to sample $\tilde{X}_g'^k$

$\tilde{\mathbf{X}}'^k \leftarrow [X'_1 \dots X'_{g-1}, \tilde{X}_g'^k, X'_{g+1}, \dots X'_G]^T$

$T_k \leftarrow T_k + T(\tilde{\mathbf{X}}'^k, Y', S')$

end for

end for

$p_g = \frac{1}{K+1} \left(1 + \sum_{k=1}^K \mathbb{I} \left(T(\tilde{\mathbf{X}}^k, Y, S) \leq T(\mathbf{X}, Y, S) \right) \right)$

end for

Return p-values $\{p_1, \dots, p_G\}$

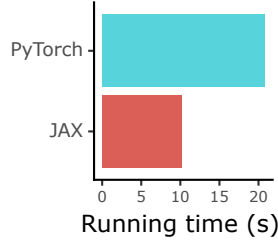


Figure 3.16: Running time comparisons. Execution time of VI-VS on a single GPU (RTX 3090) to compute p-values on a dataset containing 6,000 observations and 500 genes for 100 MC trials.

via optimization of the data likelihood, which writes for a single observation as $\mathcal{L}_\phi(y, x, s) := \sum_{d=1}^D \log p_\phi(y_d | x, s)$, assuming conditional independence of the features of y .

Two scenarios can appear when performing conditional independence testing with VI-VS in a multidimensional setup. First, it can be desirable to retrieve conditional dependencies specific to each feature of y ,

$$\mathcal{H}_{0,g}^d : y_d \perp\!\!\!\perp x_g \mid s. \quad (3.16)$$

In this case, we use $T_\phi^d(X, Y, S) := \frac{1}{N} \sum_{n=1}^N -\log p_\phi(y_d^n \mid x^n, s^n)$ as importance scores to test $\mathcal{H}_{0,g}^d$.

It can also be the case that we would like to test

$$\mathcal{H}_{0,g} : y \perp\!\!\!\perp x_g \mid s, \quad (3.17)$$

where in this case y is multidimensional. In this scenario, we use the full likelihood of the predictive model p_ϕ as importance scores.

Using VI-VS as a way to calibrate an existing feature selection method

VI-VS can be used to calibrate some existing feature selection methods that would not provide significance scores. Methods which rely on a regression fit to the data to ultimately score interactions between features and the response can straightforwardly be calibrated with VI-VS. Such methods include, for instance, regularized linear regression models for which significance scores are not easily available [130, 132], or ensembling methods [125, 126], that are popular feature selection tools employed for GRN inference.

For such methods, we first fit the regression model f on the development split, and define the importance score for the CRT as the mean prediction error of the model:

$$T(X, Y) := \frac{1}{N} \sum_{n=1}^N (y^n - f(x^n))^2 \quad (3.18)$$

Chapter 4

Multi-Resolution Variational Inference

4.1 Introduction

Over the past two decades, the use of functional genomics in large-scale, many-sample studies has been instrumental in advancing our understanding of how clinical, genetic, and environmental properties are manifested at the cellular and molecular levels [175, 176]. These studies now benefit from a potentially transformative increase in quality and resolution thanks to the maturation of large-scale single-cell genomics, which provides access to detailed information about the cellular and molecular composition of hundreds of samples [10, 15, 16, 40, 177–179]. Realizing the potential of large-scale single-cell genomics, however, requires rethinking the analysis strategy. While early on most studies relied on small numbers of samples and focused on variation between cells, the emergence of large-scale single-cell genomics now opens the way for a more in-depth understanding of the variation between samples.

There are at least two fundamental tasks in such a sample-level analysis. The first, which we refer to as *exploratory* analysis, is to divide the samples into groups based on their respective cellular and molecular properties. This idea of *de novo* grouping has seen powerful applications in clinical studies that use functional genomics to enable more precise prognoses and treatment planning [180, 181]. As a prominent example, pan-cancer analysis with functional genomics revealed that, in surprisingly many cases, cancer patients are more effectively classified using their molecular data rather than histopathology [42]. The second task is to conduct *comparative* analysis, i.e., identify cellular and molecular features that differ between pre-defined groups of samples (e.g., cases vs controls). In bulk-level studies, this was usually done in the

form of differential expression (DE) for detecting gene expression programs that are associated with conditions of interest [33]. The advent of single-cell genomics also popularized differential abundance (DA) as another form of comparison, one which seeks to discover cell states that are disproportionately abundant in a certain group of samples [45].

Current approaches for these two closely related problems suffer from limitations that preclude them from taking full advantage of the resolution afforded by single-cell genomics. Starting with exploratory analysis, a common approach for quantifying the distances between samples is to first organize the cells into groups (representing types or states) and then evaluate the differences in the frequency of each group [16, 40, 182–184]. This approach, however, may oversimplify the task by reducing the rich information we have about each sample. Furthermore, it hinges on the effective clustering of the cells (so as to represent distinct cell states), which is often complicated by the need to set an ample level of resolution, distinguish between closely related states, and harmonize different samples or datasets. Finally, this approach can miss critical effects that may only manifest in particular subsets of cells (As we later demonstrate, using cohorts of IBD and COVID patients). Similar issues also emerge in comparative analyses. Most of the current applications of DE and DA rely on *a priori* clustering of cells. It is possible, however, that DE programs span few or parts of the *a priori* defined cell subsets, thus less likely to be detected. Similarly, differentially abundant sub-populations may not clearly correspond to any annotated subset, which again limits the ability to detect them. Finally, even with access to high-quality annotation of the cells, it may still be that comparative analyses of different partitions (e.g., comparing between sexes or between age groups) are best reflected by different cell clustering schemes [41].

To mitigate these issues, a recent line of work focused on quantifying DE or DA without relying on predefined cell clusters [185–187]. These methods typically embed cells into a low-dimensional space and then consider small neighborhoods in that space to identify "local" DE or DA effects. A remaining caveat of this approach, however, is that it does not account for the uncertainty that embeddings may have (e.g., as inferred with variational autoencoders (VAEs) [22]), which can be substantial [156]. Another line of work uses VAEs to learn the effect of sample covariates on the latent embedding of cells [27, 188, 189]. The primary limitations of this approach are that it assumes the effects they evaluate are constant, meaning they are identical for all cells irrespective of their state, and that they do not account for the uncertainty in estimating these effects.

To address these challenges, we introduce MrVI (Multi-resolution Variational Inference), a probabilistic framework for large-scale (multi-sample) single-cell genomics. For exploratory analysis, MrVI identifies sample groups without requiring *a priori*

clustering of the cells. Instead, it allows for different sample groupings to be conferred by different subsets of cells that are detected automatically. For comparative analysis, MrVI enables both DE and DA in an annotation-free manner and at high resolution while accounting for uncertainty and controlling for undesired covariates, such as the experimental batch. The notion at the basis of MrVI is that of counterfactual analysis, which aims to infer what would the gene expression profile of a cell be had it come from a certain sample. This approach provides a principled methodology for estimating the effects of sample-level covariates on gene expression at the level of an individual cell. It relies on a hierarchical deep generative model architecture powered by modern techniques in deep learning, like cross-attention, to model the effects of sample covariates while at the same time providing state-of-the-art performance in the quality of sample integration. On the software side, MrVI leverages the optimization procedures included in *scvi-tools* [53], allowing it to scale to multi-sample studies with millions of cells.

In the following, we demonstrate that MrVI compares favorably to common approaches for integration, exploratory, and comparative analyses and then showcase its utility in several multi-sample studies. In a PBMC dataset from a COVID-19 study, MrVI identifies a monocyte-specific response to the disease that cannot be directly identified by more naive approaches. In a dataset of drug perturbation screens, MrVI reveals both expected and non-trivial relationships between the assayed compounds. Finally, using MrVI to study a cohort of patients with IBD, we find a previously unappreciated subset of pericytes with strong transcriptional changes in patients with stenosis.

4.2 Results

Multi-resolution variational inference

MrVI is a hierarchical Bayesian model for integrative, exploratory, and comparative analysis of single-cell RNA-sequencing data from multiple samples (e.g., corresponding to human subjects) or experimental conditions (e.g., perturbations in a screen; **Figure 4.1a**). The model utilizes two levels of hierarchy in order to distinguish between two types of sample-level covariates. The first covariate type captures properties we wish to study in either exploratory or comparative settings - we refer to these as *target covariates*. Typically, an identifier for each sample (e.g., human donor ID or experimental perturbation) is a natural choice for the target covariate to be provided as input to MrVI since it is entirely nested in other sample-level target attributes (e.g., treatment type), thus enabling their analysis. The second covariate

type is considered "nuisance", namely confounders we wish to exclude. In most cases, this will correspond to technical factors such as the sample processing site, library preparation technology, or the study-of-origin in cross-study analyses.

To formalize this, in MrVI, each cell n is associated with two low-dimensional latent variables, u_n and z_n (**Figure 4.1b**). The first variable, u_n is designed to capture the variation between cell states while being disentangled from sample covariates. The second variable, z_n , reflects the variation between cell states, in addition to the variation induced by target covariates, while still remaining unaffected by the nuisance covariates. Finally, we model the observed gene expression, x_n , as samples from negative binomial distributions whose parameters are predicted by decoding z_n conditioned on nuisance covariates.

Extending our previous work on scVI, MrVI employs a mixture of Gaussians as a prior for u_n instead of a uni-modal Gaussian. We demonstrate that this more versatile prior provides state-of-the-art performance in the integration of large datasets and in facilitating annotations of cell types and states. The distribution of z_n is learned as a function of the respective u_n and the sample ID, s_n (**Methods**). We used neural networks for all mapping functions in the model. The parameters characterizing these functions are learned through maximization of the evidence lower bound (**Methods**) [23].

The trained model can be used to perform both types of analyses - exploratory (de novo grouping of samples) and comparative (evaluating the effects of target covariates) at a single-cell resolution. For exploratory analysis, MrVI computes a sample-by-sample distance matrix, or sample distance matrix in short, for each cell n by evaluating how the sample-of-origin (s_n) affects the representation of this cell in z -space (**Figure 4.1c**). To this end, for each cell n , we compute $p(z_n | u_n, s')$, its hypothetical state had it originated from sample $s' \neq s_n$. We then define the distance between each pair of samples on cell n as the Euclidean distance between their respective hypothetical states. Then, hierarchical clustering can be used over the sample distance matrices for each cell to highlight the target covariates most likely to explain the major axes of sample-level variation. As we show next, this analysis helps capture, in an annotation-free manner, cellular populations that are influenced distinctly by target covariates (e.g., disease or tissue-of-origin).

In comparative analysis, MrVI can identify both DE and DA at a single-cell resolution (**Figure 4.1d**) using counterfactuals. Consider the case of differential expression between two sets of samples S_1 , S_2 as an illustrative example. To evaluate the group-level effects in cell n , we evaluate the extent to which the expectation of $p(z_n | u_n, s')$ depends on whether s' is in S_1 or S_2 using a linear model. We then use the decoder network (i.e., mapping from z to x) to detect which genes are affected and evaluate their effect size (fold change). Meanwhile, for local differential abundance,

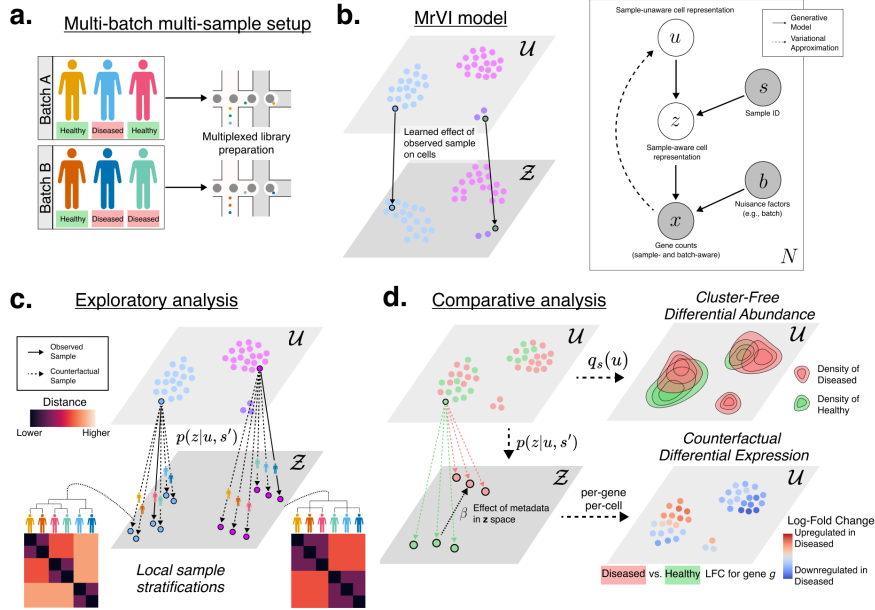


Figure 4.1: Overview of MrVI. **a.** We consider multi-batch, multi-sample experimental designs. In the canonical case, we gather single-cell measurements from several samples, which are collected across several batches. In this case, the relevant nuisance covariate is the batch. **b.** (*Left*) MrVI model illustration and (*Right*) graphical model plate diagram. MrVI relies on two cell representations u and z . A sample-unaware cell representation u captures shared type information (colored by cell type in the diagram). From this quantity and the sample-of-origin of the cell, we construct a sample-aware representation z of the cell. Last, we model gene expression as a function of this latent variable and of observed nuisance factors. Each point in the diagram corresponds to an individual cell. **c-d.** Use cases of MrVI for exploratory and comparative analyses. **c** Exploratory analysis: MrVI computes local sample stratifications. MrVI can compute counterfactual representations characterizing what would have been the representation of a cell had it originated from a different sample. By computing the distances between counterfactual representations of all samples, MrVI can identify sample-level effects on cell states. **d** Comparative analysis: MrVI quantifies differences in abundance across cell states (*d top right*), and identifies sample metadata effects on gene expressions (*d bottom*) Both the sample stratification and differential expression procedures use counterfactual z representations to compare local sample effects. The differential abundance procedure involves an approximation of the posterior density for each sample in the u latent space.

we estimate the posteriors $p(u_n | s')$ and compare the aggregate values of samples s' in S_1 vs. S_2 . An in-depth description of MrVI and its post-training analysis procedures are provided in the **Methods** section.

Sample effect characterization in a semi-synthetic experiment

In our first test case, we used a semi-synthetic dataset to evaluate how accurately MrVI captures differences between samples (through exploratory and comparative analysis) when different cell subsets are influenced by different sample-level effects. To generate this, we used a published dataset of 68K peripheral blood mononuclear cells (PBMCs) [90] profiled with 10x, consisting of 3K highly variable genes and five main cell clusters, which we refer to as subsets A-E. We assigned each cell in this dataset to one of 32 synthetic study subjects. These study subjects are characterized by two distinct sample-level covariates. Our strategy for assigning cells to the simulated subjects varied between the different cell subsets so as to simulate different covariate effects. For subset A, the assignment of cells resulted in DE across categories of *covariate 1* in a way that reflects a hierarchical grouping of the samples. For subsets B and C, our cell assignment reflected DA across categories of *covariate 2* (**Figure 4.2a**). Cells in the remaining subsets were randomly assigned to samples and hence did not contain any DE or DA effects (**Methods**).

We applied MrVI, using the simulated subject identifiers as the modeled target covariate s_n and leaving the nuisance covariate b_n empty. The resulting u space clearly reflected the differences between the cell subsets (**Figure 4.2b**). In the z space, we observed distinct subject-specific effects in cells of subset A, while cells in the remaining clusters were mixed. This result aligned with our expectations, as only subset A contained DE effects. For exploratory analysis, we used the mapping from u to z to estimate sample distances for each cell (**Figure 4.2c**). In cell subset A, the sample distance matrix (averaged over all cells of the subset) produced a hierarchical structure similar to the simulated (ground truth) dendrogram. Consistent with our simulation strategy, MrVI estimated much smaller distances between samples when considering the other cell subsets, with no discernible structure. We compared this result to the standard approach for stratifying subjects using differences in cell-type frequencies (**Supplement 4.4**). Specifically, we sub-clustered each subset using cell embedding derived with either PCA or scVI. Then, separately for each subset, we estimated the distances between subjects using the respective sub-cluster proportions. Both standard compositional analyses were less effective in capturing the hierarchy of study subjects in cell subset A (**Figures 4.8a and c**) and were more likely to introduce non-negligible distances in subsets where no differences were expected (**Figure 4.8b**).

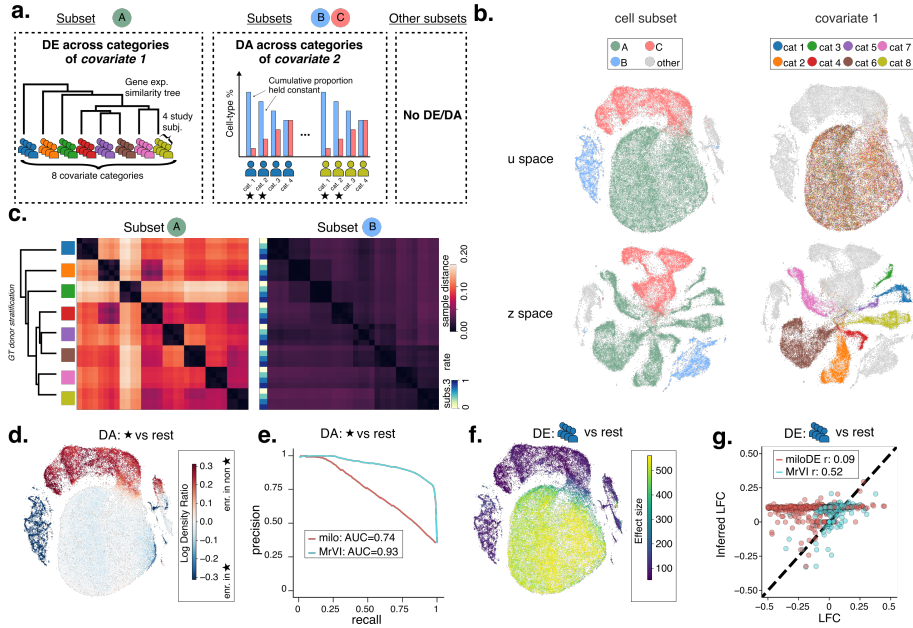


Figure 4.2: Semi-synthetic experiment. **a.** Experimental design. We generated a semi-synthetic dataset with 5 cell subsets and 32 study subjects, containing both differential expression and abundance effects. In subset A, cells show gene expression differences across subjects based on *covariate 1*, following a known hierarchy. In subsets B and C, cells show abundance differences based on *covariate 2* (4 categories). For each category, subset B cells were over-sampled and subset C cells under-sampled at fixed rates, keeping their sum constant. Stars mark categories with strong sampling effects. Other subsets show no DE or DA effects. **b.** MDEs of MrVI *u* and *z* latent spaces, colored by cell subset assignments (*Left*) and *covariate 1* category (*Right*). **c.** Cell-subset-specific distance matrices for stratified (*Left*) and non-stratified subpopulations (*Right*). **d.** and **e.** DA analysis using MrVI and Milo. We compared samples with similar vs. different proportions of subsets B and C (marked by stars in **a**). **d.** MDE of MrVI *u* space colored by log density ratios from Equation 4.4. **e.** Precision-recall curves (AUC, higher is better) for identifying DA cells based on *covariate 2*. True positives are cells from types B or C. **f.** and **g.** DE analysis using MrVI and miloDE. We compared DE in cells from one *covariate 2* category (blue) vs. others. **f.** MDE of *u* space colored by effect size (squared norm of β_n from Equation (4.3)). **g.** Comparison of LFCs from miloDE and MrVI vs. DESeq2 reference. MrVI LFCs, averaged across DE cell subset, show better correlation with reference than miloDE. Score details for Milo and MiloDE in e., f., and g. are in **Methods**.

To evaluate MrVI for DA, we partitioned the subjects into two groups according to *covariate 2* (presence or absence of a star in **Figure 4.2a**). We used the estimated posteriors $p(u|s)$ around each cell to evaluate the extent to which its state was over-represented in one group of study subjects versus the other. The resulting log ratios accurately reflected the DA effects that were simulated in cell subsets B and C (**Figure 4.2d**). Furthermore, the inferred ratios significantly diverged from zero only in subsets B and C (**Figure 4.8d**). We compared MrVI to Milo [185], a popular framework for DA. We observed that MrVI more accurately identified the DA effects and associated them with the correct cell subsets (**Figure 4.2e** and **Supplement 4.4**).

For DE analysis, we compared the subjects in one category of *covariate 1* (blue in **Figure 4.2a**) to all other subjects. In this comparison, only cell subset A was expected to contain DE effects. We used the estimated posteriors $p(z|u, s)$ around each cell to evaluate the extent to which its gene expression profile depends on the category of its sample-of-origin. The resulting effect sizes were inferred for each latent dimension in z using a linear model (see **Methods**; β_n in Equation 4.3). The squared norm of these effect sizes (aggregating all dimensions of z) was used as a measure of the overall effect of covariate 1 on gene expression in each cell.

We observed that these quantities reached much higher values in cells of subset A compared to the remaining subsets (**Figure 4.2f**). This indicates that MrVI can capture the particular groups of cells exhibiting DE effects. Next, we used the decoder function, $p(x|z, b)$, to evaluate each gene’s respective effect sizes (log-fold changes; LFC) in each cell belonging to subset A. We compared these to effect sizes obtained by a pseudo-bulk DE analysis of subset A (the latter representing an annotation-dependent analysis in the “perfect” scenario where the annotations completely align with the DE signal). We find that the two evaluations of effect sizes were highly correlated, with a substantial improvement over miloDE - a recent method for cluster-free DE analysis (**Figure 4.2g**).

These results demonstrate that MrVI can identify different sample groupings for different cell subsets without requiring an *a priori* annotation of cell states. Similarly, it is able to accurately retrieve shifts in cell state composition (DA) and gene expression (DE) and identify the respective cellular populations.

Analysis of variation among COVID-19 patients reveals Myeloid-specific stratification into clinical groups

We next employed MrVI to analyze 419k PBMCs obtained from a cohort of COVID-19 patients and healthy controls [16]. We used the sample identifier, corresponding

to unique study subjects, as our modeled target covariate s . As anticipated, the resulting u space is not affected by the sample-of-origin, and instead shows marked mixing between study subjects (**Figure 4.3a**). At the same time, the u space clearly stratified the cells into immune subsets in a manner consistent with their annotation in the original study. Considering a standard evaluation of integration performance in this case (using scIB-metrics [24]), we find that the MrVI u space embedding outperformed PCA and scVI in terms of mixing the samples while retaining their biological signal [24] (**Figure 4.9**). To effectively use this dataset, however, we would like our model not to remove the differences between samples but, in this case, to capture the effects of viral infection. While this was not readily achievable with existing integration methods, the two-level structure of MrVI allowed us to derive both a representation that is cell-type-centric (u) and one that is affected by the respective sample (z). Indeed, the z space showed clear sample-specific variation, separating COVID-19-positive patients from the control population inside each cell type (**Figure 4.3b**).

The two-level representation in MrVI enables an exploratory analysis through two fundamental questions: How do samples in this cohort stratify into groups? And - do they stratify differently when considering different immune populations? To address this, we used the counterfactual embeddings $p(z_n|u_n, s')$ to estimate a sample distance matrix for each cell n . We then clustered the cells according to these values, thus detecting groups of cells that confer similar groupings of the samples and are thus similarly affected by sample-level covariates (**Figure 4.3c**). This clustering strategy serves a different purpose than clustering cells based on their gene expression profiles. In particular, this approach aimed at identifying groups of cells of potentially different types similarly affected by samples. Using this analysis, we clustered the cells into three groups, the first containing T cells and NK cells, the second consisting primarily of monocytes along with a smaller population of dendritic cells, and the third containing B cells.

The resulting distance matrices, averaged across all cells in each respective group, provided a clear separation between the patient population and healthy controls, indicating that MrVI can identify clinically relevant groups (**Figure 4.3d, 4.10**). However, the distance matrix conferred by monocytes and dendritic cells highlighted an additional stratification of the study subjects. In this cell cluster, patients with COVID-19 were further stratified into two groups, corresponding to patient groups ① and ③ in **Figure 4.3d**. Patient group ① was enriched in individuals for whom the number of days since first symptoms (DSS) was low, while individuals ③ showed longer duration of symptoms (**Figure 4.11**; Mann-Whitney U test, $p < 0.05$). The association with monocyte activity and the time elapsed since infection has been established [190]. MrVI was able to identify this association without prior knowledge

of the DSS or any other information about the human subjects in this study.

To further interpret this data-driven stratification of COVID patients and its association with monocytes, we turned to DA and DE analyses. First, a DA analysis of the myeloid population, comparing the patient population to the healthy controls, showed a marked decrease in non-classical CD16+ monocytes and dendritic cells (**Figure 4.3e, 4.12a**) in patients. The comparison of the two groups of patients similarly showed a shift toward non-classical monocytes in the group with higher DSS (**Figures 4.3e, 4.12b**). These results are consistent with independent analysis of COVID-19 patients [190], which reported that CD14+ monocytes are highly pro-inflammatory and contribute to the cytokine release in early COVID, thereby contributing to disease symptoms.

Next, we applied our DE analysis to compare the two patient groups. Using MrVI counterfactuals we estimated the respective effect size (LFC) for each gene in every myeloid cell. We then clustered the genes based on their estimated LFC profiles (see **Methods**). This analysis uncovered three modules, each containing genes with a similar DE pattern implicating different subsets of myeloid cells (**Figures 4.3f, 4.13**). The first module, upregulated in the group of patients that had higher DSS, was enriched in genes that we also identified in myeloid cells of healthy individuals (compared to patients), again supporting the notion of a return to baseline in myeloid cells with long-standing infection. Specifically, we see a lower *CSF3R* expression in the recently-infected patients, which aligns with less mature monocytes that are released earlier from bone marrow during infection [191]. Similarly, our results indicate that early in the infection, the number of MHC-II-expressing monocytes declines but later returns to normal levels [190]. This accounts for the observed elevation in *LGALS2* and *HLA-DR2*, both of which are linked to MHC-II. The second module, over-expressed in patients with lower DSS, is enriched in interferon-related genes. Specifically, this module includes *GBP1* and *IFITM3*, which are interferon-response genes, and *IFI27*, which was reported as an early predictor of COVID-19 severity [192]. These results agree with a strong interferon signaling during early infection, especially in myeloid cells [190]. The third module, over-expressed by the higher DSS group, contained *TNF* and *NFKBIZ*. It has been demonstrated that TNF release is reduced in acute COVID, whereas *NFKBIZ* is drastically reduced in acute infection [193]. Our analysis suggests that both molecules are markers of acute infection more so than mortality.

Taken together, these results demonstrate that MrVI can simultaneously identify clinically meaningful groupings of study participants as well as the cell subsets that induce them. MrVI also provides the tools to interrogate these stratifications through cluster-free DE and DA analysis. Applied to a cohort of COVID patients, we recovered known compositional changes in the blood myeloid compartment (including a decrease

of dendritic cells and non-classical monocytes) as well as gradual gene expression changes in these cells over the course of the disease. MrVI was able to highlight the variation that ensues with DSS as a phenomenon that is more clearly associated with myeloid cells, rather than T or B cells. While the reason for this is currently unclear, monocytes have a relatively shorter half-life time in blood and are replaced by bone marrow-derived cells [194].

MrVI enables grouping and characterization of small molecules in screening assays

To demonstrate the flexibility of the notion of the target covariate in MrVI, we analyzed a chemical perturbation screen with a single cell RNA-seq readout generated with the sci-Plex assay [15]. The sci-RNA-seq3 dataset includes three cell lines and 188 small molecule drugs plus vehicle controls. Each small molecule was delivered at four different doses and the entire study was conducted with two biological replicates. The sci-Plex assay is designed such that each cell receives a single perturbation (or negative control vehicle) that can be identified in addition to its transcriptome. MrVI can serve several fundamental analyses for this type of study, namely integrating all replicates into a shared embedding, stratifying the screened compounds into groups with similar effects, and mapping out these effects at the level of gene expression and the composition of cell states.

To achieve this, we used the concatenation of the drug name and the dose level as the target covariate modeled by MrVI, resulting in 752 "samples" per cell line. Since the study was conducted with 96-well plates, with each plate containing one of the two biological replicates, we chose the plate identifier as our nuisance covariate. As found in the original study, many of the drug-dose combinations had minimal effect on transcription across the assayed cell lines [15]. For this reason, we applied a simple filter, only retaining drugs that had a minimal number of differentially expressed genes with at least one concentration and in at least one cell line (using a standard t-test between cells from each sample against the vehicle; see **Methods**). This resulted in a total of 368 perturbation samples (92 drugs with four concentrations each) that we used for training MrVI (**Methods**). In the following, we focus on the epithelial A549 lung adenocarcinoma line. We provide the results from the other two cell lines in the Supplement.

The resulting u space (**Figure 4.4a**) appears as a single cluster with no apparent sub-clusters specific to a given class of drugs, indicating a successful integration that reflects the drug-independent states of the cells. We rather observed that positioning in the u space carries information about the phase of the cell cycle. With respect

to the sample-affected (z) space (**Figure 4.4b**), we observed several sub-clusters of cells originating from distinct classes of drugs. In particular, populations of cells treated with HDAC inhibitors (expected to target epigenetic regulation) and trametinib (a block to MEK-mediated tyrosine kinase signaling) formed clear clusters, highlighting their distinct drug-induced shifts in gene expression. The distinction of HDAC inhibitors is concordant with the original study, in which the authors additionally identified acetyl-CoA deprivation as a common mechanism for this drug class, captured by drug-induced shifts in gene expression. The response to trametinib is also expected to greatly impact the Ras-driven A549 cells since the drug inhibits the downstream MEK pathway [15].

For a more quantitative comparison of the chemical perturbations, we turned to the sample distance matrices estimated by MrVI. We found that the cells in this cell-line-based assay are homogenous in terms of the distance matrices that they induce (**Figure 4.4c**) with no evident subclusters in PC space. This is in contrast to the other datasets analyzed here that explored primary cells with diverse cell types, featuring distinct distance matrices (**Figures 4.3,4.5**). Therefore, we operated on one sample distance matrix for the remainder of the analysis, taking the average over all cells.

To test whether the resulting distance matrix captures *a priori* known relationships between the samples, we formulated two performance metrics and compared MrVI to the two standard (composition-based) methods as before (**Figure 4.4d** and **Supplement 4.5**). First, we used the transcriptomic-based Connectivity Map resource [195], which provides a measure of similarity between drugs, and compared these similarities to the distances we estimated using MrVI for the maximum dose tested (10000 nM). Second, we evaluated the extent to which treatments with the same drug but at different concentrations tend to be more similar to each other than expected by chance. MrVI achieved better performance on both metrics - showing higher concordance with the Connectivity Map stratification and lower distances between treatments with the same compound. Notably, these metrics were also used for fine-tuning the hyperparameters of the MrVI model (here, the dimensions of u and z). This strategy reflects real-world cases where prior knowledge of the similarity between samples can be utilized for more effective modeling and downstream analysis (**Figure 4.15**).

To gain additional insight, we analyzed a hierarchical clustering of the sample distance matrix (**Figure 4.4e, 4.16**). We found that drug-dose combinations that had little to no effect in the A549 context mostly clustered together with the vehicle treatment (**Figure 4.17**). We observed that disproportionately many of the samples with low effect sizes also have low dosages, capturing expected dose-response relationships. The remaining samples that were most distinct from the vehicle sample

were organized into several clusters, each with a different effect on gene expression (**Figures 4.18, 4.19**). Specifically, clusters ①, ③, and ④ consist mostly of HDAC inhibitors. These three clusters stretch across a wide range of effect sizes that are correlated with the dosages, with cluster ① containing the samples with the highest dosage levels and the largest effects and cluster ③ consisting of samples with lower dosages and weaker effects (**Figure 4.18**). These groupings highlight the ability of MrVI to uncover dose-dependent effects on gene expression apparent across multiple drugs in the HDAC inhibitor class. Clusters ② and ⑧ corresponded to all doses of trametinib and YM155, respectively. In these cases, MrVI therefore suggests that the effects of these drugs were less dependent on the dose, at least when considering the range of concentrations tested.

While the clusters aligned with the expectation that drugs labeled to target the same pathway or different doses of the same drug should have similar effects, MrVI also uncovered relationships between drugs based on their effects on transcription that are supported by recent literature and the original sci-Plex study. For instance, cluster ⑥ includes rigosertib, labeled as a tyrosine kinase inhibitor but found to directly affect microtubule function [196], as well as epothilone A and patupilone, two drugs that interfere with the microtubule function. Moreover, MrVI revealed non-trivial similarities that were not captured in the original study. In cluster ⑤, two JAK2 inhibitors, fedratinib and TG101209, were grouped with JQ1, a drug labeled as a BRD inhibitor. Interestingly, recent work has shown that JQ1 inhibits the JAK-STAT signaling pathway in addition to being a BRD inhibitor, which supports the plausibility of this grouping [197].

Finally, we investigated the clusters by performing Gene Set Enrichment Analysis (GSEA;[198]) on the DE gene sets identified by MrVI (comparing each cluster of samples to the vehicle controls; **Figure 4.4f** and **Methods**). As a reference, we used the hallmark collection of MSigDB [143] that records sets of genes that contribute to major cellular processes. This analysis shed additional light on the effects of each cluster of drugs. Specifically, we found that clusters ①, ③, and ④ associate most strongly with a down-regulation in metabolic pathways, agreeing with the effect of HDAC inhibitors on carbon metabolism [15]. Furthermore, Cluster ⑥ was enriched in the p53 pathway, consistent with the categorization of its respective drugs as cell cycle regulators. Similarly, the effects of Cluster ② were enriched in genes downstream of KRAS signaling, consistent with its categorization as targeting tyrosine kinases. We provide a heatmap of the LFCs for the top DE genes across all clusters in the supplement (**Figure 4.20**). Based on this analysis, we highlight that MrVI not only provides an interpretable grouping of clusters but additionally helps highlight the genes underlying this grouping.

For the other two cell lines used in the sci-Plex experiment, the results of MrVI

were consistent with known biology (**Figures 4.21-4.30**). For instance, the hormone-receptor-positive breast cancer cell line, MCF-7, was the only cell line to exhibit strong effects in response to the hormone therapies, fulvestrant and toremifene citrate, as evident in both the z space and the sample distance matrix (**Figure 4.21**). In particular, fulvestrant, which had a strong effect at the highest two dose levels on MCF-7 [199], had little effect on other cell lines and did not appear in the top 20% of drug-dose combinations for the other two cell lines based on distance from the vehicle. The GSEA results also showed strong down-regulation of estrogen-response-related genes in response to many of the drugs. Similarly, in the Bcr-Abl positive cell line K562 [200], we observe a cluster of Bcr-Abl tyrosine kinase inhibitors (bosutinib, dasatinib, nilotinib) with a significant effect absent from the other cell lines (**Figure 4.22**).

Overall, MrVI recapitulated both expected and novel aspects of the effects of the screened drugs and the relationship between them. It did so through an end-to-end solution that coupled integration across experiments, estimation of the effects of individual molecules, stratification of the affecting molecules into biologically meaningful groups, and characterization of the effects of each group. MrVI uncovered non-trivial relationships between drug dosage and effect on cell lines, highlighting dosage-independent and -dependent effects. More broadly, this analysis demonstrates the utility of MrVI in powering screening studies (e.g., genetic or chemical) that leverage single-cell readouts for studying large numbers of perturbations.

Characterizing the role of stromal cells in stenosis in Crohn’s disease

To provide another example of the applicability of MrVI to human cohorts, we utilized a recent study of Crohn’s disease, conducted in 46 patients and 25 controls across 463k cells using single-cell RNA-sequencing [14]. In addition to gene expression, this dataset comes with metadata describing the anatomical location of sampling (in terms of tissue: colon or ileum, and in terms of the tissue layer: lamina propria or epithelial), the method of extraction (surgical or biopsy), and sample preparation detail (10X chemistry). It also includes information on the respective study participant, such as disease state and the presence of stenosis in the patient’s history (i.e., narrowing of the intestines due to inflammation). We used these metadata to evaluate the ability of MrVI to recognize meaningful patient sub-groups and to highlight cell populations that are affected by stenosis.

We trained MrVI to integrate all the samples in this dataset, using the sample identifier as the modeled target covariate and the combination of library preparation

protocol and tissue layers (lamina propria and mucosa) as the nuisance covariate. Comparing the resulting u space to the embedding obtained with scVI, we found that the default settings of MrVI yielded better mixing between the study subjects but had slightly lower performance in terms of distinguishing between cell states (using cell annotations assigned in the original study; **Figure 4.31**). Indeed, integration is challenging in this dataset due to significant differences in cell type composition in the colon and in the ileum, as well as between the mucosal and lamina propria layers. To increase the extent to which the u representation captures variation between cell states, we developed a variant of MrVI that makes use of cell type labels. Specifically, we introduce a cell-type-specific bias term for the mixture weight of the mixture of Gaussians prior used by MrVI, thus encouraging similar embedding for cells of the same state (**Methods**). Compared to scVI, this added consideration of cell type labels leads to an overall improved performance, considering mixing between samples and preserving cell type information (**Figure 4.31**).

We next used MrVI to explore how the different samples stratify and how these strata change between cell types. Using a coarse definition of cell types (**Figure 4.5a** and **Methods**), we again find that different types are associated with different patient groupings. For instance, considering a subset of immature Enterocytes (referred to herein as Enterocytes-Stem), the sample distance matrix clustered solely by their tissue-of-origin (colon or ileum) (**Figure 4.32**). We applied the DE function of MrVI on the Enterocytes-Stem subset to investigate the differences between these clusters of samples. In line with their biological functions, we find higher expression of *AQP8*, *CA2* and *CA8* in the colon, which all encode proteins that absorb water, and a higher expression of *FABP6* and *FABP2* in the ileum, which encode proteins that absorb fatty acids. Considering a population of mature Enterocytes provides a slightly different view, highlighting a specific cluster of eight patients that are distinguished from all other patients and controls. Using MrVI DE to compare mature Enterocytes in this cluster, which mainly contained colon samples, to other colon samples, we detected an up-regulation of mucins (*MUC1*, *MUC2*, *MUC12*), which is a well-described pattern in Crohn’s disease [201], and an up-regulation of *CXCL1* and *CXCL3*, which encode chemokines that attract neutrophils [202]. The expression of these chemokines was associated with stimulation of epithelial cells with *IL22* and *IL17A* that are key cytokines in Crohn’s disease. Furthermore, neutrophil infiltration is a key feature of inflamed gut regions [203]. We find here that, depending on the cell subset under consideration, the exploratory analysis of MrVI can reflect known differences between the tissues sampled (here ileum vs. colon) as well as reveal differences in a subpopulation of diseased individuals (non-inflamed vs. inflamed).

Next, we explored the use of MrVI for comparative analysis with respect to

a known covariate. We consider the distinction between the two most common complications of Crohn’s disease: stenosis (Vienna classification B2) and fistula or abscesses (penetrating; Vienna classification B3). These are evident in 11 and 7 of the subjects, respectively, while the remaining are healthy controls or patients without either complication (Vienna classification B1). Finding reliable biomarkers that can distinguish between patients experiencing the two types of complications is critical, as they may require different treatment strategies. Using our multivariate DE procedure, we evaluated the extent to which each individual cell is impacted by the presence of stenosis as well as by the inflammation status (inflamed vs. non-inflamed) while accounting for the effects of nuisance covariates such as biological sex and tissue location (**Methods**). We excluded surgical samples from this analysis due to the marked differences in cell type composition compared to biopsies (which are the source of most cells in the dataset; **Figure 4.5a** and **Methods**). We find that inflammation status had a marked effect on several cell lineages, with a strong effect on the stromal compartment. As expected, the presence of B2 disease had a more mild effect, mostly restricted to a few stromal subsets, with its highest impact in a small subset of pericytes (**Figures 4.5a, 4.33** and **4.34b**). Therefore, we focus the remainder of our analysis on stromal populations consisting of fibroblasts, pericytes, glia cells, and endothelial cells (**Figures 4.5b** and **4.35**).

We first compared B2 vs. B1 samples using DA analysis while controlling for inflammation status and other covariates (**Methods**). We find a decrease in the abundance of several endothelial populations in B2 samples (e.g., lymphatic and *LTC4S*+ endothelial cells) and an increase in fibroblast populations (e.g., *ADAMDEC1*+; **Figures 4.5b, c, h**). This result accords with the prevalence of microvascular rarefaction (i.e., loss of endothelial cells) in tissue fibrosis [204]. Using the same settings for DE, we find that in patients classified as B2, a subset of *HIGD1B*+*STEAP4*+ pericytes up-regulated *CDH11*, a biomarker of stenosis [205], as well as classical markers of tissue fibrosis (*ADAMDEC1* and *COL1A1*) and activation (*S100A6*, *MT2A*, and *JUN*) (**Figures 4.5d, e, f** and **4.34d**). While these genes are up-regulated in B2 disease by other stromal subsets, we find several genes that are affected uniquely in cells of the *HIGD1B*+*STEAP4*+ pericyte subset. This includes *LUM*, which was shown to be up-regulated after fibroblast stimulation in lung fibrosis and promotes fibrocyte differentiation [206] as well as *PDGFRB* and *TGFBI*, which are strongly up-regulated in lung fibrosis and whose pharmacological inhibition reduces fibrosis [186, 207]. Furthermore, *LUM* and *TGFBI* were reported to be up-regulated in the chronic phase of a mouse model of colitis [208], which is associated with marked intestinal fibrosis. Notably, while MrVI bases its DE estimations on its generative model, targeted analysis of the same molecules using the raw data shows consistent results (**Figure 4.5g**). Together, this analysis demonstrates the potential of MrVI

for delineating a population of cells that is associated with a disease phenotype and may facilitate a more nuanced discovery of markers for diagnosis and treatment.

MrVI additionally predicts marked up-regulation of *PDGFRB* by *CD36+* endothelial cells in B2 samples (**Figure 4.5e, f**). This is unexpected as *PDGFRB* is a common marker of pericytes and is not normally expressed by endothelial cells. We further characterized gene expression in the *CD36+* subset and found co-expression of endothelial markers (like *PLVAP*, *VAMP5*, *VWA1*) and pericyte markers (like *NOTCH3*, *RGS5* and *MYL9*) (**Figures 4.5h** and **4.34d**). In addition, we find up-regulation of markers of tissue fibrosis such as *COL1A1* and *TGFBI* in this subset (**Figure 4.34e**). Therefore, these results highlight a cell population with a mixed phenotype between the endothelial and pericyte lineages, which up-regulates markers of tissue fibrosis in B2 samples. This hints towards endothelial-to-mesenchymal transition in IBD and suggests the presence of a pericyte-like state in the gut endothelium of B2 disease. Endothelial-to-mesenchymal transition has been described for human IBD [209]. However, this phenomenon has not been explored in the original study of this cohort, nor, to date, studied with single-cell genomics [209].

In summary, MrVI uncovered previously unappreciated changes in cellular and molecular composition in patients with stenosis. It identified cells with strong changes in gene expression in stenosis, detected several genes associated with tissue fibrosis, and suggested endothelial-to-mesenchymal transition in stenosis.

4.3 Discussion

We introduced MrVI, a comprehensive solution for large-scale (multi-sample) single-cell RNA-seq studies. MrVI provides a unified probabilistic framework for several fundamental tasks, namely integration of samples from different sources, sample stratification, and analysis of the effects of sample covariates at both the cell-subset and gene levels. Based on a hierarchical latent variable model and counterfactual predictions, MrVI is capable of addressing these tasks while accounting for nuisance sources of variation and estimating uncertainty without the need for *a priori* annotations of cells into types.

The latter point is of particular importance due to the difficulty of defining cluster boundaries and their resolution, which can vary substantially across single-cell studies. For instance, [180] categorized the human brain into 17 cell types for studying autism while [210] identified 3313 clusters to characterize cellular heterogeneity in the same tissue. Both strategies proved useful for their respective study, and it is therefore not generally clear which resolution is appropriate and to which type of analysis. Instead of relying on a fixed strategy for clustering of the cells, MrVI facilitates a "bottom-up"

approach that divides the cells into groups in a manner that reflects the task at hand. Specifically, by estimating sample distance matrices around each cell, MrVI allows for the aggregation of cells into subsets that confer similar groupings of samples. Similarly, estimating DE or DA effects in every cell allows for the aggregation of genes or cells in a way that reflects a coherent response to the covariate of interest. For ease of interpretation, these aggregations can also make use of *a priori* annotations of cells into subsets (by averaging the cell-wise DE or DA effects), as long as the cells within a subset are consistently affected (which was the case in many of our analyses).

MrVI relies on multi-head attention mechanisms to model heterogeneous effects of biological and technical covariates on gene expression. This approach contrasts with traditional generative models for single-cell omics, which typically employ multi-layer perceptrons to parameterize their models. To quantify the importance of the various components used by MrVI, we conducted an ablation study that illustrates the contribution of each component to properly characterize sample effects in the semi-synthetic experiment (**Figure 4.36**).

Besides model architecture, the performance of deep learning models is often tied to the selection of hyperparameters. To demonstrate the robustness of MrVI to hyperparameter choice, we conducted sensitivity analyses on key hyperparameters, including prior cell-type annotations, on the performance of MrVI (**Tables 4.2-4.3**). This analysis helped us select default hyperparameters for the MrVI implementation that have demonstrated proper integration and stratification of samples across multiple experiments. Section 4.5 of the Supplement provides guidelines to tune hyperparameters, e.g., in cases where sample integration with the default hyperparameters is not satisfactory.

We demonstrated the capability of MrVI to perform sample integration and stratification in a few case studies. Considering a COVID-19 cohort, MrVI identified clinically relevant patient groupings and highlighted a subsets of myeloid cells in which these groupings are manifested. Post-hoc analysis of the resulting patient strata further revealed a marked agreement with the elapsed time since infection - information that was not available to the algorithm. Importantly, this patient grouping did not perfectly mirror the infection timelines. For instance, some subjects with higher DSS exhibited molecular and cellular characteristics akin to those in the recently affected group. This observation does not undermine the validity of our approach; it rather underscores the potential of MrVI to produce data-driven sample strata that may not be trivially obtained from the recorded metadata alone and instead may lead to different diagnoses or identification of new disease subtypes [42]. MrVI is particularly relevant for studies in which samples are collected from numerous individuals, possibly collected across different anatomical locations or experimental protocols. We demonstrated this using a Crohn's disease study,

where it effectively integrated samples from diverse tissue locations and highlighted changes associated with stenosis. MrVI’s utility is not restricted to clinical studies or to other comparisons that are at the level of an experimental sample. Instead, it can be applied to any discrete cell-level meta-data by designating it as the target covariate. We demonstrated this using a perturbation screen with the sci-Plex assay in which each cell is associated with a particular perturbagen, facilitating *de novo* identification of compound groups and characterization of their effects.

When considering patient cohorts, we applied MrVI using the sample identifier as our modeled target covariate, s . This design enabled us to study the effects of any sample-level property (which is trivially nested by the sample identifier). MrVI’s ability to stratify samples without relying on explicit covariate information makes it robust to scenarios where patient metadata may be incomplete or inaccurate. This strategy may prove useful for cohort studies, where sample characteristics may be challenging to measure accurately or consistently. There are, however, scenarios where it is useful to explicitly model multiple sample-level target and/or nuisance covariates, which could constitute natural extensions of MrVI. For instance, conditioning on sample-level covariates with strong and unambiguous effects (e.g., sex, disease status) could help uncover more subtle effects attributed to other target covariates. Additionally, covariates could help improve integration in the u space, especially when specific cell states are not shared across all samples. Allowing MrVI to condition on continuous covariates (e.g., drug dosage, time) would also pave the way for inferring cell-state transitions and intermediate states.

While this work primarily focused on scRNA-seq data, a natural extension of MrVI is to handle information from other measurement modalities, both separately and in parallel to RNA expression. Such an extension could pinpoint, for instance, different patient strata (and their inducing cell subsets) when considering chromatin properties vs. RNA [211]. As another extension, adapting existing transfer learning protocols to MrVI [100] could enable the analysis of smaller datasets. In the same way that transfer learning leverages annotated cell atlases to label query datasets, transfer learning for MrVI could provide a way to harmonize samples across studies, especially when essential sample metadata are missing or inaccurate. For example, a MrVI model pre-trained on a large cell atlas with rich sample-level metadata (e.g., age, sex, inflammation status) can be leveraged to probabilistically impute unrecorded properties of new samples based on how they stratify relatively to the reference samples.

MrVI is implemented using state-of-the-art software tools for deep probabilistic modeling and can thus scale to multi-sample studies with millions of cells. Beyond that, the expected increase in scale and complexity of single-cell omics raise new challenges and opportunities for which MrVI can provide a powerful framework for

analysis and a solid foundation for further developments.

4.4 Methods

The MrVI model

Generative model overview

We consider two-stage scRNA-seq experimental designs in which cells are collected from multiple samples (**Figure 4.1a**). Each sample is associated with target covariates (e.g., treated vs. untreated, donor or specimen age and sex) or nuisance covariates (e.g., the sample collection site or the study ID in cross-study analyses).

Typically, multiple target covariates can induce variation in expression across samples, but it is unknown which of these may affect cells and by what mechanism. For instance, in drug response studies, both the type of administered drug and its dosage are crucial to assessing drug impact on cell states, but the nature of the interaction between these two factors may not be known. In disease studies, cases may induce specific shifts in gene expression in specific donor subpopulations that may not be fully encoded in the available metadata.

Instead of attempting to model the effects of these covariates directly, we adopt an approach that initially requires only knowledge of sample IDs $s \in \{1, \dots, S\}$ and the nuisance covariates as $b \in \{1, \dots, B\}$. This strategy allows us, at a later stage, to highlight which target covariates drive sample variations of interest. The resulting gene expression profiles are denoted as $\{x_1, \dots, x_N\}$, where $x_n \in \mathbb{N}^G$ is the vector of RNA transcript counts for cell n over the G observed genes. For any cell n , we let s_n identify the sample ID (e.g., the donor from which cell n originates) and b_n , the nuisance covariate.

In the case of multiple nuisance covariates, we recommend using the covariate with the coarsest resolution that is still nested within any covariates expected to confound the analysis. This may require concatenating multiple nuisance covariates (i.e., the study ID concatenated with the batch ID used in each study as b_n).

Isolating sample-specific effects on cell states with *MrVI*

The generative model of MrVI writes as

$$\begin{aligned}
 u_n &\sim \text{MixtureOfGaussians}(\mu_1, \dots, \mu_K, \Sigma_1, \dots, \Sigma_K, \pi_1, \dots, \pi_K) \\
 z_n \mid u_n, s_n &= g_\theta^{u \rightarrow z}(u_n, s_n) \\
 h_n &= \text{softmax}(A_{zh} \times [z_n + g_\theta^{z \rightarrow h}(z_n, b_n)] + \gamma_{zh}) \\
 x_{ng} \mid h_{ng} &\sim \text{NegativeBinomial}(l_n h_{ng}, r_{ng}).
 \end{aligned} \tag{4.1}$$

Here, u_n and z_n are the latent (unobserved) representations of cell n , both of dimension L . A_{zh} is a matrix of dimension $G \times L$ and γ_{zh} is a bias vector of dimension G . $g_\theta^{u \rightarrow z}, g_\theta^{z \rightarrow h}$ are multi-head attention layers, with disjoint sets of parameters both denoted as θ for brevity. The size factor l_n is fixed as the total sum of counts of cell n , and $r_{ng} \geq 0$ denotes the inverse dispersion of the distribution. μ_i, Σ_i, π_i respectively denote the mean, covariance matrix, and weight of component $i \leq K$. More details about this prior are provided in Section 4.4. All these parameters, other than l_n , are learned during training.

We now unpack some key aspects of the model. *A priori*, u_n captures broad variations assumed to characterize cell types and more granular cell states but is independent of both target and nuisance covariates. As such, u_n harmonizes cells from all samples into a shared latent space. We assume a mixture of Gaussians (MoG) prior or an unimodal Gaussian prior ($K = 1$) on u_n , depending on the application and available prior knowledge about cell-state variation. When we expect cells to belong to one of several groups, a MoG prior may be more appropriate than an unimodal Gaussian prior to avoid posterior collapse and prior overregularization, two issues with variational inference reported in the field [25, 212].

When reliable cell-type annotations are available, MrVI can also rely on a prior that is weakly informed about cell-type annotations. We found the cell-type-supervised model separated existing cell-type annotations far better than its unsupervised counterpart (**Figure 4.35**). More details about the prior of u_n are given in Section 4.4.

z_n is an augmented representation of the cell, that is aware of sample effects but is disentangled from other nuisance covariates. This latent variable is constrained to be close to u_n by a term in the objective function that penalizes the L2 distance between z_n and u_n .

These assumptions on u_n and z_n hold *a priori* but not necessarily *a posteriori*. For instance, the posterior distribution of u_n may exhibit conditional dependence on sample-specific factors, allowing MrVI to capture compositional differences across samples while harmonizing cells into a shared latent space.

As z_n is expected to capture more variability within cells than u_n , allowing it to lie in a higher-dimensional space is natural. Additionally, a low-dimensional bottleneck

on u_n may improve sample harmonization. In such a case, we allow z_n to take a higher dimension than u_n by modeling $z_n|u_n, s_n = g_\theta^{u \rightarrow z}(A_{uz}u_n + \gamma_{uz}, s_n)$, where A_{uz} is a learned matrix of dimension $L_z \times L$ and γ_{uz} is a bias vector of dimension L_z , where L_z is the dimension of z_n . Without loss of generality, the remainder of the manuscript focuses on the case where z_n and u_n have the same dimension, L .

Modeling gene expression under technical effects MrVI models the normalized expression of gene g , denoted h_{ng} , as a function of both z_n and the nuisance covariate. This relationship is parameterized with multi-head attention (g_θ above) to capture nonlinear nuisance-covariate-specific effects on gene expression. More information regarding this parameterization is given in Section 4.4. Lastly, we model the observed transcript counts with negative binomial distributions and account for the technical effects of the sequencing depth using the same approach as scVI [22].

Variational approximation and training procedure

The generative model described by Equation (4.1) can be used to generate synthetic data; it does not directly inform us on the posterior distribution of the latent variables u_n and z_n given observed gene expressions and sample ID of a given cell n required for analysis. Because we model z_n as deterministic given u_n and s_n , we only need to approximate the posterior $p_\theta(u_n | x_n)$. Since this posterior term is intractable, we rely on variational inference to learn an approximation $q_\phi(u_n | x_n)$ to the posterior, where ϕ denote all parameters used to construct the variational approximation.

Modeling $q_\phi(u_n | x_n)$: We model $q_\phi(u_n | x_n)$ as a Gaussian distribution whose mean and covariance (assumed diagonal) are outputs of multi-layer perceptrons (MLPs), f_ϕ , taking x_n as inputs.

With this variational approximation, a straightforward approach to get posterior latent u_n and z_n for a given cell consists of sampling u_n from $q_\phi(u_n | x_n)$, then computing z_n as described in Equation (4.1).

Training procedure We optimize the evidence lower bound (ELBO), which we maximize over the generative model parameters θ and variational parameters ϕ using mini-batch stochastic gradient descent methods [23, 213]. In this problem, the ELBO writes as

$$\text{ELBO}(\theta, \phi) := \mathbb{E}_{x,s,b} \mathbb{E}_{q_\phi(u|x)} \left[\log \frac{p_\theta(x | z, b) p_\theta(z | u, s) p_\theta(u)}{q_\phi(u | x)} \right]$$

As mentioned previously, we also add a scaled L2 penalty to this objective on the distance between z and u . Thus, the final objective writes as

$$\mathcal{L}(\theta, \phi) := \text{ELBO}(\theta, \phi) - c\mathbb{E}_{q_\phi(u|x)p_\theta(z|u,s)}[\|z - u\|_2^2]$$

where c is a scalar coefficient set such that the penalty term matches the log probability of z for an isotropic Gaussian distribution centered around u .

Model architecture Overall, MrVI uses a variety of MLPs, multi-head attention layers, and learnable embeddings to estimate the parameters of the different variational and likelihood distributions described above (**Figure 4.6**).

Exploratory analysis of sample effects on cell states

A common scenario in large-scale studies is that sample-level covariates are incomplete, noisy, or inconsistent across datasets. It may also be the case that the most relevant sample characteristics affecting gene expression are unobserved. Here, MrVI can identify the most relevant sources of heterogeneity between samples without assuming access to relevant target covariates (**Figure 4.1c**); the most salient axes of variation across samples can then be related back to observed target covariates. This type of analysis relies on cell-state counterfactuals, which are used to quantify sample distances at the cellular level.

Predicting counterfactual cell states After model fitting, MrVI can be used to predict the effect of a given sample on any cell. We aim to predict the counterfactual state of the cell, that is, its state had it been collected from another sample. We achieve this by substituting the sample-of-interest $s' \neq s_n$ for the true sample-of-origin s_n in Equation (4.1) to obtain

$$z_n^{s'} := u_n + g_\theta^{u \rightarrow z}(u_n, s'), \quad (4.2)$$

where u_n is the inferred cell state for cell n obtained via variational inference. $z_n^{s'}$ captures the counterfactual state of cell n had it been collected from sample s' .

Estimating sample distances at the cellular level MrVI allows for unsupervised sample stratification by comparing distances between counterfactuals from Equation 4.2. In fact, we can assess the differences between samples s_a and s_b on a cell n by computing the distance between their respective counterfactual cell states

$z_n^{s_a}$ and $z_n^{s_b}$. In particular, low distances between counterfactuals indicate that the two samples have similar effects on the cell according to the model.

Based on this observation, we summarize the sample stratification for a given cell as a sample distance matrix by computing the distance between counterfactuals for all pairs of samples. More precisely, for any cell n , we let $D(n)$ denote its sample distance matrix between counterfactuals. In this matrix, the element at the position indexed by (s_a, s_b) , where s_a and s_b are indices representing different samples, corresponds to the Euclidean distance between the counterfactual cell states $z_n^{s_a}$ and $z_n^{s_b}$.

These matrices inform sample stratification at single-cell resolution. They can first be used to identify cell populations with homogeneous sample stratifications. Clustering cells using their distance matrices as feature vectors can identify populations of cells with homogeneous sample stratifications. To do so, we embed each flattened distance matrix using PCA before clustering cells using the Leiden algorithm [214]. In any resulting cluster of cells, we then assess sample stratification in aggregate. We first compute the average sample distance matrix of the cluster, which we then use to cluster samples using hierarchical clustering.

Due to the uncertainty in u_n , even two samples with identical underlying distributions will have non-zero distances between their estimated counterfactual cell states. To account for this uncertainty, MrVI optionally computes Monte Carlo estimates of the distribution of distances between two counterfactual cell states derived from the same sample. These distribution estimates can then be used to z-score the original distance matrix values. In detail, for a cell, x_n , one can sample $u_n^1, u_n^2 \sim \hat{p}(u|x_n)$, then compute the L2 distance between them, $\|u_n^1 - u_n^2\|_2$. One can estimate the mean, $\hat{\mu}$, and standard deviation, $\hat{\sigma}$, of this term with more Monte Carlo samples and use these to compute normalized distances for $D(n)$ as $\frac{D(n) - \hat{\mu}}{\hat{\sigma}}$.

Assessing compositional and expressional sample differences

With observed sample covariates at hand, MrVI can also highlight which cells have different abundances or expression levels across groups (Figure 4.1d). Such characteristics can, for instance, correspond to age, sex, or disease status when samples correspond to different donors. The target covariate may also be derived from a stratification of samples based on the procedure described in the previous section. This section outlines how MrVI can be employed for both DE and DA analyses assessing sample differences in gene expression and cell composition.

Cluster-free assessment of differences in expression

First, MrVI can characterize differential expression patterns across samples. Suppose we observe C target covariates in the form of a vector $c^s \in \mathbb{R}^C$ for each sample s . To identify affected cells and genes by each target covariate, we fit the following linear model for each cell n :

$$z_n^{s'} = c^{s'T} \beta_n + u_n, \quad \forall s', \quad (4.3)$$

where $z_n^1, \dots, z_n^S$ are S counterfactuals for cell n obtained from Equation (4.2). Here, $\beta_n \in \mathbb{R}^{C \times L}$ is the vector of regression coefficients obtained via least-squares regression.

Identifying the effect of covariates on cells This linear model can first quantify the overall effect of an observed covariate on any cell. We compute, for any cell n and covariate index $j \leq C$, the Chi-squared statistic of β_n^j . This statistic quantifies the extent to which the observed covariate j explains the variation in the counterfactual cell states.

Detecting cells strongly affected by covariate The results of the linear regression can help identify cells strongly affected by a covariate. We compute the L2 norm of the vector β_i for a covariate i . This yields effect strengths in the z representation for the specific covariates and can be used to compare effect strength across multiple cell types.

Detecting differentially expressed genes The results of the linear regression can also identify DE genes associated with a given covariate in any cell. For simplicity, assume that the covariate of interest is binary. Let $\beta_n^1 \in \mathbb{R}^L$ denote the regression coefficients of the covariate of interest for cell n . To identify the associated DE genes, we decode the counterfactual cell state $z_n^1 = \beta_n^1 + u_n$ and the reference cell state $z_n^0 = u_n$. This computation yields two vectors of decoded gene expressions, denoted as h_n^1 and h_n^0 . We then compute the log-fold change between these two vectors, measuring the effect of covariate j on each of the observed genes g in the cell.

Accounting for out-of-distribution samples Prior to conducting the described procedure, we first identify and discard samples that are out-of-distribution for any given cell. Samples will be out-of-distribution for a cell if no cell from that sample was collected in a similar cell state in the u_n space. For these samples, the model has insufficient information to accurately infer realistic counterfactual cell states in Equation (4.2). Thus, we conservatively discard the sample s for cell state u if the

maximum density reached at u with respect to the approximate variational posterior distributions falls below a given threshold τ . More details on how the densities are computed and how τ is chosen are given in Section 4.4.

Assessing cluster-free differences in composition

Last, our approach can identify differentially abundant cell populations over groups of samples using log ratios of aggregated posterior densities. For this purpose, we introduce q_s , the aggregated posterior distribution for a given sample s , which corresponds to $q_s(u) := 1/n_s \sum_{n:s_n=s} q(u | x_n)$, where n_s denotes the number of cells in the considered sample and $q(u | x_n)$ is the variational approximation to the posterior distribution over the u space for cell n . We can then quantify the density of any set of samples $A \subset \{1, \dots, S\}$ in the u space as $q_A(u) := \frac{1}{|A|} \sum_{s \in A} q_s(u)$.

For two disjoint sets of samples A and B , we quantify the relative overabundance of cells from A compared to B at any cell state u by computing the log density ratio of the aggregated posterior densities of the two groups, i.e.,

$$r_{AB}(u) := \log \frac{q_A(u)}{q_B(u)}. \quad (4.4)$$

We can then identify enriched or depleted regions of u in A compared to B via inspection of the log ratio $r_{AB}(u)$. This approach has several benefits. As MrVI assesses differential abundance in the u latent space, the captured differential abundance effects are orthogonal to the differential expression effects quantified in the previous section. Furthermore, this approach allows us to identify enriched cell states without requiring cell-type or neighborhood assignments.

At a cluster level, we also devise a strategy to identify cell-enriched or depleted subpopulations with statistical confidence. For this purpose, letting A denote a cluster of interest, we collect log-ratios for (i) all cells in cluster A (ii). those not in A . We then test for difference in the mean of these two sets of log-ratios using a two-sided, two-sample t-test. To avoid detecting differences as significant due to large sample sizes, the t-test rejects for the composite null that the mean difference is, in absolute value, less than a given threshold δ . Throughout the experiments, we set $\delta = 0.1$.

Additional model details

Mixture of Gaussians prior for u_n

MrVI posits a mixture of Gaussians (MoG) prior on u_n , that writes as

$$\begin{aligned} c_n &\sim \text{Categorical}(\pi_1, \pi_2, \dots, \pi_K), \\ u_n \mid c_n = c &\sim \mathcal{N}(\mu_c, \Sigma_c). \end{aligned} \tag{4.5}$$

In practice, we assume the covariance matrices to be diagonal and learn $\mu_1, \dots, \mu_K$, $\Sigma_1, \dots, \Sigma_K$, and $\pi_1, \dots, \pi_K$ during training using maximum likelihood estimation.

When cell-type annotations are available, MrVI can weakly encourage the mixture of Gaussians to align with these annotations. In this case, we set K to be the number of unique cell-type annotations and reparameterize the mode of the Gaussian distribution as $c_n \sim \text{Categorical}(\pi'_{1,n}, \pi'_{2,n}, \dots, \pi'_{K,n})$, where $\log \pi'_{k,n} = \log \pi_k + \epsilon \mathbb{I}(y_n = k)$. Here, ϵ is a positive constant (with a default of 10), y_n is the cell-type annotation of cell n , and $\mathbb{I}(\cdot)$ is the indicator function.

Parameterization of multi-head attention layers

Two components of MrVI rely on multi-head attention layers, corresponding to the mappings $g_\theta^{u \rightarrow z}$ and $g_\theta^{z \rightarrow h}$ from Equation (4.1). We now provide details on how these layers are parameterized, illustrated in Figure 4.7.

Parameterization of $g_\theta^{u \rightarrow z}$ $g_\theta^{u \rightarrow z}$ takes as inputs the cell-state u_n and the sample s_n from which the cell was collected. We associate each sample ID $s \in \{1, \dots, S\}$ with an embedding $e_s \in \mathbb{R}^L$, learned during training. We then rely on a multi-head attention mechanism to capture the effect of s_n on u_n in a nonlinear fashion, considering u_n as queries and e_{s_n} as keys/values. This output is then passed through a series of two fully connected layers, with ReLU activations, to obtain the actual output of $g_\theta^{u \rightarrow z}$.

Parameterization of $g_\theta^{z \rightarrow h}$ $g_\theta^{z \rightarrow h}$ relies on the exact same parameterization as $g_\theta^{u \rightarrow z}$, but takes z_n and b_n as inputs instead of u_n and s_n .

Out-of-distribution checks

MrVI DE module actively filters for out-of-distribution cell/sample pairs. It may, for instance, be the case that a given sample contains no cells of a given type; in this case,

this sample should be discarded for the DE analysis in Equation (4.3). We identify these out-of-distribution samples by defining the following admissibility score:

$$A_s(u) := \max_{n:s_n=s} q(u \mid x_n). \quad (4.6)$$

For a given cell n , we filter out sample s whenever $A_s(u) \leq \tau_s$, where τ_s is set to the 5% quantile of reference admissibility scores $R'_s(n)$ over all of the cells collected in s , where

$$R'_s(n) := \max_{\{i:s_i=s\} \setminus n} q(u \mid x_i).$$

Importantly, when computing the baseline admissibility scores over in-sample observations, we exclude the cell itself. After computing the set of admissible samples for every cell, additional filtering can be performed on the level of cells to eschew those with very few admissible samples (e.g., a rare cell type observed in one sample) for which counterfactual estimates may be generally unreliable (see **Figure 4.37**).

Benchmark

Baselines

Exploratory analyses We considered two approaches that stratify samples based on differences in cell cluster abundance. Both approaches compare subcluster proportions between samples to yield distance matrices. More particularly, they subcluster each predefined cell group with the Leiden algorithm [214] using low-dimensional cell representations, i.e. PCA for Composition (PCA) or scVI for Composition (SCVI). The scVI model was given the batch IDs to correct for batch effects in the latent representation. The PCA algorithm, on the other hand, does not have an explicit way to handle batch IDs. The distance between two arbitrary samples is then defined as the Euclidean distance between their subcluster proportions.

Guided analyses We also considered Milo [185] and miloDE [215], which leverage estimates for DA and DE, respectively, in guided analyses. Milo is a statistical framework that aims to detect cell neighborhoods enriched in certain sample groups based on a nearest-neighbor graph of cells. Built on top of Milo, miloDE [215] performs differential expression tests for each neighborhood identified by Milo by comparing each neighborhood against adjacent ones. These approaches, however, do not provide effect sizes for DA and DE at the cell level and instead group cells into neighborhoods that may obscure effect sizes at a single-cell resolution [41]. To

compare these approaches to MrVI, we computed cell-level effect sizes for Milo and miloDE by defining cell-level effect sizes as the average effect size of the neighborhoods to which each cell belonged.

Metrics

Cell-type silhouette scores We consider averaged silhouette width scores computed as in [24] to assess the relevance and the proper mixing of the latent representation u under the assumption that the same cell types appear across the considered samples. To do so, we first compute the silhouette score with respect to author-provided cell-type annotations. For any cell n with cell representation $r(n)$, belonging to annotation C_o , let $d(n, C)$ denote the mean distance of $r(n)$ to representations of annotation C , excluding n if $C = C_o$. let $a(n)$ denote the average distance of $r(n)$ to cells of the same annotation, and $b(n)$ the smallest mean distance of $r(n)$. The silhouette score for cell n is computed as

$$s(n) = \frac{\min_{C, C \neq C_o} d(n, C) - d(n, C_o)}{\max\{\min_{C, C \neq C_o} d(n, C), d(n, C_o)\}}, \quad (4.7)$$

and the overall dataset silhouette score is the average of rescaled silhouette scores across all cells in the data. The rescaling, $\tilde{s}(n) = \frac{1}{2}(s(n) + 1)$, puts the dataset score in the range $(0, 1)$. This score assesses to what extent the data representations cluster according to the annotations. When the dataset score is equal to 1, representations with the same annotation perfectly cluster together.

Batch silhouette scores We also used the silhouette to measure the extent to which batch IDs mix together in the latent space. To do so, we follow the procedure described in [24], which consists of, for each previously-annotated cell type: (i) computing cell silhouette scores with respect to the batch assignments, (ii) rescaling these scores, such that $\hat{s}(n) = 1 - |s(n)|$, and (iii) computing an overall silhouette score computed as a weighted average of $\hat{s}(n)$, to ensure that each cell type gets the same contribution.

Data and preprocessing

We here provide a brief description of the datasets and of the preprocessing steps applied to them. More details can be found in the **Supplementary Methods**.

Semi-synthetic experiment We constructed a semi-synthetic dataset containing controlled DE and DA effects, based on an original PBMC dataset of 68K cells [90]

(**Figure 4.38**). The synthetic dataset contains 32 synthetic study subjects as well as five cell subsets, each of which can be seen as a cell type. In the first subset, denoted as A, there are differences in expression across subject groups characterized by *covariate 1*. While we have no exact ground truth for the subject-subject distance in subset A, our ground truth consisted of a dendrogram, or tree, over study subjects, characterizing the similarities between subjects in terms of gene expression in the cell subset A. Specifically, all subjects sharing the same *covariate 1* value shared similar gene expression values for cells in subset A. In two other cell subsets, denoted as B and C, cells had no differences in expression over subjects but exhibited differences in abundance, either corresponding to enrichment or depletion of these cell subsets in a specific group of samples. In particular, all subjects sharing the same *covariate 2* value shared similar proportions of cell subsets B and C but different proportions with the other subjects.

COVID experiment The original dataset [16] contained a total of 650K PBMC cells sequenced across three sites: Cambridge, Sanger, and Newcastle. We discarded cells coming from Cambridge and Sanger, and focused on data points sequenced in Newcastle. We retained the 10,000 most variable genes using Seurat v3. The resulting dataset contained 418,768 cells, originating from 55 patients. No additional cell or gene filtering was performed.

sci-Plex experiment The original sci-Plex dataset [15] contains gene expression over three cell lines, exposed to 188 small-molecule drugs at four different doses. We filtered for drugs with no effects, resulting in a final dataset containing 251,088 cells with 92 drugs at all four doses and the vehicle cells. We applied highly variable gene selection using Seurat v3, retaining the top 5,000 genes.

IBD experiment We downloaded the dataset [14] from the Broad Institute Single Cell Portal, conducted in 46 patients and 25 controls across different tissue regions and sample preparation protocols, for a total of 463k cells. We retained the 10,000 most variable genes using Seurat v3.

Code availability and reproducibility

MrVI is available within scvi-tools (<https://scvi-tools.org/>). Code to reproduce the experiments is available at <https://github.com/YosefLab/mrvi-reproducibility>.

4.5 Supplementary Information

Supplementary Figures

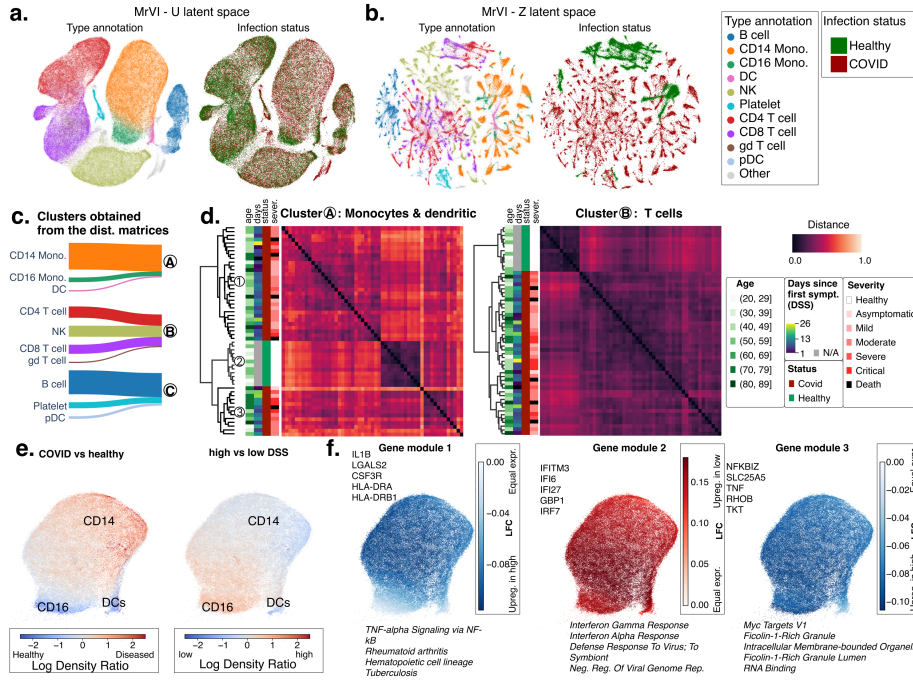


Figure 4.3: Analysis of a COVID-19 cohort with MrVI. **a.** and **b.** Minimum distortion embeddings (MDEs) of u and z latent spaces from MrVI, colored by cell-type annotations and COVID-19 status. Legends for cell-type annotations and COVID-19 status are shared and displayed on the right of **b.** **c.** Sankey plot mapping cell-type annotations to three clusters obtained by Leiden clustering of cell-specific distance matrices. Cluster **A** contained monocytes and dendritic cells, Cluster **B** contained T cells and NK cells, and cluster **C** contained B cells. Cell-type/cluster pairs with less than 1% of the total cells were not displayed. **d.** Sample distance matrices averaged over cells from two subpopulations identified in **c.** For each matrix, we computed the affinity dendrogram between samples and colored each row by patient age, days since first symptoms (DSS), infection status, and disease severity. **e.** Differential abundance analysis using MrVI log density ratios for myeloid cells (cluster **A**). *Left:* COVID-19 positive patients vs healthy controls. *Right:* COVID-19-positive patients with high vs low DSS. **f.** Differential expression analysis using MrVI between COVID-19-positive patients with high and low DSS. MrVI identified three DE modules of genes. Each plot shows the activity of the module in the u latent space, with LFCs averaged over all genes in the module. Low and high DSS patients correspond to donor clusters **1** and **3** in **d.**

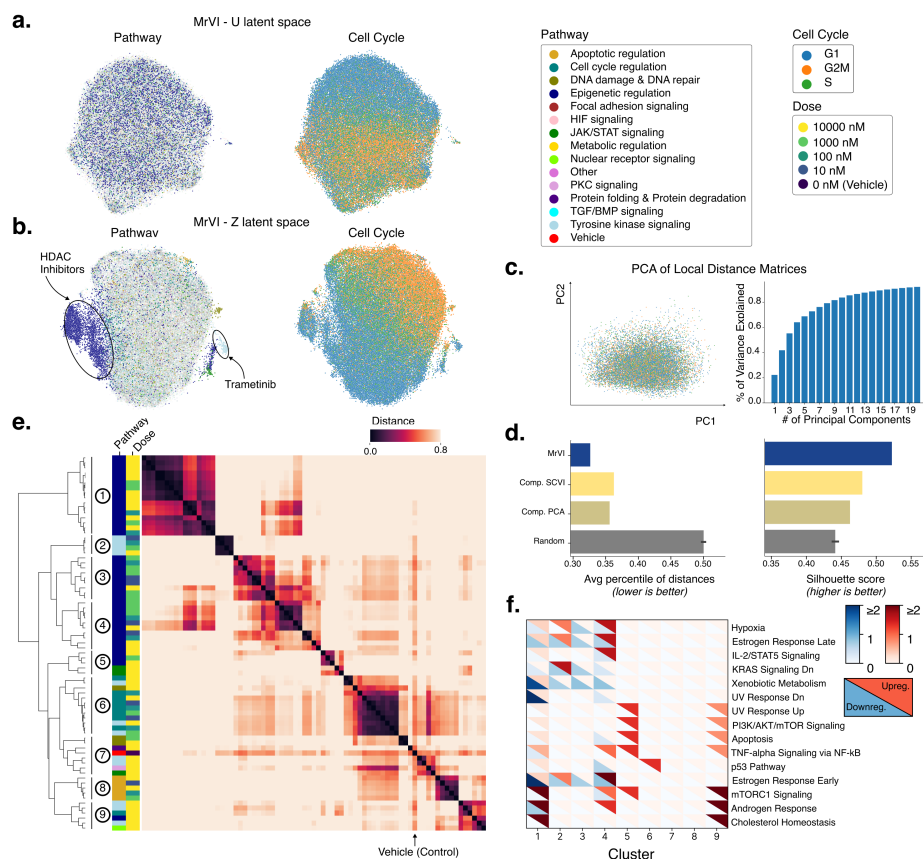


Figure 4.4: Analysis of the A549 cell line in the sci-Plex experiment. We fit the model over 92 drugs each at four doses that passed our simple DE-gene filter. **a.** and **b.** MDEs of the u and z latent spaces from MrVI colored by drug target pathway (*left*) and cell cycle stage (*right*). Top 20% of samples by distance from vehicle shown in full opacity. **c.** PCA of sample distance matrices showing no cell cycle-based subclusters (*left*) and variance explained (*right*). **d.** Performance metrics comparing MrVI to benchmarks: average distance percentile between same-drug samples (*left*) and silhouette score based on Connectivity Map (*right*). **e.** Hierarchical clustering of sample distance matrix annotated by pathway, dose, and inferred clusters. **f.** GSEA scores for MSigDB Hallmark gene sets comparing each cluster to vehicle cells. Upper-right and bottom-left triangles show up- and down-regulated genes. Analysis in **e.** and **f.** focuses on top 20% of drug-dose combinations by distance from vehicle (74/368). See **Figure 4.17** for full matrix and **Figure 4.16** for labeled version. Legends for **a.**, **b.**, **c.**, and **e.** in top right.

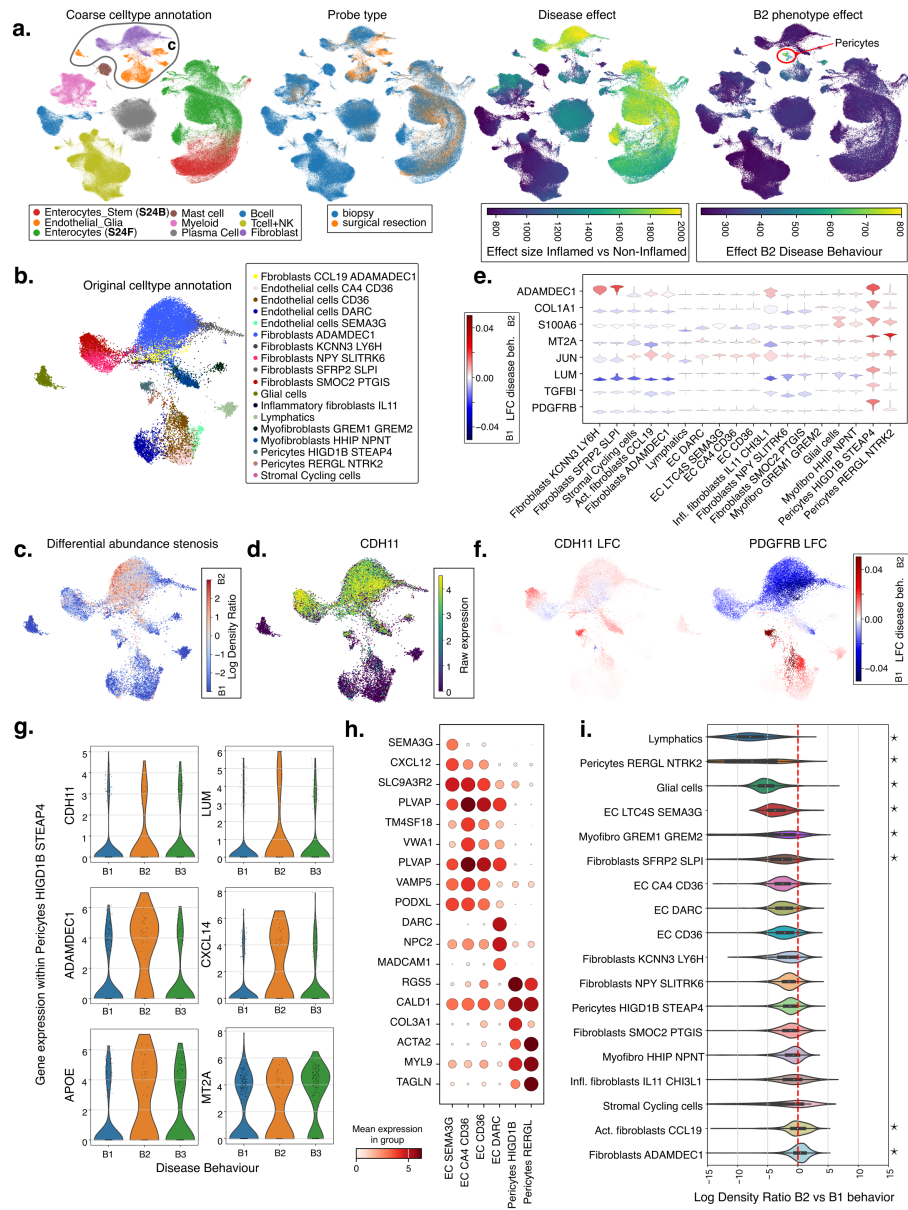


Figure 4.5: Characterization of stenosis in Crohn’s disease. **a.** UMAP embedding of u latent space colored by different sample-level covariates. From left to right: coarse labels identified by us to stratify cell types for unguided analysis (circled is the subset analyzed in **b-e**); tissue collection method highlights a bias for stromal cells in surgical specimens; the inferred effect size of inflamed vs. non-inflamed and the inferred effect size of B2 disease behavior (highlighted are the small subpopulations of pericytes with the strongest effect (**Figure 4.33**)). For these last two plots, this effect size corresponds to the squared norm of β from Equation (4.3); it characterizes the overall effect of the covariate on gene expression. **b-e.** Analysis of the cell population circled in **a**, using the same UMAP embeddings as **a**. **b.** Original cell-type labels provided in the original study. Displayed are fine annotations for stromal cells and coarse labels for the other cells. **c.** DA analysis of stromal cells based on MrVI comparing the B2 disease phenotype against the B1 phenotype (red denotes higher in B2 disease behavior). Displayed is the log density ratio between B2 and B1 disease behavior. **d.** Raw expression of *CDH11* inside all stromal cells. Values are library-size normalized and log1p transformed. Cells are sorted for display based on higher expression. **e.** Inferred LFCs from MrVI for the comparison of B2 and B1 disease behavior based on multivariate analysis in MrVI while correcting for inflammation status, sex, tissue location, and chemistry in the different cell types. The violin plots display the distribution of estimated LFCs per cell type. We relied on hierarchical clustering to determine the order of cell types displayed in the violin plots. **f.** UMAP of two genes highlighting intra-cell-type DE variation. Cells are colored based on the genes’ LFC for the comparison of B1 and B2 classifications. **g.** Violin plots illustrating changes in normalized gene expression between inflamed and non-inflamed biopsies from patients categorized as B1, B2, and B3 disease behavior after subsetting to *Pericytes HIGD1B STEAP4*. Raw gene expression values are normalized by library size and log1p-transformed. **h.** Dotplots displaying top 3 marker genes for the different endothelial cell (EC) subclusters. Lymphatic endothelial cells were excluded from this analysis. **i.** Differential abundance from **c** displayed as violin plots to demonstrate changes in abundance across cell types. Displayed is the log density ratio between samples from B2 and B1 disease behavior. We report the significance of the hypothesis that the difference of log density ratios between a given cell type and all other cell types is above 1 in absolute value (see **Methods**).

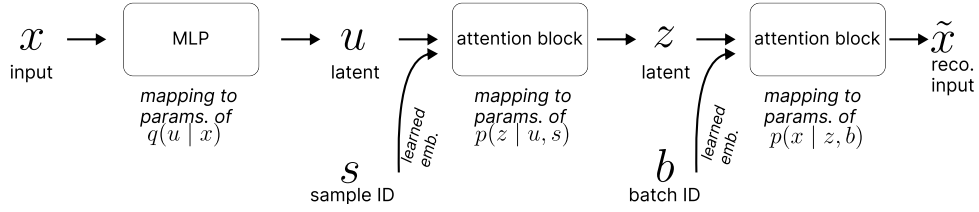


Figure 4.6: Functional description of MrVI.

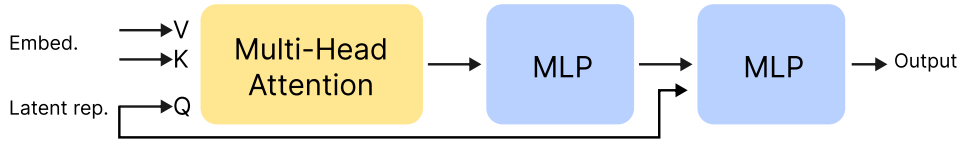


Figure 4.7: Illustration of the parameterization of the multi-head attention mechanism used in MrVI. For any cell n , this mechanism takes a embedding and a latent representation as inputs. These two inputs respectively serve as keys/values and queries for the attention mechanism. This output is then passed through a series of fully connected layers to obtain the final output.

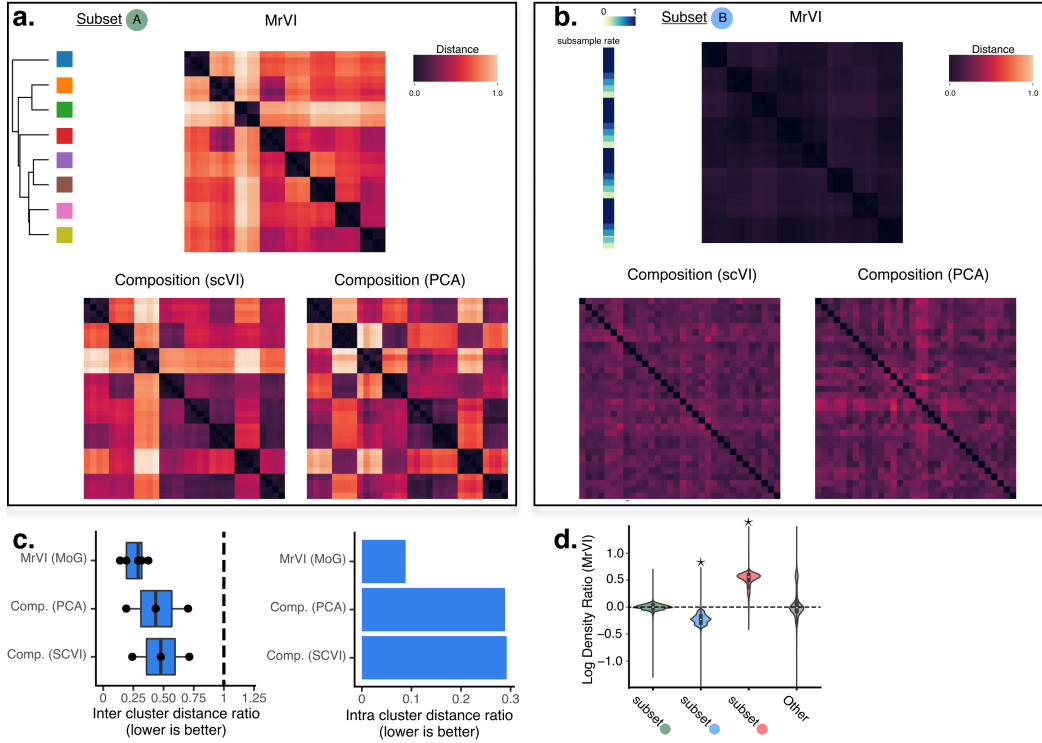


Figure 4.8: Supplementary results for the semi-synthetic experiment. **a.** and **b.** Distance matrices for MrVI and compositional baselines, computed for cell subsets A and B. Row and column ordering is the same for all approaches. Because cells of subset **B** were randomly assigned to synthetic samples, we expect no differences between samples in **b.** **c.** Distance matrix quality scores. *Left:* Intra-cluster distance ratios, quantifying sample effects in subset A (with DE) vs other subsets (without DE) (*lower is better*). For each cell cluster r different than A, we computed the ratio of average distance matrix elements for r to A (excluding diagonal). Since r has no intra-cell-type variation by design, these ratios should be close to 0. *Right:* Mean ratio of within-category to between-category smallest distance for *covariate 1* (*lower is better*). This ratio measures how well the sample distance matrix for subset A retrieves the ground-truth sample stratification. **d.** Violin plot of MrVI DA log ratios from Figure 4.2d. Stars denote statistically significant DA ($p < 10^{-8}$) in the given subset based on t-tests with Benjamini-Hochberg correction. Only the two cell subsets that are DA by construction are detected as DA.

Method	Bio conservation					Batch correction					Aggregate score		
	Isolated labels	KMeans NMI	KMeans ARI	Silhouette label	cLISI	Silhouette batch	iLISI	KBET	Graph connectivity	PCR comparison	Batch correction	Bio conservation	Total
MrVI (MoG)	0.58	0.71	0.50	0.58	1.00	0.73	0.15	0.71	0.82	0.00	0.48	0.67	0.60
scVI	0.62	0.70	0.49	0.57	1.00	0.80	0.09	0.44	0.81	0.00	0.43	0.68	0.58
PCA	0.48	0.67	0.43	0.55	1.00	0.86	0.10	0.50	0.79	0.00	0.45	0.63	0.55

Figure 4.9: SCIB metrics for MrVI, PCA, and scVI evaluated on the COVID-19 experiment. Here, scVI used donor IDs as batch indices, and PCA was applied with $K = 50$ first principal components on log-counts per 10^4 normalized data. We computed the SCIB metrics using cell-type annotations (4.3C) from the original study as labels, and the donor ID as the batch key.

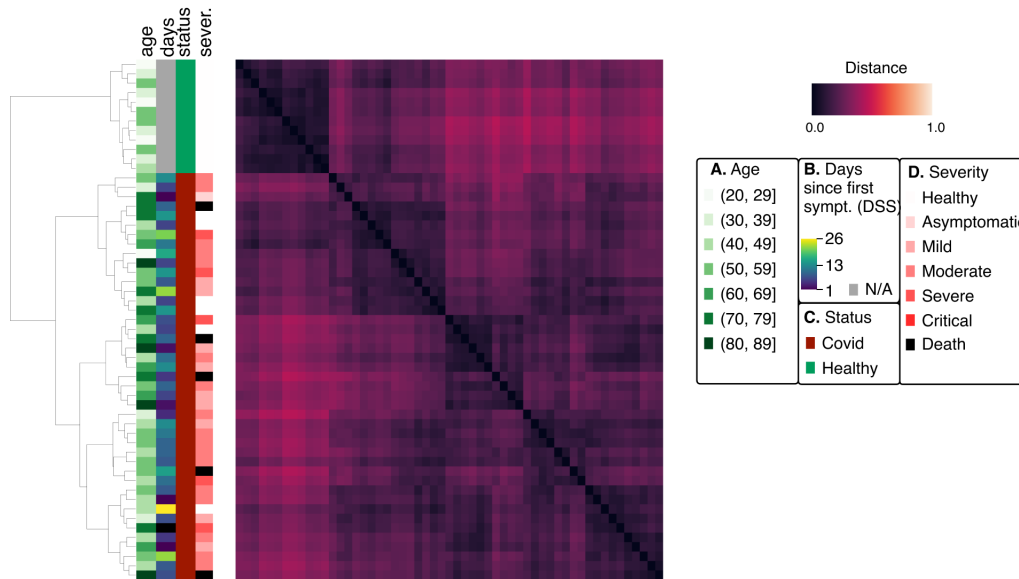


Figure 4.10: Distance matrix averaged over all cells for cluster C, corresponding to B cells, identified in Figure 4.3c for the COVID-19 experiment. This figure relies on the same color scheme as in Figure 4.3d. We see a clear separation of healthy individuals.

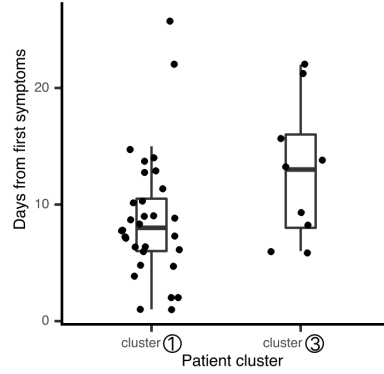


Figure 4.11: Comparison of time since symptom onset (“Days from onset” in [16]) for the two COVID-19 patient clusters from Figure 4.3d. These differences are significant under a Mann-Whitney U test ($p < 0.05$). We find several outliers, which might be attributed to the exact course of disease, or to noise due to the fact that symptom onset is self-reported. This ambiguity can’t be resolved with the current data.

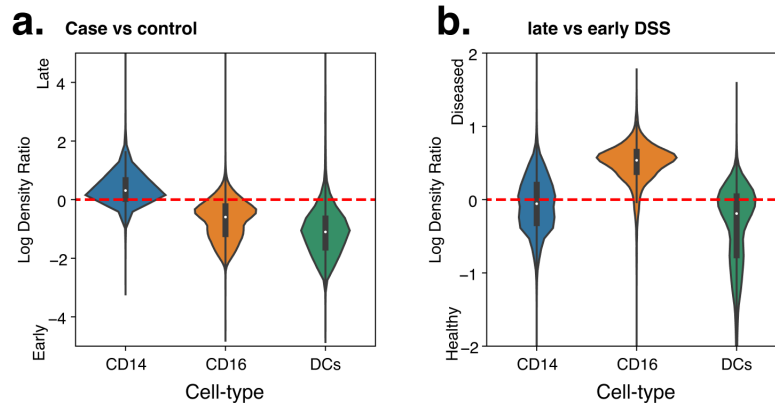


Figure 4.12: Comparison of the log density ratio displayed in Figure 4.3e across the different cell types contained in cluster ①.

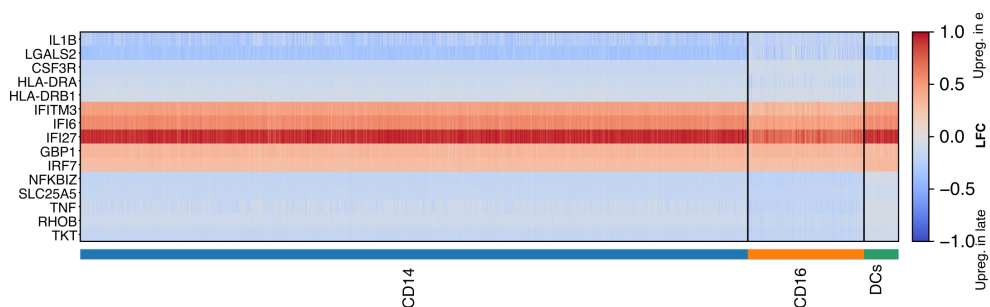


Figure 4.13: Heatmap of MrVI LFCs for the DE genes identified by in Figure 4.3f, for the comparison of the two groups of COVID patients, corresponding to late and early onsets.

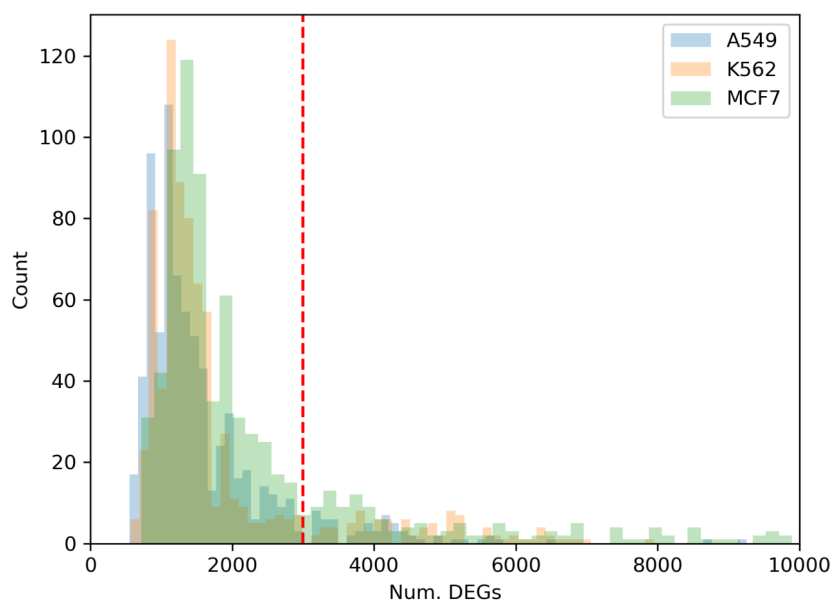


Figure 4.14: Histograms of the number of DEGs for the maximum dose (10000nM) of each drug-cell-line combination with respect to the vehicle for the three cell lines in the sci-Plex dataset. The vertical red dotted line corresponds to the cutoff used to filter out drugs with insignificant effects in all of the cell lines.

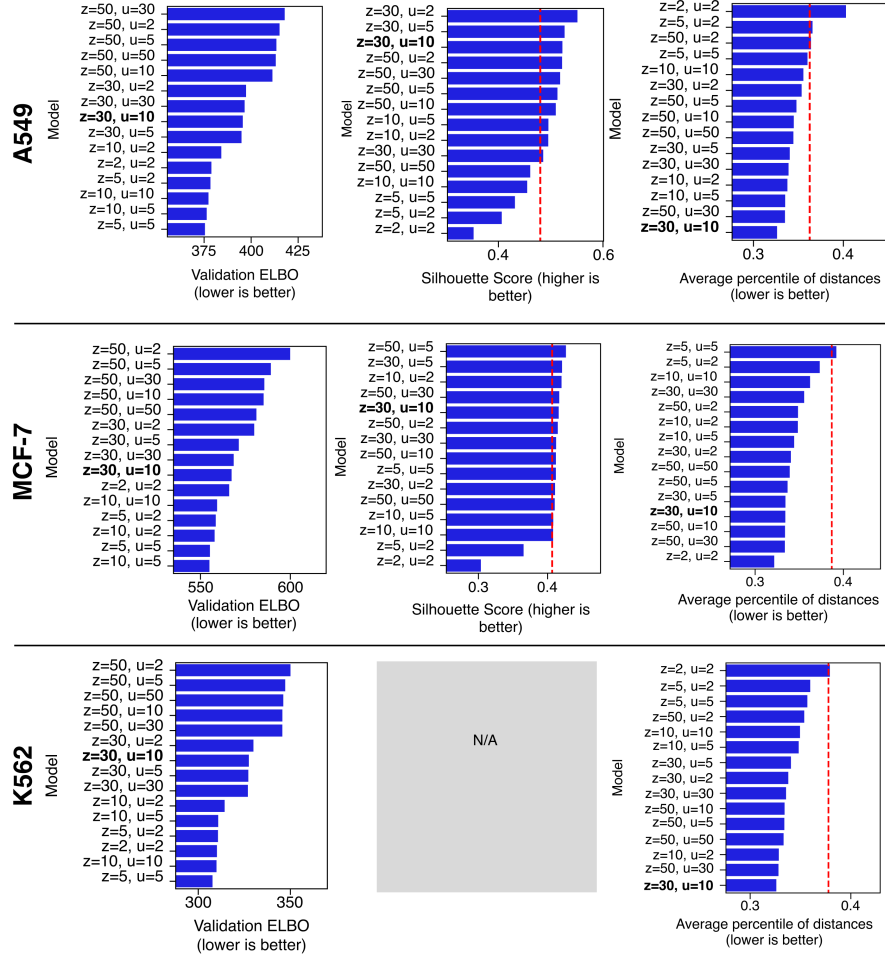


Figure 4.15: Comparison of performance of MrVI for different parameter combinations for the sci-Plex dataset. Model ELBO evaluated on validation cells, silhouette score with respect to clusters found in transcriptomic Connectivity Map data, and average percentile of within-drug sample distances over each cell line. There was no available Connectivity Map data for the K562 cell line, so we could not compute the silhouette metric for this cell line. The bold configuration denotes the hyperparameters chosen for the analysis in Figure 4.4, 4.21, 4.22 respectively. The red dotted vertical lines indicate the best performance between the CompositionPCA and CompositionSCVI baselines for the silhouette score and average percentile metrics.

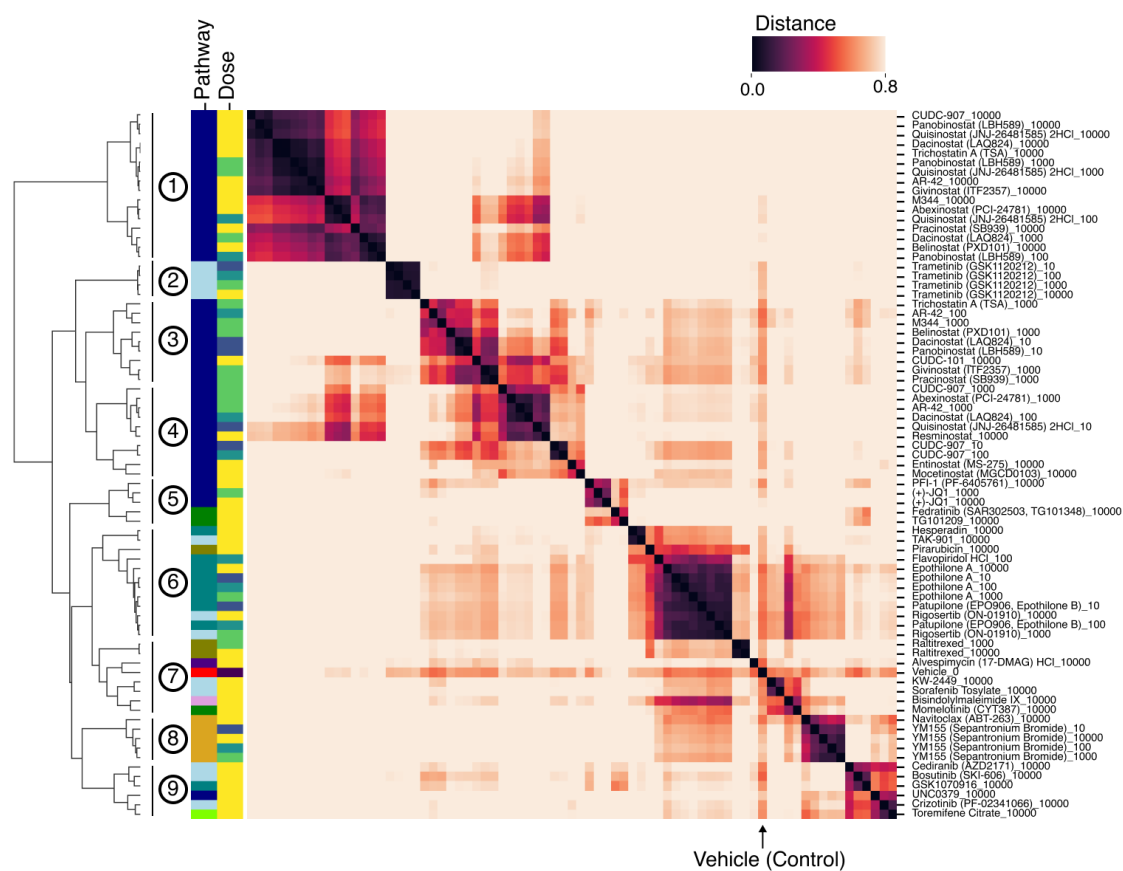


Figure 4.16: Copy of Figure 4.4e with row labels for each drug-dose combination.

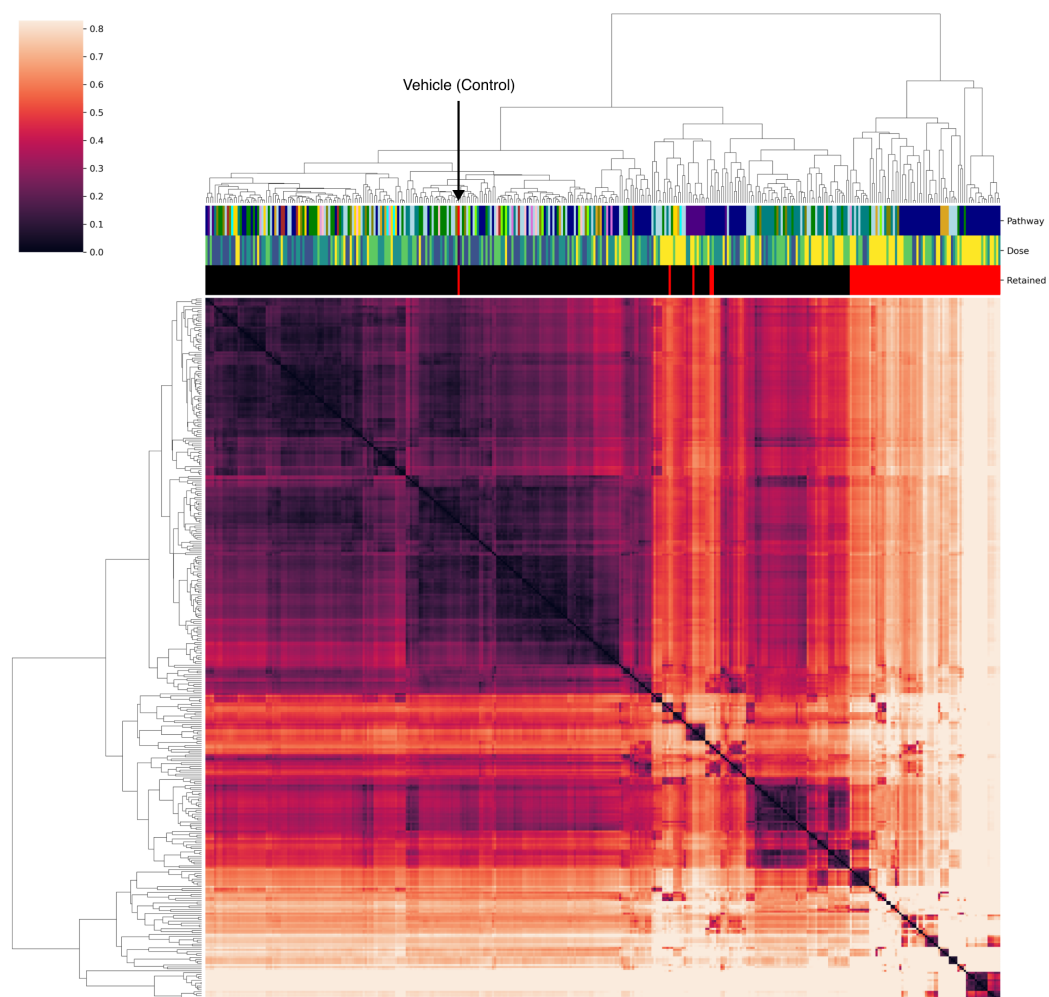


Figure 4.17: Sample distances of all 369 samples (92 drugs at four doses and vehicle) used in the analysis for the A549 cell line. The columns are annotated by each drug's pathway annotation from the original study, dosage level, and whether the sample was retained for the remaining analysis (top 20 percent of samples based on distance from vehicle). The hierarchical clustering was performed with the Ward variance minimization algorithm.

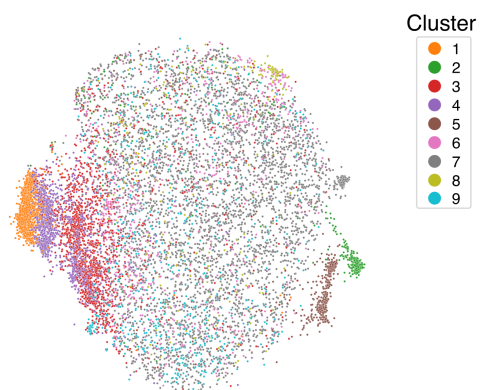


Figure 4.18: MDE of the z latent space from MrVI of data from the A549 cell line colored by the same cluster assignments as in Figure 4.4e. Clusters ③, ④, ① capture a large group of samples with similar and increasingly divergent cell states corresponding to increasingly larger doses of HDAC inhibitors.

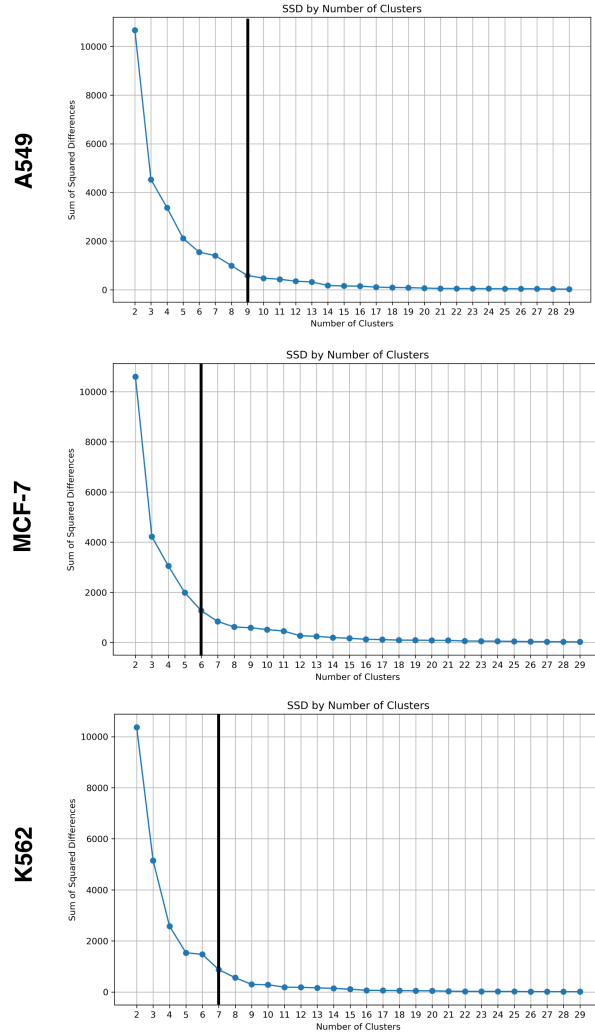


Figure 4.19: Sum of squared differences against the number of clusters used for hierarchical clustering for each cell line in the sci-Plex dataset. This metric was computed by applying the `maxclust` algorithm in `scipy.cluster.hierarchy.fcluster` over the output of agglomerative clustering, then aggregating the within-cluster, pairwise squared differences. Each vertical line denotes the number of clusters selected for the remaining analysis for each cell line, which was identified as the elbow point from a visual inspection of the plot.

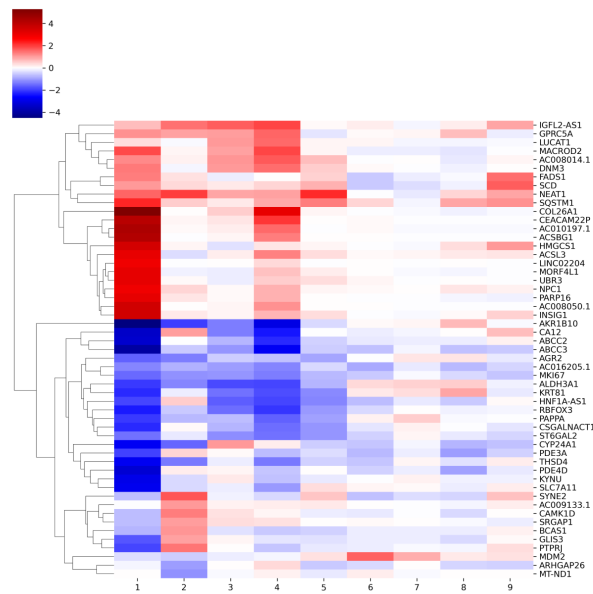


Figure 4.20: Heatmap of LFCs of the top DE genes identified by MrVI in the A549 cell line. Each column corresponds to a cluster as labeled in Figure 4.4e. Displayed is the LFC estimated by MrVI averaged over all cells. Concordant with the distance matrix, we find the most significant and widespread gene-specific effects in clusters ① and ④.

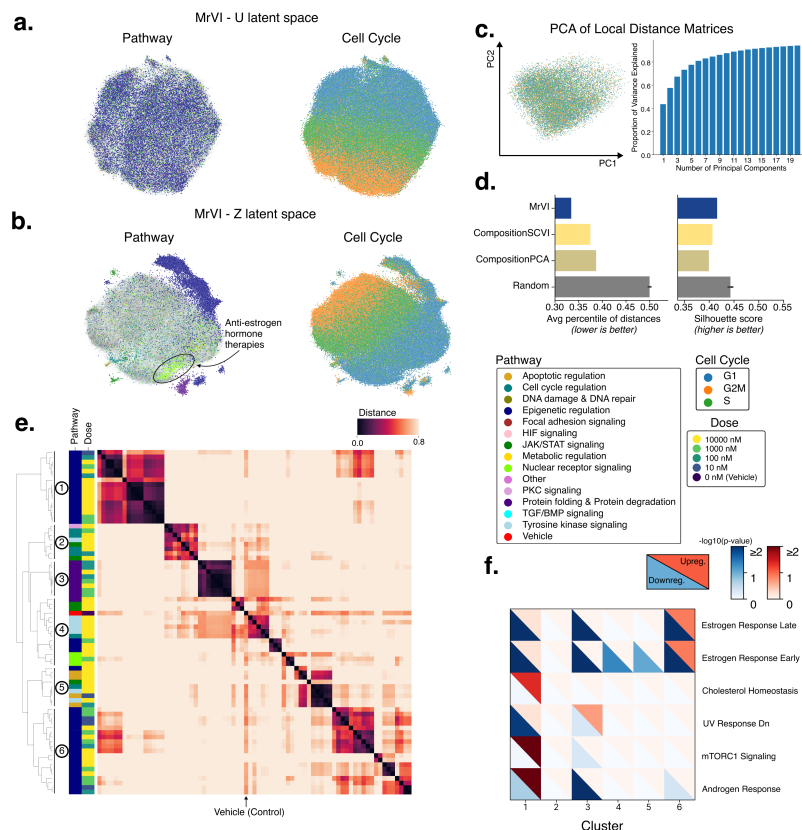


Figure 4.21: Analysis of the MCF-7 cell line in the sci-Plex experiment using the same analysis as in 4.4. MrVI was fit over 92 drugs each at four doses that passed the DE-gene filter. **a.** and **b.** MDEs of the u and z latent spaces from MrVI colored by the target pathway of the drug used to treat each cell (*left*) and the cell cycle stage of each cell (*right*). For the MDEs colored by target pathway, only the top 20 percent of samples based on the distance from the vehicle are shown in full opacity. **c.** PCA of sample distance matrices. **d.** Barplots comparing MrVI against the benchmark methods for two performance metrics that determine alignment with prior knowledge. **e.** Hierarchically clustered sample distance matrix. **f.** GSEA scores for the Human MSigDB Hallmark gene set collection for differentially expressed genes found for each cluster found in panel (e) with respect to the vehicle cells.

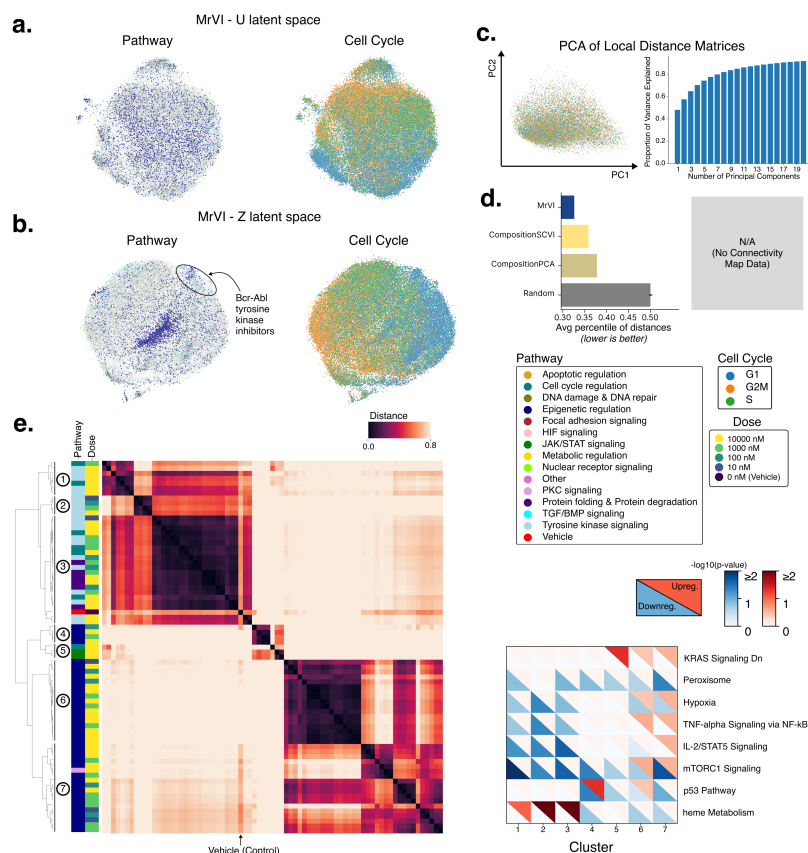


Figure 4.22: Analysis of the K562 cell line in the sci-Plex experiment using the same analysis as in 4.4. MrVI was fit over 92 drugs each at four doses that passed the DE-gene filter. **a.** and **b.** MDEs of the u and z latent spaces from MrVI colored by the target pathway of the drug used to treat each cell (*left*) and the cell cycle stage of each cell (*right*). For the MDEs colored by target pathway, only the top 20 percent of samples based on the distance from the vehicle are shown in full opacity. **c.** PCA of sample distance matrices. **d.** Barplot comparing MrVI against the benchmark methods for a performance metric that determines alignment with prior knowledge. The average percentile of distances measures how much closer samples with the same drug and different doses are to each other relative to the rest of the distances. We expect the average percentile to be low. There was no available Connectivity Map data for the K562 cell line, so we could not compute the silhouette metric for this dataset. **e.** Hierarchically clustered sample distance matrix. **f.** GSEA scores for the Human MSigDB Hallmark gene set collection for DE genes found for each cluster found in e. with respect to the vehicle.

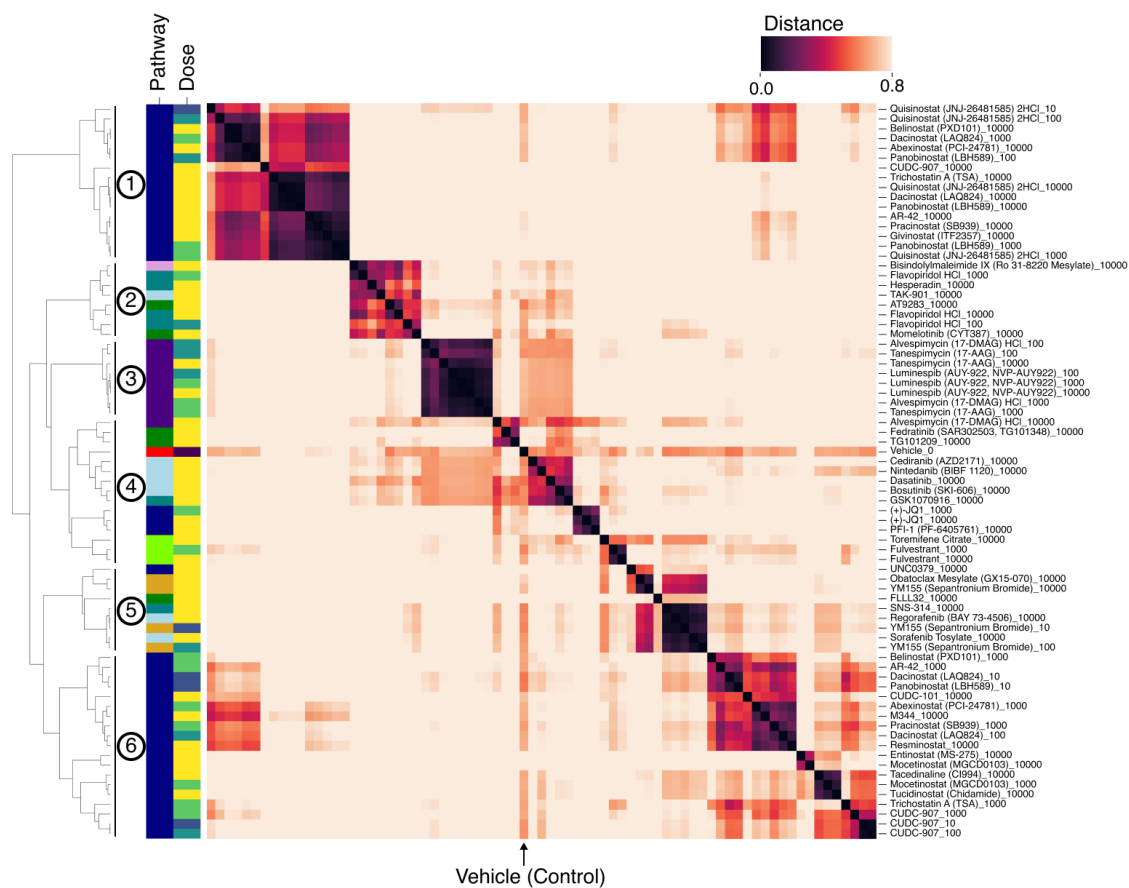


Figure 4.23: Copy of Figure 4.21e with row labels for each drug-dose combination.

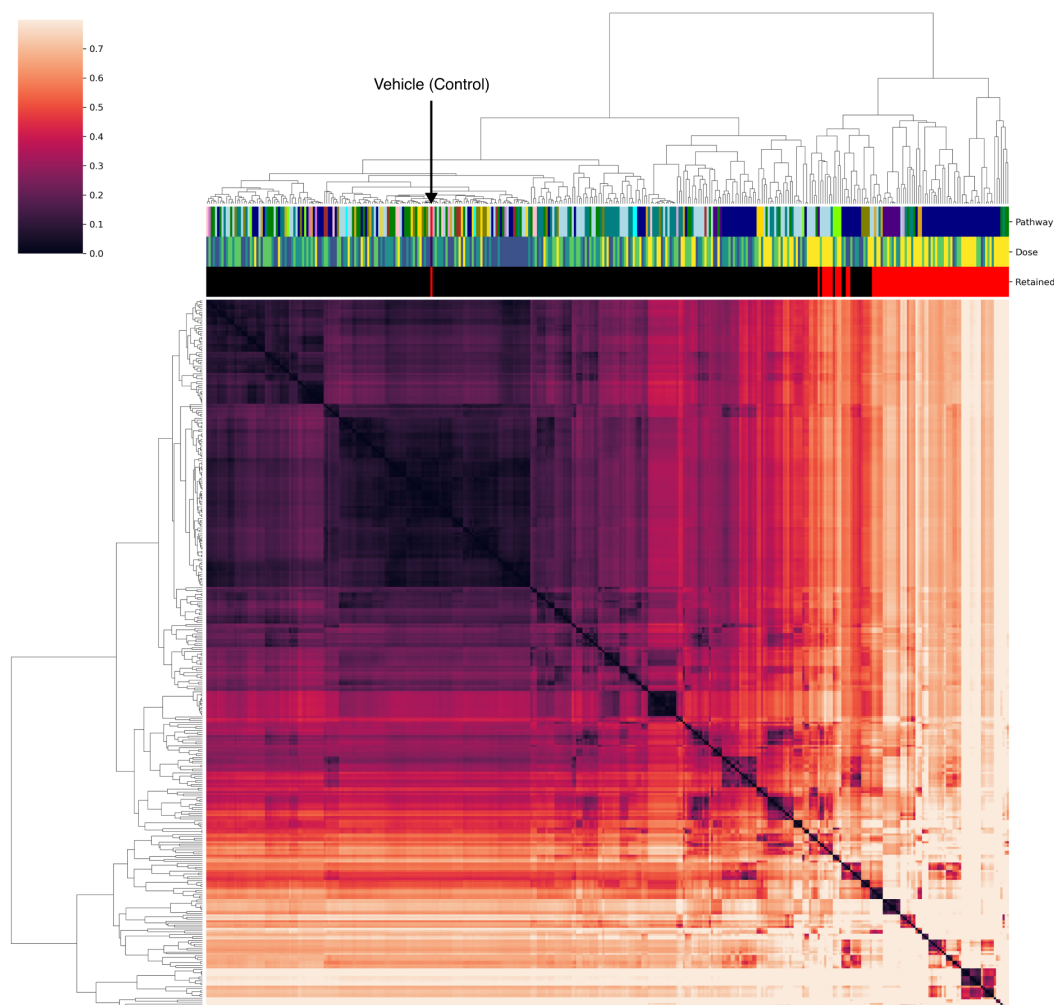


Figure 4.24: Sample distance matrix of all 369 samples (92 drugs at four doses and vehicle) used in the analysis for the MCF7 cell line. The columns are annotated by each drug's pathway annotation from the original study, dosage level, and whether the sample was retained for the remaining analysis (top 20 percent of samples based on distance from vehicle). The hierarchical clustering was performed with the Ward variance minimization algorithm.

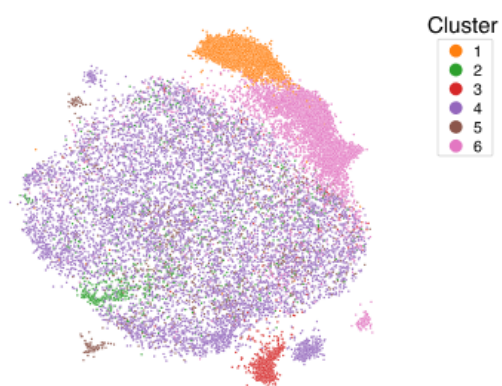


Figure 4.25: MDE of the z latent space from MrVI of data from the MCF7 cell line colored by Cluster as labeled in Figure 4.21e. Clusters ⑥ and ① capture a large group of samples with similarly divergent cell states corresponding to HDAC inhibitors.

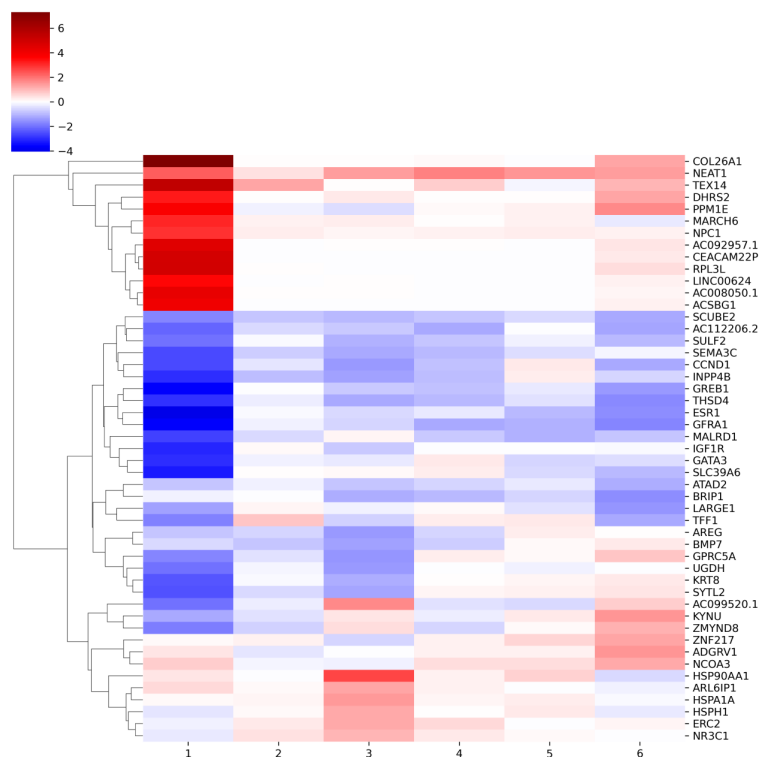


Figure 4.26: Heatmap of LFCs of the top DE genes identified by MrVI in the MCF7 cell line data. Each column corresponds to a cluster as labeled in Figure 4.21e. Displayed is the LFC estimated by MrVI averaged over all cells. Concordant with the distance matrix, we find the most significant and widespread gene-specific effects in cluster ①.

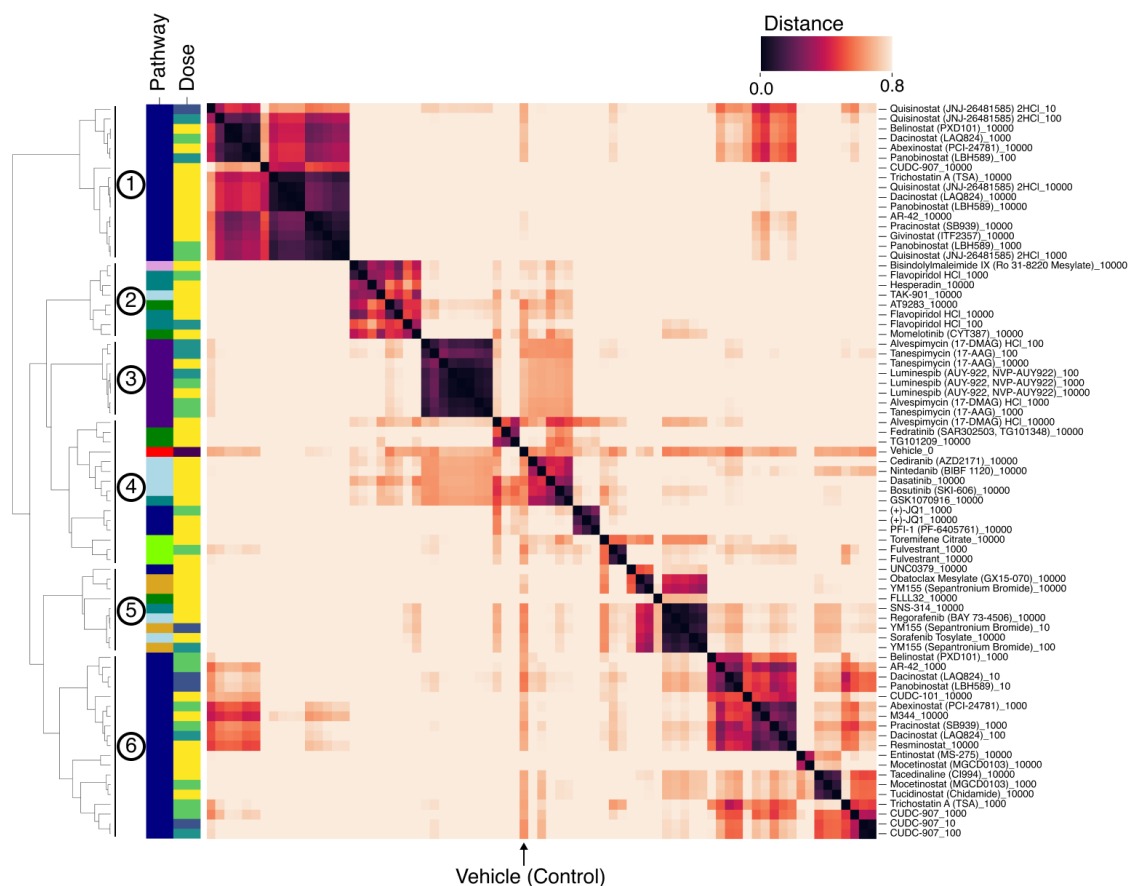


Figure 4.27: Copy of Figure 4.22e with row labels for each drug-dose combination.

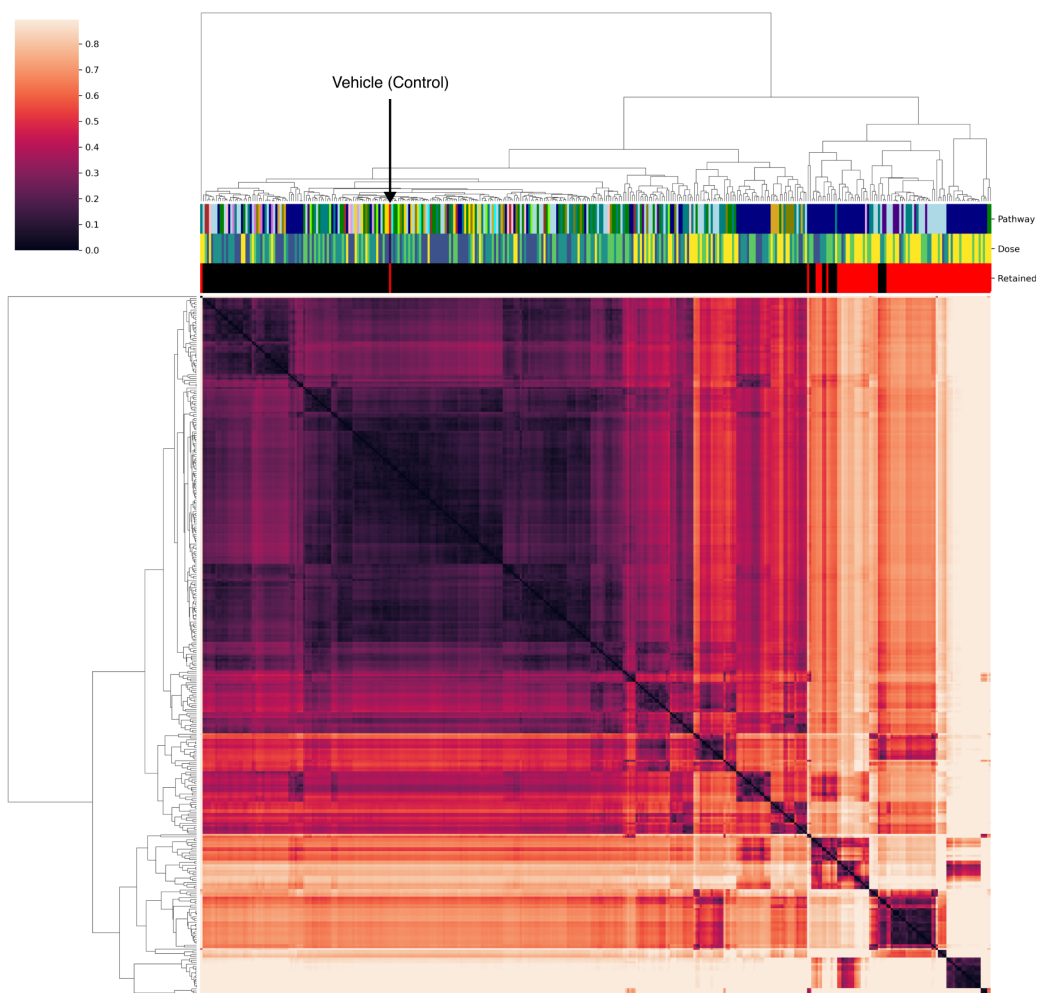


Figure 4.28: Sample distance matrix of all 369 samples (92 drugs at four doses and vehicle) used in the analysis for the K562 cell line. The columns are annotated by each drug's pathway annotation from the original study, dosage level, and whether the sample was retained for the remaining analysis (top 20 percent of samples based on distance from vehicle). The hierarchical clustering was performed with the Ward variance minimization algorithm.

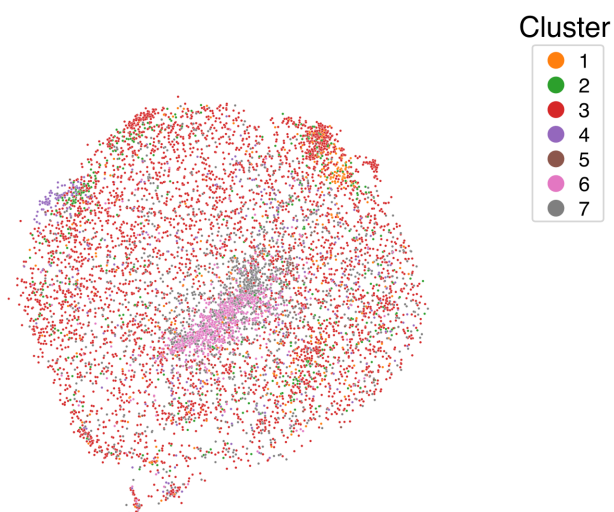


Figure 4.29: MDE of the z latent space from MrVI of data from the K562 cell line colored by Cluster as labeled in Figure 4.22e. We note that clusters ⑥ and ⑦ capture a large group of samples with similar transcriptomic responses corresponding to HDAC inhibitors.

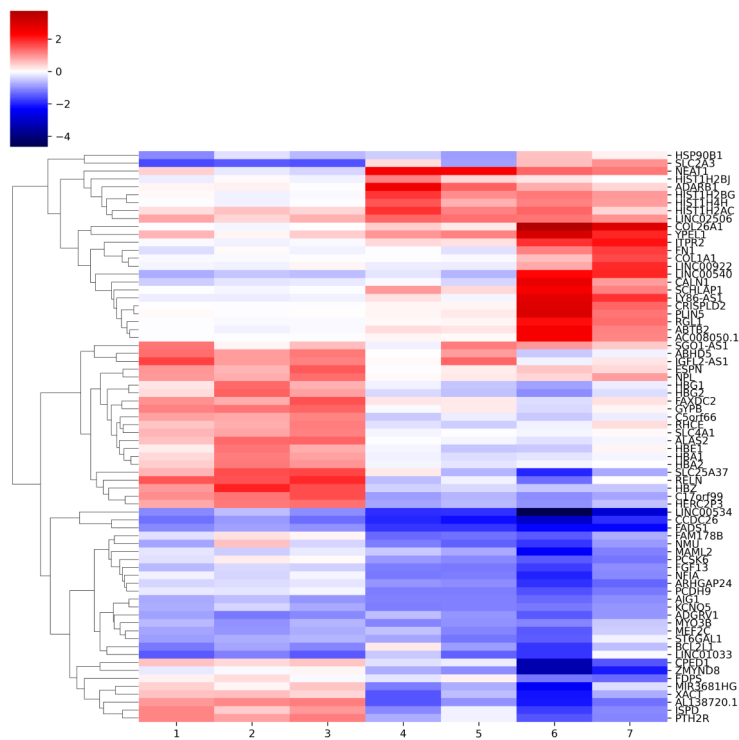


Figure 4.30: Heatmap of LFCs of the top DE genes according to MrVI over the K562 cell line data. Each column corresponds to a cluster as labeled in Figure 4.22e. Displayed is the LFC estimated by MrVI averaged over all cells. Concordant with the distance matrix, we find the most significant and widespread gene-specific effects in clusters ⑥ and ⑦.

Method	Bio conservation					Batch correction					Aggregate score		
	Isolated labels	KMeans NMI	KMeans ARI	Silhouette label	cLISI	Silhouette batch	iLISI	KBET	Graph connectivity	PCR comparison	Batch correction	Bio conservation	Total
MrVI(CT prior)	0.48	0.65	0.26	0.53	0.99	0.89	0.23	0.49	0.77	0.00	0.48	0.58	0.54
MrVI	0.41	0.62	0.26	0.53	0.99	0.87	0.42	0.56	0.68	0.00	0.50	0.56	0.54
SCVI	0.48	0.64	0.25	0.54	0.99	0.86	0.07	0.41	0.76	0.00	0.42	0.58	0.52
PCA	0.48	0.63	0.26	0.53	0.99	0.89	0.03	0.35	0.70	0.00	0.39	0.58	0.51

Figure 4.31: SCIB metrics computed on the IBD dataset. Tissue (colon or ileum) was used as batch key, and the original study’s annotations were used as cell-type labels as displayed in 4.34. MrVI (CT prior) uses the cell-type specific bias for the mixture weights using the existing annotations (**Methods**), while MrVI is the default version relying on mixture of Gaussians without cell-type biases.

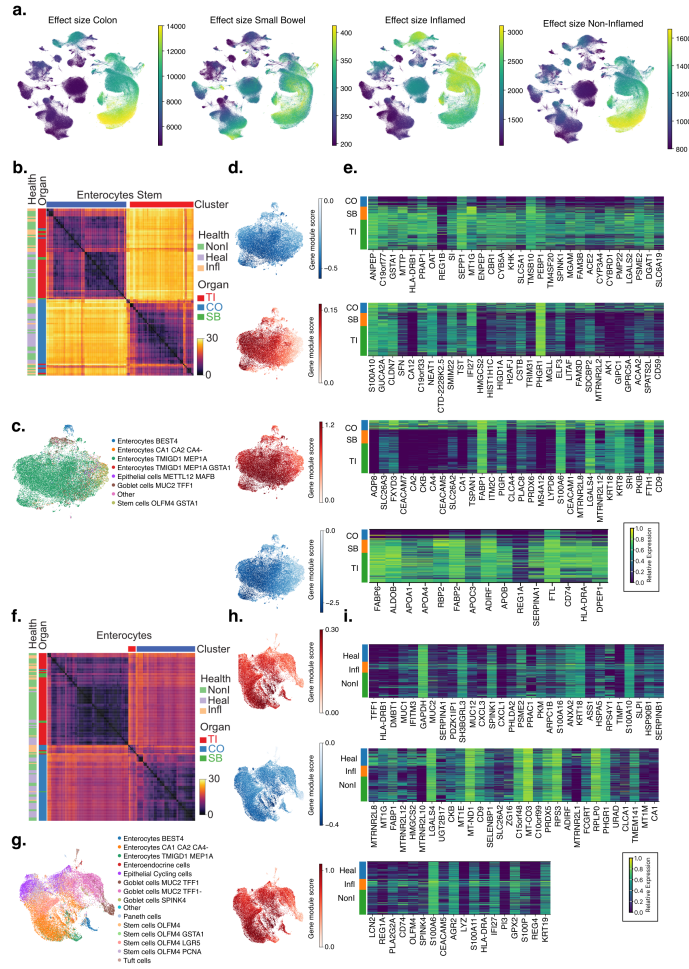


Figure 4.32: Unbiased analysis of the IBD dataset. **a.** We performed a guided analysis of differential gene expression (**Methods**). We computed the effect strength by computing the L2 norm of the covariate-specific vector in the z representation for each of the covariates (**Methods**). The effect size for the colon and small bowel use the ileum as reference, effect sizes for the inflamed/non-inflamed biopsies use the healthy biopsies as reference. **b.** Sample distance matrix of all enterocytes. **c.** UMAP embedding of all enterocytes according to MrVI u latent space. **d.** Scores of four gene modules identified between both highlighted clusters in the sample d matrix overlaid on UMAP plot. The score is the signed sum of all up- or downregulated LFCs within a module. **e.** Original expression after library-size normalization and log1p-transformation. **f-i** Same analysis as in **b-e** for *Enterocytes*.

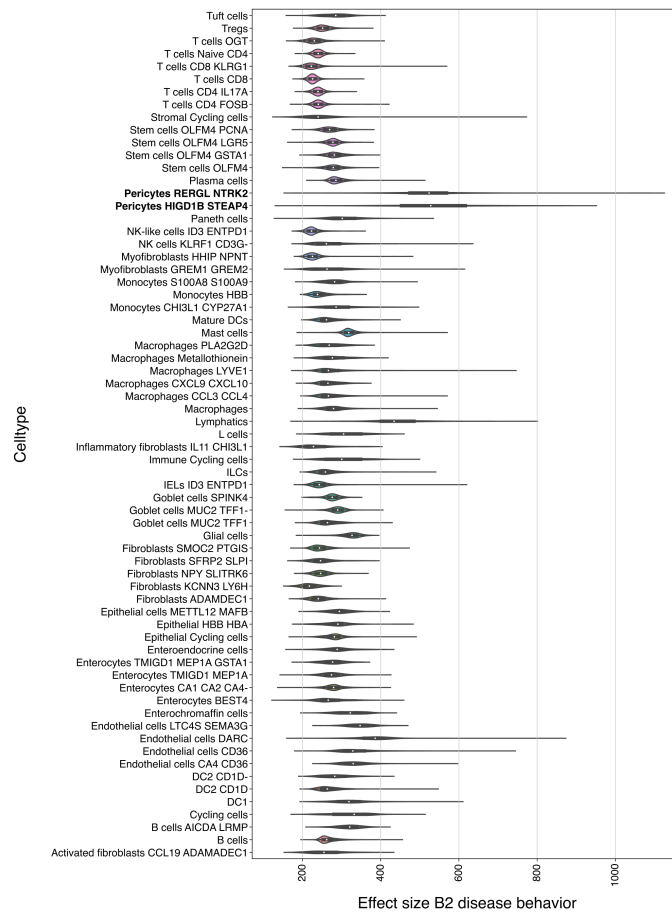


Figure 4.33: Violin plots of effect sizes of B2 disease behavior split by original cell-type labels. We quantify the estimated effect size displayed in **Figure 4.5a right**. Displayed is the violin plot for each cell type, and the plotted values are the estimated effect size for each cell. Highlighted in **bold** are both pericyte cell types that have the highest effect sizes.

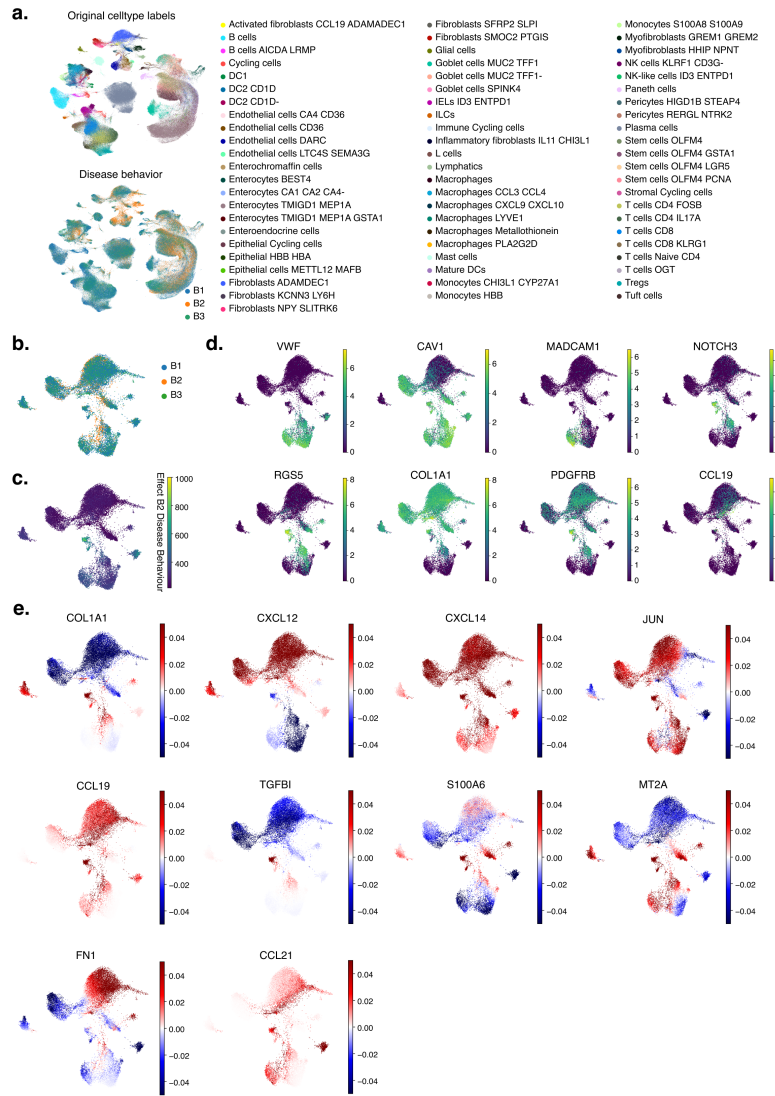


Figure 4.34: Additional analysis of stromal cells in stenosis. **a.** UMAP colored by all cell types in the original study as well as colored by disease behavior. **b-e** UMAP subset to stromal cells. **b.** colored by disease behavior. **c.** Same display as in **Figure 4.5a** of effect size of disease behavior B2 highlighting high score in a subset of pericytes. **d.** Gene expression for marker genes of endothelial-to-mesenchymal transition. **e.** Additional predicted LFCs in patients with stenosing course of disease (B2) versus patients with B1 disease behavior patients estimated by MrVI.

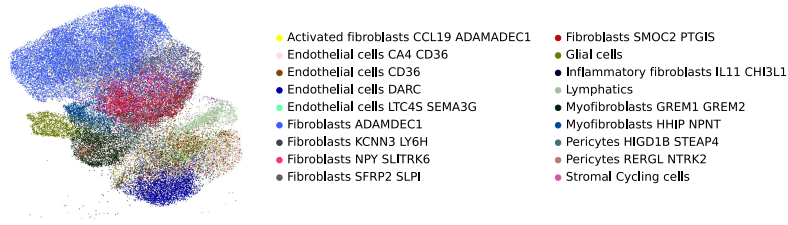


Figure 4.35: u latent representations of the cells displayed in Figure 4.5b, using the unsupervised setting of MrVI, instead of the supervised setting employed in the main text.

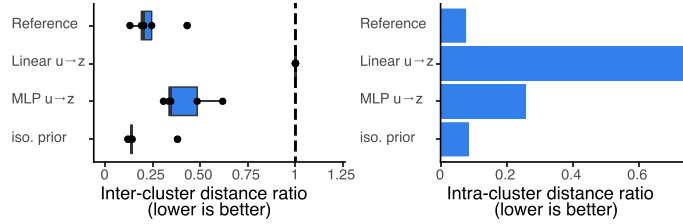


Figure 4.36: Ablation study of MrVI conducted on the semi-synthetic dataset. Multiple model configurations are compared based on the distance matrix quality scores of Figure 4.8. We compare MrVI with all its components (*Reference*) to ablated configurations Linear $u \rightarrow z$ and MLP $z \rightarrow u$ respectively denote MrVI implementations where the multi-head attention mechanism $g_{\theta}^{u \rightarrow z}$ from Equation (4.1) is replaced by a linear mapping and an MLP respectively. *iso. prior* denotes the use of a standard normal prior instead of a mixture of Gaussians prior for u . We observe that multi-head attention is a key component to ensure that MrVI properly retrieves the ground-truth sample effects.

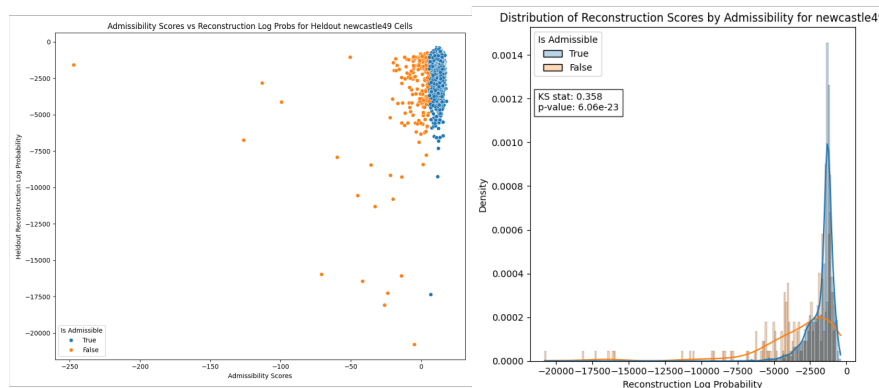


Figure 4.37: Admissibility scores vs. held-out reconstruction log probability densities for a subset of cells held out from the "newcastle49" sample of the COVID dataset [16]. (left) Scatterplot of admissibility scores vs. held-out reconstruction log probability densities. Each point is one cell, and the points are colored by whether they meet the threshold for admissibility based on their admissibility score. (right) Normalized histograms of held-out reconstruction log probability densities colored by admissibility category. Results of two-sided KS test between admissible and inadmissible points.

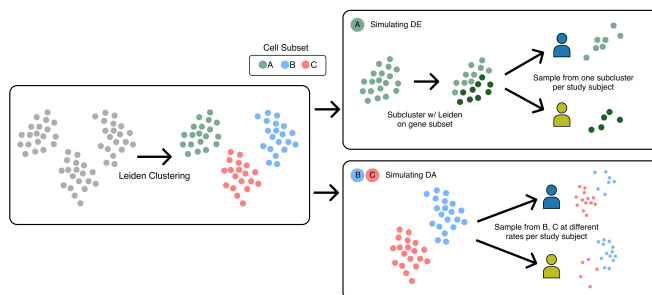


Figure 4.38: Cartoon of semi-synthetic benchmark construction presented in Figure 4.2. Data from a PBMC dataset of 68K cells [90] are clustered using the Leiden algorithm. Then, for cluster A, synthetic DE effects are introduced by assigning subclusters of cells to different study subjects. For clusters B and C, different proportions of each cell type are sampled depending on the study subject, producing a ground truth DA effect.

Supplementary Notes

This note contains additional information on preprocessing steps employed to conduct the different experiments of the paper. It also contains a user guide aimed to facilitate the use of MrVI for single-cell analysis, as well as sensitivity analyses to understand the impact of key parameters on model performance.

Data and preprocessing

Semi-synthetic experiment

We constructed a semi-synthetic dataset containing controlled DE and DA effects (**Figure 4.38**). Starting from a PBMC dataset of 68K cells [90], we generated a semi-synthetic scRNA-seq dataset containing a total of 32 study subjects. In particular, the original metadata has no importance in this experiment; we only relied on the gene counts to redefine synthetic notions of “cell-types”, which we will refer to as cell subsets and samples, which will here correspond to synthetic study subjects. The general recipe we employed to construct the semi-synthetic data from the original gene counts was the following. We first assigned each cell to one of five clusters, viewed as cell subsets, and then assigned study subjects to each cell in a way that introduced (1) DE between subject groups defined by *covariate 1* in cell subset A and (2) DA between subject groups defined by *covariate 2* in cell subsets B and C.

While we have no exact ground truth for the subject-subject distance in subset A, our ground truth consisted of a dendrogram, or tree, over study subjects, characterizing the similarities between subjects in terms of gene expression in the cell subset A. Specifically, all subjects sharing the same *covariate 1* value shared similar gene expression values for cells in subset A.

In two other cell subsets, denoted as B and C, cells had no differences in expression over subjects but exhibited differences in abundance, either corresponding to enrichment or depletion of these cell subsets in a specific group of samples. In particular, all subjects sharing the same *covariate 2* value shared similar proportions of cell subsets B and C but different proportions with the other subjects.

For all other cell subsets, we ensured that they neither had a DA nor a DE effect. The following are details of generating this semi-synthetic dataset.

Assigning cells to cell-types To assign cells to cell subsets, we clustered cells using the Leiden algorithm [214] on the log-median normalized counts and denoted the three most abundant clusters as A, B, and C. In this experiment, it is not important for these clusters to capture plausible cell types.

Introducing DE between samples in cell subset A Once cell subsets were defined, we assigned cells of subset A to study subjects to introduce DE between subjects. To achieve this, we stratified cells of subset A into subpopulations, each of which characterized a group of study subjects sharing the same expression profile for subset A. To do so, we picked 100 genes at random and performed hierarchical clustering on cells of subset A using these genes only, with a total number of eight clusters. The resulting dendrogram stratified cells of subset A into eight subpopulations, each of which contained cells that would be subsequently assigned to subjects with a shared category for *covariate 1*. To be precise, for each cluster and the associated group of subjects (each group consisting of four subjects) sharing a category for *covariate 1*, we assigned the cells uniformly at random between the four subjects. This strategy produced a total of $4 \text{ identifiers} \times 8 \text{ categories} = 32 \text{ subjects}$.

We relied on DESeq2 [33] to compute reference LFCs for the comparison of cells of subset A in the first subject to the rest of the subjects.

Introducing DA between subjects in cell subsets B and C We used a different subject assignment strategy in cell subsets B and C, specifically designed to introduce DA effects between subjects without introducing DE effects. To achieve this, we first assigned each cell of subsets B and C to one of the 32 subjects uniformly at random. Each subject was then assigned a depletion rate according to its value of *covariate 2* (one of four), which we denote r_s , that determined the rate of over-sampling in subset B and under-sampling in subset C. In other words, if we denote the original rate of sampling for cell subsets B and C as r^B and r^C , respectively, we then over-sampled cells from cell subset B in each subject s with probability $r^B + r_s$, and under-sampled cells from cell subset C in each subject s with probability $r^C - r_s$. This strategy ensured that the relative proportion of the other cell subsets in each subject remained constant while introducing DA effects between subjects in cell subsets B and C.

Subject assignments in the other cell subsets In the remaining cell subsets, subjects were assigned uniformly at random, effectively ensuring that these cells exhibited neither DE nor DA effects.

COVID experiment

Dataset & preprocessing We only retained cells from the Newcastle site from the original study [16], and retained the 10,000 most variable genes using Seurat v3. All throughout the experiments, we relied on the original study annotations, that were slightly simplified according to the following scheme: B_cell $\mapsto$ B cell, CD14 $\mapsto$ CD14 Monocyte, CD16 $\mapsto$ CD16 Monocyte, CD4 $\mapsto$ CD4 T cell, CD8 $\mapsto$ CD8

T cell, DCs $\mapsto$ DC, gdT $\mapsto$ gd T cell, NK_16hi $\mapsto$ NK, NK_56hi $\mapsto$ NK, pDC $\mapsto$ pDC, Plasmablast $\mapsto$ Other, Platelets $\mapsto$ Platelet, Treg $\mapsto$ Other, HSC $\mapsto$ Other, MAIT $\mapsto$ Other, Lymph_prolif $\mapsto$ Other, RBC $\mapsto$ Other, Mono_prolif $\mapsto$ Other, Lymph_prolif $\mapsto$ Other.

Model parameters The sample identifier used by MrVI corresponded to `patient_id`. The batch identifier was left empty. We used the same model hyperparameters as for the sci-Plex dataset, except for the following two differences. First, we used the described mixture of Gaussians prior ($K = 20$). Second, we set the dimensions of u and z to respectively be 5 and 30. We trained the model with minibatch sizes of 1,024 observations. MrVI was trained using early stopping with a patience of 30 epochs based on the validation ELBO.

Analysis Visualizations of MrVI latent variables relied on minimum distortion embeddings (MDE; [216]), applied with 15 neighbors and a repulsive fraction of 0.7. We identified different profiles of sample stratifications across cells by clustering cell-specific distance matrices with Leiden (resolution= 0.05), which returned three different cell clusters, one containing monocytes and dendritic cells, another one containing T cells, and the other one containing the rest of the cells. Once these clusters identified, we computed cluster-specific distance matrices by averaging cell-specific distance matrices across all cells of the cluster. We then used hierarchical clustering on these matrices to stratify donors using Ward’s method [217]. We then focused on the monocytes/dendritic cell cluster and studied DA and DE across different donor strata. We relied on Equation (4.4) to perform DA with MrVI. For differential expression, we compared high and low DSS patient groups, as identified in the distance matrix clustering analysis, using Equation (4.3). This analysis produced a cell-by-gene matrix of log fold-changes for the comparison of early to late patients, each row corresponding to a myeloid cell, and each column to a gene. We used this matrix to identify three modules of genes with potentially different patterns of DE across myeloids. For this purpose, we clustered genes by applying KMeans to the gene-by-cell matrix of log fold-changes after PCA dimensionality reduction.

sci-Plex experiment

Dataset The sci-RNA-seq3 dataset consists of RNA expression observations from the sci-Plex chemical perturbation screen over three different cell lines (A549, MCF-7, K562) [15]. This study involved 188 small-molecule drugs, each at four different doses (10nM, 100nM, 1000nM, 10000nM), as well as untreated control samples, referred to as "vehicle" samples. The assay was conducted in 96-well plates, with two

biological replicates for each drug-dose combination. We chose to concatenate the drug name and dose level as the modeled target covariate and the plate number as the modeled nuisance covariate. So, the two biological replicates mapped to each drug-dose combination were treated as one sample. Because each pair of biological replicates was conducted on different plates, any significant differences between the replicates were corrected for in the z_n latent space. The naming convention used in the figures for the sample covariate is "{drug name}_{dosage (nM)}".

Preprocessing Several steps were taken to preprocess the dataset. As discussed in [15], many of the small-molecule perturbations tested had little to no effect on the cell lines. To focus the analysis on the perturbations with significant effect, we performed a simple differential gene expression (DEG) analysis to filter out a subset of drugs with no effect and to simplify visualizations later for each cell line. For each cell line, we performed a t-test-based differential expression test for each drug-dose combination with respect to the vehicle in each cell line and adjusted the p-values with Benjamini-Hochberg correction. We defined a DEG as one with a p-value less than 0.05 and an absolute log-fold change (LFC) greater than 0.5. Then, we created a histogram of the number of DEGs for the maximum dose (10000nM) of each drug-cell-line combination (**Figure 4.14**). We chose 3,000 DEGs as a cutoff to capture the tail of this histogram, which should generally correspond to the combinations with significant, widespread differences from the vehicle cells. For the remaining analysis, we filtered out drugs that did not reach this cutoff for any dose-cell-line combination. Additionally, we tracked which drug-dose combinations reached the cutoff number of DEGs for each cell line for visualization purposes. The resulting dataset combining all three cell lines contained 251,088 cells with 92 drugs at all four doses and the vehicle cells. Last, we applied highly variable gene (HVG) selection using Seurat v3 [19] with the cell line as the batch key and retained the top 5,000 HVGs.

Model parameters MrVI was run separately on each of the three cell lines with the same set of hyperparameters. Since only one cell line was contained in the dataset for each model fit, we expected u_n to contain continuous variation corresponding to cell cycling effects but otherwise lack any variation corresponding to inter-cell-line differences. For this reason, a mixture of Gaussians prior was not used for u_n in these experiments. Instead, we assumed u_n to follow an isotropic Gaussian distribution ($K = 1$ with $\mu_1 = \vec{0}$ and Σ_1 as the identity matrix in Equation (4.1)). Otherwise, the model parameters mostly follow our recommended defaults, including the use of the MAP estimate on $z_n|u_n$, an isotropic Normal prior on $z_n - u_n$, and attention-based decoders. The dimensions of the u_n and z_n latent spaces were the

main hyperparameters that needed to be tuned accordingly. To ensure the model fit reflected prior knowledge about the sci-Plex dataset, we developed custom metrics to select the final hyperparameters. We discuss this in the following section.

Model selection We developed two dataset-specific metrics to determine our final model hyperparameters. First, we created a metric to reflect how similar the drug-dose combinations using the same drug perturbation were according to the model. Generally, we expected the effects of different dosages of the same drug to be more similar than the effects of two entirely different drugs. To capture this pattern for a given model fit, we computed the average percentile of the sample distances between same-drug-different-dose sample pairs relative to all pairwise sample distances (excluding distances to self). A lower value for this metric validated that the model was able to capture this general pattern without access to the drug-dose metadata. Second, we used L1000 bulk gene expression data [195] via the iLINCS platform [218] to validate drug-drug similarities determined by the models. Data for a subset of small molecules were available for the A549 and MCF7 cell lines. For each small molecule, we determined the set of DEGs using the criteria of p-values ≤ 0.05 and absolute log-fold change ≥ 0.5 using the differential expression data provided in the dataset. Then, we constructed a drug-drug similarity matrix for each cell line by computing the Jaccard similarity between the DEG sets of each drug. The Jaccard similarity is defined as $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$ for gene sets A, B . We clustered the drugs using the Leiden algorithm over the similarity matrix. Finally, we computed the rescaled silhouette score, $\tilde{s}(n)$ (Equation 4.7), of these clusters with respect to the subset of the sample distance matrix output by the MrVI model. In this case, a higher score implied the model’s sample distance matrix aligned more closely with the results of the L1000 gene expression profiles.

Lastly, we looked at the validation ELBO for each model, which describes how well the model generalizes to data it has not trained on. We found that optimizing over the validation ELBO alone did not provide good sample stratification according to the other metrics. This may be because, for smaller latent dimensions, the parameter space is smaller and thus can achieve a better ELBO by Bayesian Occam’s Razor [219]. Effectively, these smaller models tradeoff, modeling finer patterns in sample-specific variation for lower penalty attributed to the model prior. The other metrics provide a way to decide between the models with the best validation ELBO scores.

We tested different values for the dimensions of $\dim(u_n) \in [2, 5, 10, 30, 50]$ and $\dim(z_n) \in [2, 5, 10, 30, 50]$ where $\dim(u_n) \leq \dim(z_n)$ and found $\dim(u_n) = 10$ and $\dim(z_n) = 30$ had top values for all of the described metrics (**Figure 4.15**).

Analysis First, as a high-level check to see that u_n and z_n exhibited the expected variation, we visualized each space using a minimum distortion embedding (MDE) with a repulsive fraction of 1.5 and 15 neighbors. We colored the projection by cell cycle and drug class (as reported in the original dataset) (**Figure 4.4a,b**). The cell-cycle labels were computed using ScanPy’s `tl.score_genes_cell_cycle` function and the cell-cycle markers from [220].

Then, we computed the sample distance matrices and took the mean across all cells. Typically, one would perform the aggregation for subsets of cells exhibiting similar sample stratification (e.g., cell types). However, in this case, since each model was run over a single, relatively homogeneous cell type, the aggregation was performed over all cells. To ensure this aggregation was reasonable in a data-driven manner, we ran PCA over the learned sample distance matrices to check if there were groups of cells with distinct stratification (**Figure 4.4c**). We found that the top two PCs ($> 40\%$ of variation explained for each cell line) showed one cluster of cells for every cell line.

Then, we visualized the distance matrices reported by each model. To focus each visualization on the drugs with the most significant effects on each cell line, we filtered for the drug-dose-cell-line combinations in the top 20 percent based on distance from the vehicle controls. For example, this left 75 drug-dose combinations, including the vehicle, for the A549 cell line. Over this filtered matrix, we performed hierarchical clustering using Ward’s method. We then chose the number of clusters as roughly the elbow of the plot between the sum of squared differences and the number of clusters (**Figure 4.19**).

For each cluster found by the hierarchical clustering, we applied our model-based differential expression procedure for each cluster against the vehicle controls for the appropriate cell line. For each gene and each cluster, we averaged all the log-fold changes (**Figure 4.20**) then defined the DEGs as those exceeding an absolute average log-fold change of 1. We applied gene set enrichment analysis (GSEA) [198] using the MSigDB Hallmark 2020 gene set [143] separately over the set of up-regulated genes (genes with average LFC > 1) and the set of down-regulated genes (genes with average LFC < 1) and visualized the resulting scores for MSigDB gene sets where at least one cluster had a p-value < 0.05 (**Figure 4.4f**).

We presented the results of the A549 cell line in the main paper (**Figure 4.4**) and left the results of the remaining two cell lines, MCF7 and K562, to the supplement (**Figures 4.21, 4.22**). We include the analogous supplementary figures for MCF7 (**Figures 4.23, 4.24, 4.25, 4.26**) and K562 (**Figures 4.27, 4.28, 4.29, 4.30**) as well.

IBD experiment

Dataset & preprocessing We downloaded the data set from the Broad Single Cell Portal and concatenated the data sets in all organs and fractions. As the original dataset did not contain raw counts, we reverted the applied normalization in the following way. We transformed the normalized counts by `expm1`, divided them by 100,000 (original normalization), and multiplied by the original counts per cell. As the resulting gene expressions still contained small deviations from expected integer representations due to numerical inaccuracies, we rounded the values to the closest integer. We selected 10,000 highly variable genes using "`seurat_v3`" flavor in `scanpy` and using the respective 10X chemistry as batch key. We also filtered cells expressing more than 300 genes, 1K counts, or more than 20 mitochondrial reads, as these cells contained low-quality events.

Model parameters We used a cell-type prior for the mixture of Gaussian, set the dimensionality of z-space to 200 and of u-space to 10 and reduced `n_epochs_kl_warmup` to 25. For all other hyperparameters, we used default parameters. We set the `biosample_id` as the sample key and the concatenation of layer and chemistry as the batch key and trained the resulting model for 150 epochs.

Analysis For the UMAP embedding of all cells, we used 7 nearest neighbors and a minimum distance of 0.3. Sample distances were computed with normalized distances and using 20 Monte-Carlo samples. We used the upper-triangle of these distances and flattened those, followed by scaling these values and computed 50 PCs on the distance matrices. The resulting PCs were used for Leiden clustering (10 neighbors, cosine distance) with a resolution of 0.1 to yield the coarse cell-type labels displayed in **Figure 4.5a**. We annotated these coarse labels on the basis of the composition of the original labels in each coarse label.

The sample similarity matrices were displayed using `PyComplexHeatmap`. We subset the distance matrix to each coarse cell type and compute the mean distance. Furthermore, we filter all samples with an average admissibility below 0.3 using the out-of-distribution criterion with a quantile threshold of 0.05 (Equation (4.6)). The linkage and clusters were computed using `scipy.cluster.hierarchy` with `fcluster` setting it to 5 clusters and using the criterion `maxclust`. In `PyComplexHeatmap` we used `ClusterMapPlotter` without scaling and without row or column clustering.

To study sample stratification, we performed a multivariate analysis with batch-specific offsets, filtering donors with a quantile of 0.0, and an L1 regularization of 0.01. We computed 50 mc samples and computed differentially expressed genes for

each cluster identified from the sample distances, using the same approach as in the COVID experiment.

For further analysis, we excluded samples collected during surgery, as the experimental design of this study was rather imbalanced; the batch of surgical samples only included patients with stenosis, while the endoscopy biopsies contained a mix of patients with different disease subtypes. This imbalance introduces biases in downstream analysis because we found that surgical samples had significantly fewer immune cells and more stromal cells than biopsy samples (**Figures 4.5a**). To study stenosis, we reran the multivariate analysis on all cells from diseased individuals (excluding healthy controls and samples from surgeries) using Type (inflamed, non-inflamed), Site (colon, ileum), and Sex and Disease Behavior (B1, B2, B3) as covariates. We computed the DE genes as described in the previous paragraph and displayed the effect sizes of the multivariate analysis to identify cell populations strongly affected by covariates. For the UMAP plots of stromal cells, we used 20 nearest neighbors and a minimum distance of 0.3. We relied on Equation (4.4) to perform DA with MrVI. We filtered out all samples with less than 50 stromal cells as the DA estimates become noisy for low numbers of cells. Consequently, the Epithelial layer samples, which contained few stromal cells, were filtered out, leaving only the Lamina propria samples. To obtain a smooth estimate of differential gene expression, we compute normalized KNN connectivities (20 neighbors, normalized to a sum of 1) and multiply those normalized connectivities twice with the estimated log-fold changes. For violin plots and dotplots, we used the defaults in Scanpy. For the dotplots, we selected the top three marker genes in Scanpy using default setting for DE testing based on raw expression.

User guide

Model usage

MrVI is implemented within the `scvi-tools` Python package [53]. **Figure 4.39** provides a minimal working example illustrating how to use MrVI in a few lines of code. A tutorial and an user guide can be found within the `scvi-tools` documentation, respectively at https://docs.scvi-tools.org/en/latest/tutorials/notebooks/scrna/MrVI_tutorial.html and https://docs.scvi-tools.org/en/latest/user_guide/models/mrvi.html.

```

# adata: scanpy AnnData with raw counts in .X and sample IDs contained in .obs['sample_id']
from scvi.external import MRVI

MRVI.setup_anndata(
    adata,
    sample_key="sample_id",
    batch_key="batch_id", # optional, default: None
    cell_type_key="cell_type", # optional, default: None
)

model = MRVI(adata)
model.train()

latent = model.get_latent_representation() # u latent reps
dists = model.get_local_sample_distances(
    keep_cell=False, groupby="cell_type"
) # computes distances per cell type

cov_keys=["Status"] # Here, Status is the sample covariate of interest
de_res = model.differential_expression(sample_cov_keys=cov_keys)
da_res = model.differential_abundance(sample_cov_keys=cov_keys)

```

Figure 4.39: Snippet of code for the use of MrVI in a single-cell analysis.

Computational performance

To provide insight into the computational toll of MrVI, we present running times for the COVID-19 dataset, run on a custom-built workstation with one NVIDIA RTX 3090 GPU and a 12th Gen Intel i9-12900KF CPU.

Table 4.1: Execution times of MrVI on the COVID-19 dataset

Task	Time
Model fitting	8.28 min
Latents generation	8.91 sec
Distance matrices computation	39.37 min

These running times demonstrate that MrVI can process large-scale single-cell datasets in a reasonable timeframe with a modest computational setup. Furthermore, MrVI is a lightweight model, only requiring 800 MiB of GPU memory during inference.

Hyperparameter sensitivity analysis

Table 4.2: Integration metrics of MrVI on the COVID-19 dataset for supervised and unsupervised variants of MrVI computed on all three sites [24]. For each configuration, we report two SCIB batch correction scores. **Donor corr.** is computed using donor IDs as batch key. **Site corr.** is computed using site IDs as batch key. We also report the SCIB biological conservation score, using the study’s annotations as labels and the reconstruction loss of the model, computed on a held-out validation set randomly sampled from the original study. Here, *config 1* refers to the set of hyperparameters employed in the COVID-19 study (Figure 4.3). *config 2* refers to the hyperparameters employed in the IBD study (Figure 4.5)

	Donor corr.	Site corr.	Bio cons.	Val. Reco. loss
config 1 w/o cell-type prior	0.43	0.55	0.67	1204.07
config 1 w/ cell-type prior	0.44	0.55	0.68	1201.06
config 2 w/o cell-type prior	0.45	0.56	0.68	1187.50
config 2 w/ cell-type prior	0.46	0.56	0.70	1189.23

We studied the effect of key hyperparameters on MrVI performance

An important consideration for MrVI is the use of pre-existing cell-type annotations in the model. When available, MrVI can leverage such annotations to learn a cell-type-informed prior for u (Equation (4.5)). The analyses conducted in this work reflect different scenarios where reliance on cell-type annotations may or may not be relevant. In the IBD study, for instance, cell types from different lineages need to be integrated across tissue layers and preparation protocols, making a cell-type-specific prior beneficial to guide the integration. In contrast, since the COVID-19 study considered in this work was limited to a single site, the unsupervised variant of MrVI was successful at retrieving cell-type variations in the u latent space.

To appreciate the importance of the cell-type prior for integrating complex datasets, we considered the complete COVID-19 dataset and compared several configurations of MrVI in **Table 4.2**. In particular, we compared both supervised and unsupervised variants of MrVI, along with the different hyperparameter configurations employed in the main text. The configuration employed in the main text, benefiting from both cell-type annotations to refine the u prior and larger latent dimensionalities obtains the best integration metrics, both in terms of harmonization and biological conservation. The unsupervised configurations reached lower scores, suggesting that the cell-type prior is beneficial for the integration of complex datasets.

We also studied how other hyperparameters affected integration metrics (**Table 4.3**). MrVI was robust to changes in the considered parameter ranges, which suggests that the model allows for a certain degree of freedom in the choice of these parameters. At the same time, we found that the hyperparameters that produced the best integration metrics in the u space typically also had the lowest validation reconstruction loss.

Stability of distance matrices Last, we verified to what extent the distance matrices produced by MrVI were stable across different runs. For this purpose, we computed distance matrices for CD14+ monocytes for all model runs of **Tables 4.2 and 4.3**. We then computed the average Spearman correlations between all pairs of distance matrices. We observed a very high correlation between these matrices, with low standard deviations across runs (mean Spearman correlation: 0.976, standard deviation: 0.006). This observation highlights the robustness of MrVI for stratifying samples, not only across random initializations but also across different hyperparameter configurations.

Hyperparameter selection guidelines

Based on the results of the sensitivity analysis, we suggest the following guidelines for the choice of hyperparameters.

Cell-type annotation usage We suggest starting with the unsupervised variant of MrVI, and validate that the u latent representation properly integrates samples and technical factors, e.g., via computation of integration metrics [24]. If strong technical variations persist, e.g., separation of clusters by tissue type, sequencing batches, or experimental sites, we recommend using the supervised variant of MrVI, when high-quality cell type annotations are available.

Optimization of other hyperparameters For users with access to high-performance computing clusters, we recommend optimizing hyperparameters using grid search or Bayesian optimization on held-out reconstruction error. In the absence of such resources, we suggest using the default parameters used within the scvi-tools implementation of MrVI, which have demonstrated robust performance across multiple datasets in this study.

Table 4.3: Sensitivity analysis for key hyperparameters of MrVI computed on the COVID-19 dataset [24]. We studied three important model hyperparameters: the dimensionality of the z latent space (d_z), the dimensionality of the u latent space (d_u), and the prior choice for u . For each configuration, we report the batch correction, biological conservation, and total SCIB metrics (using the study’s annotations as labels and donor IDs as batch key). We also report the reconstruction loss of the model, computed on a held-out validation set randomly sampled from the original study.

Ablation	Parameter	Batch corr.	Bio cons.	Total	Reco. loss
z latent dim.	$d_z = 20$	0.42	0.66	0.57	1154.75
	$d_z = 30$	0.42	0.66	0.56	1153.90
	$d_z = 50$	0.42	0.68	0.58	1150.75
	$d_z = 100$	0.42	0.67	0.57	1151.72
	$d_z = 200$	0.42	0.64	0.55	1150.95
u latent dim.	$d_u = 5$	0.42	0.65	0.56	1150.09
	$d_u = 10$	0.43	0.66	0.57	1147.97
	$d_u = 20$	0.44	0.66	0.57	1140.17
u prior	Isotropic, $K = 1$	0.42	0.63	0.54	1154.28
	MoG prior, $K = 5$	0.42	0.66	0.56	1152.75
	MoG prior, $K = 10$	0.43	0.65	0.56	1147.62
	MoG prior, $K = 20$	0.42	0.67	0.57	1152.97

Chapter 5

AutoEval Done Right: Using Synthetic Data for Model Evaluation

5.1 Introduction

Our goal is to evaluate machine learning systems—assessing their accuracy, fairness, and other metrics—with as few data points as possible. This goal is important for reducing the human effort required to collect extensive validation datasets [164] for such tasks. Towards that end, we will explore an approach called *autoevaluation*, wherein we evaluate models in a two-stage procedure: (i). Produce synthetic labels using AI on a large unlabeled dataset, and (ii). evaluate AI models using the synthetic labels.

Autoevaluation can save months or years of time and potentially millions of dollars in annotation costs; see, e.g., Scale AI, or the recent work of Zheng et al. [221] using `gpt-4` to rank alternative language models’ answers to questions with high agreement with human annotators. However, the synthetic labels may not be trustworthy, especially for the purpose of certifying a model’s worst-case safety, multi-group accuracy and fairness, or to understand if observed differences between models are significant. This motivates the need for serious statistical inquiry on the general question of autoevaluation.

This work introduces methods for *autoevaluation done right*. Given a small amount of human data and a large amount of synthetic data, we will construct autoevaluation procedures that combine these datasets to get better estimates of performance.

In other words, our methods will increase the effective sample size of human data without compromising statistical validity. Intuitively, we use the limited human data in order to measure the bias of the synthetic data. Then, we evaluate the model on the

synthetic data and correct the bias using this estimate. The core statistical tool used for this debiasing is called prediction-powered inference (PPI) [48]; we will describe this tool in detail in the coming text. This approach can improve both metric-based evaluations (Section 5.2) and pairwise-comparison-based evaluations (Section 5.3), and can readily be applied using an existing Python software. We will include code snippets throughout for producing more precise unbiased evaluations. These lower-variance evaluations are also accompanied by confidence intervals.

Related Work

Autoevaluation has been a subject of interest, particularly in language modeling, well before the current wave of progress in machine learning [222–224]. Since the development of powerful machine learning systems such as `gpt-4`, the accuracy of the annotations that these systems produce has started to approach that of humans [221], giving substantial credence to autoevaluation as an alternative to human evaluations. The prohibitive cost of human annotation has also encouraged the development of automatic metrics used to evaluate model performance without human aid [225, 226], representing a distinct but related approach to autoevaluation. Automatic metrics can be computed on the fly, rely on more data points and are hence less noisy, which can be more informative than human evaluations when the latter are scarce [227]. Standard autoevaluation methods are generally ad hoc, and resulting estimates of model performance can systematically differ from those obtained by human evaluation [224]. In parallel, classical solutions for generating confidence intervals, such as rank-sets [228], cannot take advantage of the AI-generated data. It has been unclear how AI-generated data can be *combined* with human data to improve the quality of evaluations. Towards this end, [229] produced lower-variance estimates of machine translation performance by combining human preferences with automated metrics via control variates.

Prediction-powered inference (PPI) is a set of estimators that incorporate predictions from machine learning models [48] to get lower-variance estimators that remain unbiased. In our case, we employ an optimized variant, PPI++ [230], in order to estimate metrics using synthetic data. From a statistical perspective, PPI is closely related to the fields of multiple imputation and semiparametric inference, perhaps most notably the augmented inverse propensity weighting (AIPW) estimator [34, 231] (see [230] for a careful review). Indeed, we are not the first to notice this application of PPI; the work of Saad-Falcon et al. [232] describes an autoevaluation method for evaluating and ranking language models from pairwise comparisons for the purpose of retrieval-augmented generation. A preprint by Chatzi et al. [233], posted concurrently with ours, also considers the problem of ranking models from pairwise comparisons,

and constructs approximate rankings with coverage guarantees. Our approach is complementary to these existing works. Our specific contribution is to develop an instantiation of PPI that is practical and yields tight confidence intervals, is easy to implement using existing software, and is compatible with existing evaluation systems such as Chatbot Arena [234, 235]. Moreover, we evaluate our PPI method on real data. Along the way, we develop an interesting extension of the PPI algorithms to the case where the annotation model outputs not just a single synthetic Y , but a distribution over Y .

5.2 Autoevaluating Accuracy and other Metrics

We begin by describing how to use prediction-powered inference for estimating metrics. The most commonly used metrics are accuracy and loss, so we focus on these; however, our tools will be general and allow autoevaluation of any metric.

Defining the Goal

Basic notation We observe inputs X in some space $\mathcal{X}$, such as the space of natural images, natural language, and so on. We seek to predict labels Y in some space $\mathcal{Y}$, such as the space of classes, next tokens, actions, etc. Towards this end, let $f_1, \dots, f_M$ denote M pretrained models mapping inputs in $\mathcal{X}$ to label estimates in some third space $\hat{\mathcal{Y}}$. We often have $\hat{\mathcal{Y}} = \mathcal{Y}$, in which case the model directly outputs predictions of the label. However, we leave open the possibility that $\hat{\mathcal{Y}}$ is some other space—such as the space of softmax scores in the case of classification. The appropriate output space for $f(X)$ will be easy to infer from context.

Metrics We will evaluate the performance of the models by estimating the expectation of some metric function $\phi : \hat{\mathcal{Y}} \times \mathcal{Y} \rightarrow \mathbb{R}$; in other words, the *metric* of model m will be

$$\mu_m = \mathbb{E}[\phi(f_m(X), Y)] \quad (5.1)$$

for some metric function ϕ and every $m = 1, \dots, M$. We are interested in estimating the M -length vector $\mu = (\mu_1, \dots, \mu_M)$. For example, in the case of the accuracy, we would want to measure $\text{accuracy}_m = \mathbb{E}[\phi_{\text{acc}}(f_m(X), Y)]$ where $\phi_{\text{acc}}(y, y') = 1$ if $y = y'$ and 0 otherwise, for every $m \in 1, \dots, M$.

Accuracy is not the only quantity that can be framed within this setup. As another example, when the predictors are multilabel classifiers, one performance metric of interest could be the average precision of the model, that is, $\phi_{AP}(\hat{y}, y) := \frac{|\hat{y} \cap y|}{|\hat{y}|}$. In the case of regression, μ_m could correspond to the mean squared or absolute error of model m , in which case $\phi(\hat{y}, y) := (\hat{y} - y)^2$ or $\phi(\hat{y}, y) := |\hat{y} - y|$, respectively. Finally,

one can imagine estimating multiple losses at once; for example, for the purpose of assessing fairness, one may want to evaluate accuracy across many groups.

Data We assume access to two datasets: a small human-annotated dataset, $\{(X_i, Y_i)\}_{i=1}^n$, and a large amount of unlabeled data points, $\{X_i^u\}_{i=1}^N$, whose ground-truth labels $\{Y_i^u\}_{i=1}^N$ are unavailable. Importantly, both datasets are i.i.d.; extensions to some limited non-i.i.d. regimes are handled in [48], but we will not discuss them here. One should think of the regime where $N \gg n$: we have far more synthetic labels than real ones. For both datasets and every model, we also assume access to a *synthetic label distribution* that approximates $p(Y | X)$. We denote $\{\tilde{P}_{i,m}\}_{i=1}^n$ and $\{\tilde{P}_{i,m}^u\}_{i=1}^N$ as the set of synthetic label distributions conditioned on the labeled and unlabeled input data points, respectively. For each i and m , we will use the notation $d\tilde{P}_{i,m}(y)$ to represent the estimated PDF or PMF evaluated at label y .

For the sake of intuition, we make a few remarks regarding this data generating process. First, the synthetic data distributions can be seen as distributions over labels produced by one or several “annotator models”, that can either be related or different from the models to evaluate. In the latter case, the synthetic label distribution, for a given input, is the same for each model $f_1, \dots, f_M$. We do not need the subscript m in this scenario, and can simply denote the synthetic label distribution as $\tilde{P}_i$. However, the general case of $\tilde{P}_{i,m}$ allows for each model to have a different annotator model, and possibly allow models to self-annotate, that is, to themselves produce synthetic labels. Second, we note that the framework we described applies directly to the case where the annotator model produces single predictions of Y instead of distributions, by setting up $d\tilde{P}_{i,m}(y)$ to be a delta function at the prediction (the distribution is entirely concentrated on the prediction of Y).

The Algorithm

We combine the labeled and unlabeled data to estimate μ . In particular, we seek to benefit from the large sample size of the automatically annotated dataset to produce an estimator with low variance, while ensuring that this estimator remains unbiased. We will begin with the case of estimating accuracy, and then generalize our algorithm to arbitrary metrics.

Warm-Up: Model Accuracy

The classical approach to estimating model accuracy is to compute the fraction of correct labels:

$$\hat{\mu}_m^{\text{classical}} = \frac{1}{n} \sum_{i=1}^n \mathbb{1}(\hat{Y}_{i,m} = Y_i),$$

where $\hat{Y}_{i,m} = \operatorname{argmax}_y f_m(X_i)_y$ and $f_m(X_i)$ is the softmax output of model m . Instead, we propose estimating the accuracy of a classifier differently: by using the classifier's own confidence on the unlabeled data as a signal of its accuracy. Let $p_{i,m} = f_m(X_i)_{\hat{Y}_{i,m}}$ denote the top softmax score of model m on labeled example i , and $p_{i,m}^u$, $\hat{Y}_{i,m}^u$ be defined analogously. We will use the estimator

$$\hat{\mu}_m := \underbrace{\frac{\lambda}{N} \sum_{i=1}^N p_{i,m}^u}_{\text{accuracy}} + \underbrace{\frac{1}{n} \sum_{i=1}^n \Delta_{i,m}^\lambda}_{\text{bias correction}}, \quad (5.2)$$

where $\Delta_{i,m}^\lambda := \mathbb{1}\{\hat{Y}_{i,m} = Y_i\} - \lambda p_{i,m}$. Here, λ is a tuning parameter—for the time being, think of $\lambda = 1$. The above estimator decomposes into two natural components. Interpreting the top softmax score as the probability of correctness, the first term captures the model's internal estimate of its accuracy on the unlabeled data. The second term is the bias of the first term.

This estimator has two beneficial properties: *unbiasedness* and *variance reduction*. Unbiasedness means that $\mathbb{E}[\hat{\mu}] = \mu$. This implies that the inclusion of machine learning predictions in our estimator does not introduce systematic errors for estimating the accuracy. Variance reduction means that the use of synthetic data reduces the variance of our estimator: $\operatorname{Var}(\hat{\mu}_m) \leq \operatorname{Var}(\hat{\mu}_m^{\text{classical}})$.

This is formally true for the optimally chosen parameter λ ; indeed, the optimal choice of λ ensures that our estimator is always better than $\hat{\mu}^{\text{classical}}$ (in an asymptotic sense). See [230] for details and a formal proof; refer to Supplement 5.5 to see how to compute this estimator in Python.

General Metrics

The approach we have presented for evaluating classifier accuracy is an instance of a more general framework for evaluating properties of machine learning models. In particular, we can use our annotator model to output an approximate expectation of each label $\{Y_i\}_{i=1}^n$ and $\{Y_i^u\}_{i=1}^N$ in the following way:

$$\begin{cases} \hat{\mathbb{E}}_{i,m} &= \int_{y \in \mathcal{Y}} \phi(f_m(X_i), y) d\tilde{P}_i(y) \\ \hat{\mathbb{E}}_{i,m}^u &= \int_{y \in \mathcal{Y}} \phi(f_m(X_i^u), y) d\tilde{P}_i^u(y). \end{cases} \quad (5.3)$$

These expressions look complicated, but have a simple interpretation: the annotator model, given X_i , thinks the distribution of Y_i is $d\tilde{P}_i$, and we are simply calculating the expected metric under that estimated distribution. This explains the hats on the expectation symbols; these are not real expectations, but rather, estimated

expectations according to the annotator model. Indeed, in the case of classification, we see that, as is intuitive, the expected accuracy of the m th model on the i th data point is equal to its top softmax score:

$$\hat{\mathbb{E}}_{i,m} = \int \phi_{\text{acc}}(\hat{Y}_{i,m}, y) d\tilde{P}_{i,m}(y) = d\tilde{P}_{i,m}(\hat{Y}_{i,m}) = p_{i,m}.$$

Along the same lines, our previous estimator can be generalized to the case of arbitrary metrics as

$$\hat{\mu}_m := \underbrace{\frac{\lambda}{N} \sum_{i=1}^N \hat{\mathbb{E}}_{i,m}^u}_{\text{metric on synthetic data}} + \underbrace{\frac{1}{n} \sum_{i=1}^n \Delta_{i,m}^\lambda}_{\text{bias correction}}, \quad (5.4)$$

where now $\Delta_{i,m}^\lambda := \phi(f_m(X_i), Y_i) - \lambda \hat{\mathbb{E}}_{i,m}$. The first sum in the above expression is the average metric predicted by model m over all synthetic labels. If the annotator model is near-perfect and N is large, then this term will almost exactly recover the metric. However, if the synthetic label distribution is not good, this can bias our estimate of the metric. The second term corrects this bias by calculating it on the labeled dataset and subtracting it off.

Returning to the role of the tuning parameter: $\lambda \in [0, 1]$ is a discount factor on our synthetic data. When the synthetic data is very good, we can set $\lambda = 1$; when it is bad, setting $\lambda = 0$ will throw it away entirely. One can asymptotically optimize the variance of $\hat{\mu}$ in order to set λ , as in [230].

Again, it is straightforward to see that for any fixed value of λ , our estimator in (5.4) is unbiased, meaning $\mathbb{E}[\hat{\mu}] = \mu$, and will be strictly lower-variance than its classical counterpart when λ is optimally chosen.

Variance and Confidence Intervals

As we have explained above, the main benefit of AutoEval is to reduce the number of human-labeled data points to achieve a particular variance. We can formalize this by analyzing the variance of $\hat{\mu}_m$ and $\hat{\mu}_m^{\text{classical}}$. In particular, we can write the covariance matrix of $\hat{\mu}$ as

$$\frac{1}{n} V = \frac{1}{N} \lambda^2 \text{Cov}(L_i^u) + \frac{1}{n} \text{Cov}(\Delta_i^\lambda),$$

where $\Delta_i^\lambda := (\Delta_{i,1}^\lambda, \dots, \Delta_{i,M}^\lambda)$. This expression admits a simple plug-in estimator; it also indicates that we should pick λ to minimize V in the appropriate sense. It also allows for the production of non-asymptotic confidence intervals using concentration.

We opt to use asymptotic confidence intervals for μ . In particular, we have that as n and N approach infinity,

$$\sqrt{n}\widehat{V}^{-1/2}(\hat{\mu} - \mu) \rightarrow \mathcal{N}(0, \mathbb{I}_M),$$

where $\widehat{V}$ is the plug-in estimator of V , corresponding to

$$\widehat{V} = \frac{n\lambda^2}{N} \widehat{\text{Cov}}(L_i^u) + \widehat{\text{Cov}}(\Delta_i), \quad L_i^u = (\hat{\mathbb{E}}_{i,1}^u, \dots, \hat{\mathbb{E}}_{i,M}^u).$$

Note that when $\lambda = 0$, we exactly recover $\hat{\mu}^{\text{classical}}$ —but this may not be the parameter that minimizes the variance $\widehat{V}$. Indeed, we can explicitly choose λ to minimize the variance. An explicit expression for this estimate can be found in [230].

Another beneficial aspect of the asymptotic analysis is that it allows us to construct confidence intervals with which we can reliably rank models. For example, coordinatewise, the following is an asymptotically valid $1 - \alpha$ confidence interval *marginally* for each $\hat{\mu}_m$:

$$\mathcal{C}_m = \left(\hat{\mu}_m \pm \frac{z_{1-\alpha/2} \widehat{V}_{m,m}}{\sqrt{n}} \right). \quad (5.5)$$

The above interval comes with the following (standard) guarantee for all $m = 1, \dots, M$: $\lim_{n,N \rightarrow \infty} \mathbb{P}(\mu_m \in \mathcal{C}_m) = 1 - \alpha$.

As an alternative to producing confidence intervals for a single coordinate μ_m based on Equation (5.5), we might want to create confidence sets that contains the entire vector μ , that is, simultaneously valid intervals. The simultaneous interval can be constructed using the chi-squared distribution as

$$\mathcal{C}^x = \left\{ \mu : n \left\| \widehat{V}^{-1/2}(\hat{\mu} - \mu) \right\|_2^2 \leq \chi_{1-\alpha, M}^2 \right\},$$

where $\chi_{1-\alpha, M}^2$ denotes the $1 - \alpha$ quantile of the chi-squared distribution with M degrees of freedom. This interval has the following (standard) guarantee:

$$\lim_{n, N \rightarrow \infty} \mathbb{P}(\mu \in \mathcal{C}^x) = 1 - \alpha,$$

and thus, it can be used to rank the models by checking whether the m and m' coordinates of $\mathcal{C}^x$ overlap for each model m and m' in $1, \dots, M$.

Application to Rank Computer Vision Models

We applied the described methodology applies for evaluating computer vision models. We considered five trained computer vision models (ResNet-18, ResNet-34,

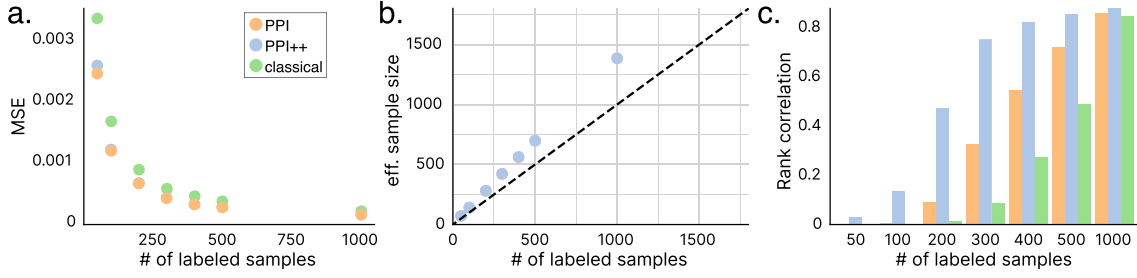


Figure 5.1: **ImageNet experiment.** For every approach, we built confidence intervals around the average accuracy of different ResNet architectures. **a.** MSE of the point estimates of the model accuracies. **b.** ESS of *PPI* and *PPI++* against the classical approach. **c.** Correlation between the estimated and true model rankings. Here, and in all following figures, obtained metrics are averaged across 250 random splits of the validation data into labeled and unlabeled data.

ResNet-50, ResNet-101, and ResNet-152) optimized over the training set of ImageNet and sourced from PyTorch [236]. We considered the task of estimating their accuracy on the validation set of ImageNet in a low-data regime, using a subset of labeled data points. The ground-truth model accuracies were computed as the mean accuracies evaluated over the entire validation dataset.

We considered two different approaches to estimate the accuracy of these models. The first is referred to as *PPI* [48], and corresponds to (5.2) with $\lambda = 1$. The second strategy, *PPI++* [230] optimizes λ to minimize the variance. These approaches were benchmarked against $\hat{\mu}^{\text{classical}}$ along with a standard z-test confidence interval.

To reflect a low-data regime, we randomly sampled a small number n of observations to be used as labeled data points available for these approaches. The rest of the observations in the validation data were used as unlabeled data points for *PPI* and *PPI++*. Our synthetic label distribution $d\tilde{P}_{i,m}$ is the softmax vector of model m on labeled data point i ; $d\tilde{P}_{i,m}^u$ for an unlabeled data point is analogous.

The mean-squared error of our estimates of the model accuracies improved over the classical baseline (Figure 5.1a). Both *PPI* and *PPI++* had lower mean-squared errors than the baseline, no matter the size of the labeled set. Little to no difference was observed between *PPI* and *PPI++*, which probably means that the imputed accuracy scores are reliable proxies for the true quantities. Our approach hence provided more accurate point estimates of the model accuracies. When uncertainty quantification does matter, *PPI* and *PPI++* provided calibrated confidence intervals across all labeled set sizes, and produced tighter confidence intervals than the baseline (Supplement 5.7).

The benefit of using unlabeled data can be measured by computing the effective

sample size (ESS) of PPI and PPI++ relative to the classical approach (Figure 5.1b). This value can be interpreted as the equivalent number of labeled data points for the classical approach that would be required to achieve the same level of precision as PPI or PPI++. Our ESS exceeds that of the classical approach by approximately 50%, which demonstrates the utility of unlabeled data for evaluating model performance.

Here, and in the other experiments, we also evaluated our approach for the purpose of model ranking, by ranking models based on their confidence intervals after Bonferroni correction. Models with overlapping confidence intervals were considered tied. Figure 5.1c shows the correlation of the estimated model ranks with the ground truth ranking (computed on all data) for different n and averaged across labeled-unlabeled data splits. This experiment showed dramatic differences between the approaches. PPI++ showed much stronger correlations with the ground truth than the other approaches, meaning that its rankings were more accurate and less prone to ties.

Application to Evaluate Protein Fitness Prediction Models

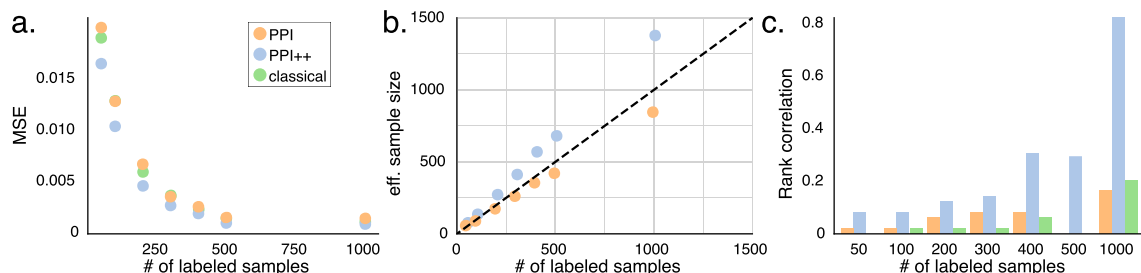


Figure 5.2: **Protein fitness experiment** for building confidence intervals and point estimates for the Pearson correlation of seven protein language models with the experimental fitness scores, using a held-out model to produce synthetic labels. **a.** MSE of the point estimates of the model correlations. **b.** ESS of *PPI* and *PPI++* against the classical approach. **c.** Correlation between the estimated and true model rankings.

We also used AutoEval to rank regression models, and more specifically, protein fitness prediction models. Protein fitness prediction is a crucial task in computational biology, aiming to predict the biological relevance of protein mutations based on their amino acid sequences. The recent development of deep learning models for protein language modeling has enabled the emergence of powerful models, trained on millions of protein sequences, used to predict protein fitness in a zero-shot manner [237].

Unfortunately, evaluating these models for a specific task remains challenging due to the scarcity of experimental data that can be used for evaluation, typically requiring expensive, time-consuming, and poorly scalable wet-lab experiments [238, 239].

We applied AutoEval on ProteinGym [240], which gathers several assays containing both experimental fitness measurements, used as ground-truth labels, and predicted fitness scores from various fitness predictive models. We focused on ranking protein language models for predicting the fitness of mutations in the IgG-binding domain mutations of protein G based on an assay of $N = 536,962$ pairwise mutations [241].

We considered a scenario where one aims to select the best model for zero-shot fitness prediction for a specific protein, using a small experimental dataset and a large set of potential mutations for which fitness is not measured. We focused on the Pearson correlation between predicted and experimental fitness scores as a validation metric for ranking models. More specifically, we aimed to estimate the metric $\mu_m = \mathbb{E}[Y f_m(X)]$, where Y, X, f_m are the experimental fitness, the protein sequence, and the m -th fitness predictor, respectively, assuming that Y and $f_m(X)$ have zero mean and unit variance. This fits in our general metric evaluation framework, where the metric function in Equation (5.1) is $\phi(y, y') = yy'$.

We used VESPA [242], a protein language model, as an (held-out) annotator model for synthetic label generation. VESPA produce point estimates of the experimental fitness scores $f_{\text{VESPA}}(X_i)$ based on the protein sequence X_i aiming to approximate the experimental fitness scores Y_i . While VESPA does not provide uncertainty estimates for its predictions, we can still use it as an annotator model in our framework. Specifically, we applied our estimator to estimate model m 's Pearson correlation with the experimental fitness scores, by setting $\mathbb{E}_{i,m}^u = f_m(X_i^u) f_{\text{VESPA}}(X_i^u)$ for the synthetic term and $\Delta_{i,m}^\lambda = f_m(X_i) Y_i - \lambda f_m(X_i) f_{\text{VESPA}}(X_i)$ for the bias correction term in Equation (5.4).

The results of this experiment are shown in Figure 5.2. The effective sample sizes of PPI++ were systematically higher than the classical approach (Figure 5.2b), by approximately 50%. Furthermore, the ranks obtained by our approach were also much closer to the true model ranks than the classical approach (Figure 5.2c), with a five-fold improvement for $n = 1000$.

PPI++ confidence intervals for models' correlations with the experimental fitness scores were also slightly tighter than the classical approach, yet remained calibrated (Supplement 5.7). The PPI estimator performed worse than the classical approach. This is a known issue of this estimator, that PPI++ mitigates.

A question remains: how good does the annotator model need to be to allow AutoEval to work well? Figure 5.3 compares the effective sample size of PPI++ obtained with different annotator models. As expected, the better the annotator model, the higher the effective sample size. We importantly note that even with a

very poor annotator model, PPI++ performs at least as well as the classical approach. When the annotator labels do not correlate with the true labels, PPI++ falls back to the classical approach ($\lambda = 0$), effectively ignoring the synthetic labels. That being said, we observe that even mediocre annotator models, such as CARP, provide a 10% increase in effective sample size compared to the classical approach. Altogether, these observations suggest that AutoEval can provide better point estimates and tighter confidence intervals compared to the classical approach even when the annotator model is mediocre.

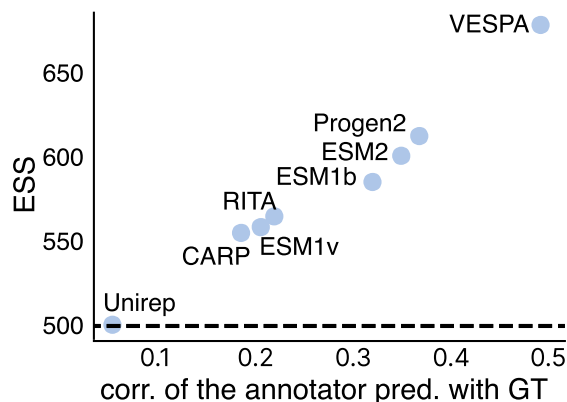


Figure 5.3: ESS of PPI++ against annotator model performance for $n = 500$ labeled points for the protein fitness experiment. The horizontal line denotes the ESS of classical.

5.3 Evaluating Model Performance from Pairwise Comparisons

Characterizing the absolute performance of ML models for the purpose of ranking them is challenging. The previous section described a methodology to compare models based on a common performance metric. Unfortunately, metrics serving as proxies for model performance might either not exist, or diverge from human judgment [243].

In such cases, assessing *relative* model performance might be more appropriate. This can typically be done by comparing different model predictions to each other. The Chatbot Arena project [234], for instance, allows human annotators to state preferences over different LLM predictions to the same prompt. Comparison-based evaluation is also an exciting opportunity for autoevaluation [221, 244]. In particular,

an external LLM, prompted to serve as an annotator, agrees with human annotators with high fidelity. Still, it is unclear how biased an AI annotator might be, which drastically limits the usefulness of the validation data it produces. [244] conducted studies into biases within AI annotators, such as stylistic and model-specific biases, highlighting the need for more robust inference. This section describes how to leverage such AI-generated preferences while making statistically valid inferences about model performance.

A Model to Assess Relative Performance

The canonical model for assessing relative performance of models based on pairwise comparisons, as in a tournament, is called the Bradley-Terry (BT) model [245–247]. The BT model is used in the Chatbot Arena [235], by the World Chess Federation, the European Go Federation, and many other competitive organizations as a tool for ranking players.

Now we describe the BT model. Imagine, among M models, we are trying to compare the strength of model A to the strength of model B . Towards this end, we give a prompt Q to both models, and they give us an answer. We show this answer to a human, who gives us $Y = 1$ if the answer of model B is better than the answer of model A , and vice versa. The assumption of the BT model is that Y follows a logistic relationship,

$$P_{\zeta}(Y = 1 \mid A, B) = \frac{1}{1 + e^{\zeta_A - \zeta_B}},$$

with some parameter vector ζ of length M , whose entries are called the *Bradley-Terry coefficients*. Each model m has a BT coefficient ζ_m which, when large relative to the other coefficients, signifies that it is more likely to win the pairwise comparison. (Also, because the model in (5.3) is invariant to addition of a constant to every coordinate of ζ , we can, without loss of generality, set $\zeta_1 = 0$, making the model identifiable.)

It is well-known that, given a labeled dataset of n pairwise comparisons, $\{A_i, B_i, Q_i, Y_i\}$, the maximum-likelihood estimator of the BT coefficients is a logistic regression [248]. Let X_i be the vector of all zeros except at indexes A_i and B_i , where it is -1 and 1 respectively. The logistic regression estimate of the BT coefficients is

$$\hat{\zeta}^{\text{classical}} = \underset{\zeta \in \mathbb{R}^{M-1}, \zeta_1=0}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n \ell_{\zeta}(X_i, Y_i),$$

where ℓ is the binary cross-entropy loss.

Autoevaluation of Relative Performance

Prediction-powered inference can be applied out-of-the-box to the BT model, making it possible to leverage large numbers of AI-generated preferences while controlling for their potential bias. In addition to the set of human preferences defined above, additionally define the unlabeled dataset $\{(A_i^u, B_i^u, Q_i^u)\}_{i=1}^N$. On both the labeled and unlabeled datasets, we have the prompt and both models' answers; we use a LLM in place of the human to choose between the answers. This gives us a prediction $\hat{Y}_i$ and $\hat{Y}_i^u$ of the pairwise comparison on both datasets. The PPI++ estimator of the BT coefficients is given by

$$\begin{aligned} \hat{\zeta} = \operatorname{argmin}_{\substack{\zeta \in \mathbb{R}^{M-1} \\ \zeta_1=0}} \frac{1}{n} \sum_{i=1}^n \left(\ell_{\zeta}(X_i, Y_i) - \lambda \ell_{\zeta}(X_i, \hat{Y}_i) \right) \\ + \frac{\lambda}{N} \sum_{i=1}^N \ell_{\zeta}(X_i^u, \hat{Y}_i^u), \end{aligned} \quad (5.6)$$

where $\lambda \in [0, 1]$ controls the weight we give to the AI-generated preferences. Although this estimator departs from the arguments given in Section 5.2, it has a very similar interpretation; it constructs an unbiased and lower-variance loss function for the true logistic regression, and then minimizes it.

The resulting BT coefficient estimates have the same appealing properties as above. In particular, they are unbiased for any fixed λ , and one can construct valid confidence intervals around them using PPI and PPI++; see [48, 230] for this and other generalized linear models, as well as methods for optimally choosing λ . Supplement 5.5 describes how to compute this estimator easily in Python.

Autoevaluation of LLMs from Pairwise Preferences

We evaluated our approach on the Chatbot Arena project [235]. We first extracted 16K observations from the Chatbot Arena dataset, in which a total of 20 recent LLMs were compared (Table S1). Each observation contains a prompt written by a human, responses from two of the 20 LLMs, and the preference of the human over the two responses. For each observation, we used `gpt-4o-mini` as a judge [221] by prompting it to identify the most useful response, following the same prompting approach as [244]. We focused on scenarios where only a few of the 16K human preferences were available, and compared our approach (using both available human preferences and all `gpt-4o-mini` preferences) to the classical approach.

Results are shown in Figure 5.4. We observed that the BT coefficients were better estimated by PPI++ than by the classical approach, hinting that the point estimates

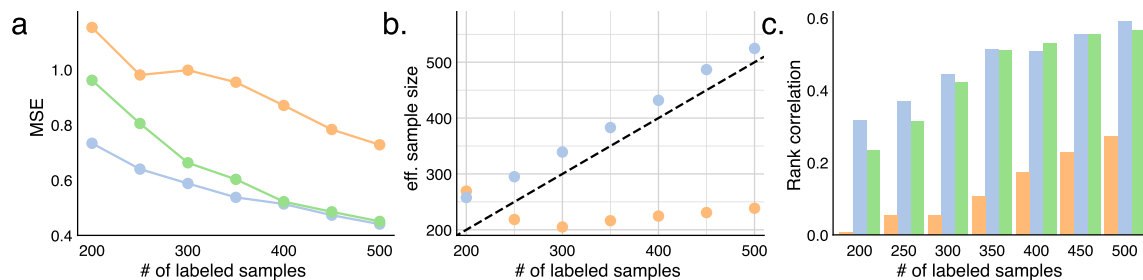


Figure 5.4: **LLM experiment** for building confidence intervals and point estimates for the BT coefficients of different LLMs. **a.** MSE of the point estimates of the BT coefficients. **b.** ESS of PPI and PPI++ against the classical approach. **c.** Correlation between the estimated and true model rankings.

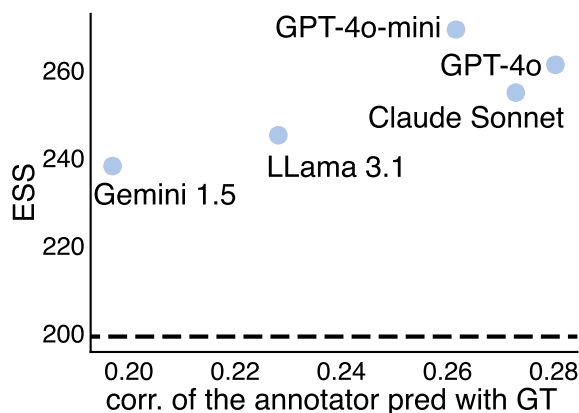


Figure 5.5: ESS of PPI++ against annotator model performance for $n = 200$ labeled points in the LLM experiment. The horizontal line denotes the ESS of classical.

of AutoEval are more accurate (Figure 5.4a). We also observed ESS showing a 20% to 25% improvement over the classical approach (Figure 5.4b). Finally, we observed that the estimated rankings of the models were more correlated with the true rankings when using PPI++ (Figure 5.4c).

We also studied other choices of LLM judges (Figure 5.5). Similar to the protein fitness experiment, the quality of the LLM judge had a large impact on the performance of our approach. Furthermore, we observed that all considered judges allowed our approach to outperform the classical approach, obtaining ESS improvements between 20% to 35% depending on the LLM judge.

5.4 Discussion

AutoEval is a promising direction to reduce the cost and effort of model evaluation. We have presented a methodology that makes it possible to use such synthetic data for model evaluation, improving over classical approaches in a statistically rigorous way. Our implementation is available as a Python package.

It is worth noting that the statistical methods we have presented apply beyond autoevaluation. Indeed, many setups involve trading off low-quality or unreliable validation labels, which are plentiful, with high-quality but scarce validation labels. Our methodology applies readily to such setups, and could, for instance, help integrate crowd-sourced validation labels with expert validations.

One limitation of our approach is that it requires the curated expert and autoevaluated data to be representative of the data in production. Exploring strategies to account for distribution shifts is an important direction for future work. One straightforward extension, for example, would be to apply the importance sampling approach in [48] to handle covariate shifts.

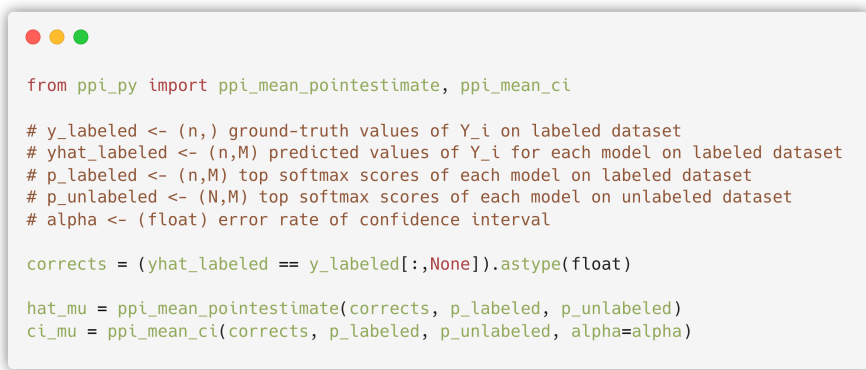
Finally, it is interesting to consider other metrics by which one could evaluate models with PPI-type approaches. Herein, we handled mean estimation and logistic regression, but the framework can do more. One might want to evaluate other metrics, such as fairness and bias, e.g., via estimating the least-squares coefficients relating sensitive attributes and prediction error. For any deployed machine learning system, it is important to test many of these metrics to ensure good performance, in which case having precise estimates and tight confidence intervals becomes especially important.

Data and Code Availability

All code used to reproduce this work is available at <https://github.com/pierreboyeau/autoeval>. Tools to apply the described methodology for model evaluation are available as a Python package. The package is available at https://github.com/aangelopoulos/ppi_py.

5.5 Code snippets

This section provides code snippets to produce confidence intervals and point estimates for model accuracy and pairwise comparisons with the existing Python package `ppi_py` [48].



```

from ppi_py import ppi_mean_pointestimate, ppi_mean_ci

# y_labeled <- (n,) ground-truth values of Y_i on labeled dataset
# yhat_labeled <- (n,M) predicted values of Y_i for each model on labeled dataset
# p_labeled <- (n,M) top softmax scores of each model on labeled dataset
# p_unlabeled <- (N,M) top softmax scores of each model on unlabeled dataset
# alpha <- (float) error rate of confidence interval

corrects = (yhat_labeled == y_labeled[:,None]).astype(float)

hat_mu = ppi_mean_pointestimate(corrects, p_labeled, p_unlabeled)
ci_mu = ppi_mean_ci(corrects, p_labeled, p_unlabeled, alpha=alpha)

```

Snippet 1: Python code to produce CIs and point estimates for model accuracy. The variable meanings are explained in the code comments.

5.6 Experimental details

Data acquisition and preprocessing

ImageNet We downloaded model weights from Pytorch’s model zoo for the different ResNet models, trained on the training set of ImageNet. We then computed the different models’ predictions on the validation set of ImageNet on a high-performance computing cluster.

Protein fitness We relied on ProteinGym¹ to access both the ground-truth fitness values and the predictions of the different protein language models for a specific assay corresponding to IgG-binding domain mutations of protein G (SPG1_STRSG_Olson_2014). All fitness scores were normalized as a preprocessing step.

LLM We considered a subset of the Chatbot Arena dataset aiming to rank twenty recent LLMs (Table S1). The data contained 16K human preferences over pairs of LLM answers to the same (human-provided) prompts. In parallel, using a similar procedure as in [221], we prompted a judge LLM (gpt-4o-mini) to provide its own preferences for the same prompts and LLM answers.

¹<https://github.com/OATML-Markslab/ProteinGym>


```

from ppi_py import ppi_logistic_pointestimate, ppi_logistic_ci

# X_labeled <- (n,M-1) one row per battle; -1 for model A, 1 for model B, 0 else
# y_labeled <- (n,) 0 if model A wins, 1 if model B wins
# yhat_labeled <- (n,) predicted values of Y_i on labeled dataset
# X_unlabeled <- (N,M-1) one row per battle; -1 for model A, 1 for model B, 0 else
# yhat_unlabeled <- (n,) predicted values of Y_i on unlabeled dataset
# alpha <- (float) error rate of confidence interval

hat_zeta = ppi_logistic_pointestimate(
    X_labeled, y_labeled, yhat_labeled,
    X_unlabeled, yhat_unlabeled
)

ci_zeta = ppi_logistic_ci(
    X_labeled, y_labeled, yhat_labeled,
    X_unlabeled, yhat_unlabeled, alpha=alpha
)

```

Snippet 2: Python code to produce CIs for the Bradley-Terry coefficients (without multiplicity correction). The variable meanings are explained in the code comments. For clarity, the matrix `X_labeled` has one row per pairwise comparison. The i th row is a two-hot vector, with -1 at position A_i and $+1$ at position B_i . The matrix `X_unlabeled` is analogous. Note that `X_labeled` and `X_unlabeled` have only $M - 1$ columns, since ζ_1 does not need to be estimated.

Methodological details

Monte Carlo trials In all experiments, we randomly split the data into labeled and unlabeled sets 250 times, and computed all point estimates in the main text and in this supplementary material as the average estimate over these splits.

Model ranking To rank models with the different estimators, we computed 90% confidence intervals for the different approaches after Bonferroni correction. Models with overlapping confidence intervals were assigned the same rank.

Experimental setup

All AutoEval experiments were run on a workstation with 12th generation Intel (R)Core (TM) i9-12900KF, 128GB of RAM, and on a compute cluster relying on CPU nodes with four cores. We relied on the Python package `ppi_py`, except for the

Table 5.1: Overview of the evaluated language models. Models are grouped by their base architecture family/provider.

Family	Model	Version/Date
OpenAI	GPT-4	2023-03-14
	GPT-4	2023-06-13
	GPT-4-Turbo	2024-04-09
	GPT-4 Preview	2023-11-06
	GPT-4-Online	2024-05-13
Anthropic	Claude-3-Opus	2024-02-29
	Claude-3-Sonnet	2024-02-29
	Claude-3-Haiku	2024-03-07
Google	Gemini-1.5-Pro	2024-05-14
	Gemini-1.5-Flash	2024-05-14
Meta	LLaMA-3-70B-Instruct	—
	LLaMA-3-8B-Instruct	—
Mistral AI	Mistral-Large	2024-02
Other	Command-R	—
	Command-R-Plus	—
	PHI-3-Medium-4K-Instruct	—
	Qwen-1.5-72B-Chat	—
	Qwen-2-72B-Instruct	—
	Starling-LM-7B-Beta	—
	YI-1.5-34B-Chat	—

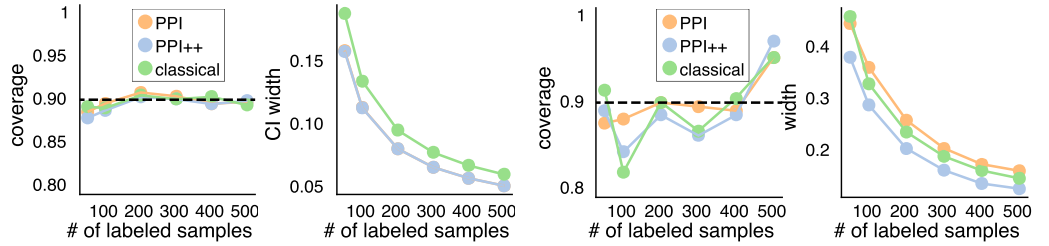
LLM experiment, for which we relied on Jax to implement PPI and PPI++ for the Bradley-Terry model.

5.7 Additional experiments

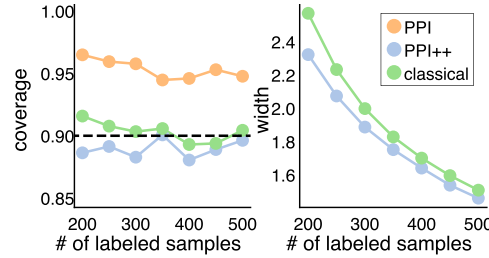
This section provides additional experiments for the different experiments presented in the main text.

Table 5.2: Running times for the different experiments (in seconds). Displayed are the times to produce confidence intervals for PPI++, PPI, and the classical approach on a single labeled-unlabeled split.

	ImageNet	ProteinGym	LLM
Execution time	$2.3 \times 10^{-2}s$	$2.6 \times 10^{-1}s$	8.4s



(a) ImageNet: estimation of model accuracy. (b) Protein fitness: estimation of model correlation.



(c) LLM: estimation of BT coefficients

Figure 5.6: **Interval metrics for the different experiments.** a. and b. Coverage and width of the 90%-confidence intervals (without multiplicity correction). Each experiment is described in the main text and focuses on the estimation of a different metric.

Interval metrics

We here show that the proposed methodology provides *calibrated* and *tight* confidence intervals for the various measurements of model performance described in the main text. Figure 5.6 displays the coverage and width of the 90%-confidence intervals for the different experiments, all of which show that both PPI and PPI++ are calibrated, as well as showing that PPI++ provides tighter confidence intervals than the classical

approach.

Chapter 6

Uncertainty Quantification of Algorithmic Errors onto Cellular Measurements in Spatial Omics

6.1 Introduction

For years, efforts to understand cell phenotypes from molecular readouts often required removing cells from tissues, leading to the loss of spatial contexts necessary to understand how cells influence and are influenced by their environment. Spatial transcriptomics (ST) now directly addresses this challenge, offering new ways to map gene expression in context and at subcellular resolution [249–252]. Spatial differential expression (DE) is now a key analysis to map gene expression changes across different tissue locations. By comparing cells based on their region or neighborhood, these analyses clarify the specific roles cells play within tissues. Spatial DE has revealed key biological insights across scales, from gene expression gradients in cortical layers [253], to molecular heterogeneity in aging tissues [254], or cell-cell communication within specialized niches [255].

Current ST technologies provide rich information on transcripts localization in tissues, but critically do not directly inform on the cell that produced the transcript. Cells, and not transcripts, are the fundamental units spatial DE focuses on. For this reason, two main preprocessing steps are typically required to reconstruct the cellular and cell-state information essential for spatial DE. The first step is cell segmentation, which defines the boundaries of individual cells within the tissue. This process effectively assigns transcripts to their parent cells, creating the distinct cellular units whose gene expression will be compared. Segmentation is often guided

by morphological images (like stains for cell nuclei or membranes) or by leveraging spatial patterns directly from the transcriptomic data itself [155, 256]. Once cells are defined, they are typically classified into specific types based on their inferred gene expression profiles [257, 258]. With both cellular boundaries and types established, spatial DE can then effectively compare, for example, how gene expression in a specific cell type changes across different regions, or how distinct cell types behave within the same local environment.

Unfortunately, inaccuracies in cell segmentation compromise the reliability of biological insights obtained from spatial DE. Errors typically occur at cell boundaries, or across z-planes, and translate into misattribution of RNA content to cells. When these errors are *systematic*, they introduce artificial patterns of spatial DE, which may be misinterpreted as biological signals. A well-documented example is colocalization effects [256], where proximity between cell groups can lead to false differential expression signals for genes that are not actually expressed by the cells under comparison. Critically, recent work by [51] reveals these artifacts dominate the top gene expression differences found by spatial DE, effectively masking true biological signals. Similar problems likely arise during cell type classification, in particular with automated or semi-automated methods, where misclassifications can also significantly impact spatial DE.

Despite active research on cell segmentation [155, 256], resulting errors are likely to continue impacting spatial DE analyses for the foreseeable future. Current ST measurements, limited in sensitivity and resolution across z-planes, may not contain enough information to define all cell boundaries unambiguously. This ambiguity in boundary definition means RNA misattribution between neighboring cells may be unavoidable regardless of algorithmic improvements. Any such consistent misassignment of RNA will bias spatial DE analysis. This requires rethinking analytical strategies to confidently draw inferences from imperfectly processed data for spatial DE.

To address these challenges, we introduce Corrected Spatial Differential Expression (CSDE), a statistical framework for robust spatial DE analysis. CSDE accurately quantifies log fold-changes even when cell segmentation is imperfect. Our approach leverages prediction-powered inference [48], which uses a small, manually curated validation dataset to assess and correct for uncertainties arising from models applied to large-scale ST data. This yields calibrated p-values for spatial DE, ensuring valid statistical inference despite segmentation errors. Furthermore, CSDE incorporates an efficient workflow to generate the required validation data, prioritizing cells of interest and minimizing expert annotation time. CSDE is available as an open-source Python package at <https://github.com/YosefLab/CSDE>.

6.2 Related work

Several strategies have been considered to address poor segmentation effects on spatial DE. A first direction has been to develop segmentation-free methods, operating directly on the raw data and without requiring to identify individual cells via segmentation [259, 260]. [260], for instance, bypasses the need for cell segmentation by mapping cell-type regions directly from spatial transcript data. Another direction has consisted in improving cell segmentation. These efforts have been consolidated by considering more complex segmentation models [49], trained on larger datasets [261], potentially using fine-tuning on target data [50]. These efforts have been complemented by machine learning-assisted segmentation methods, facilitating cell segmentation from imaging data [262, 263]. Alternative segmentation methods have also been engineered for spatial transcriptomics data by leveraging information on transcriptome localization [155, 256]. Recently, post-segmentation tools have emerged, aiming to identify and correct missassignments of transcripts to cells based on subcellular distribution patterns [51] or via deconvolution [264]. These strategies, however, cannot help quantify the effect of segmentation errors on downstream analysis, e.g., spatial DE.

This fundamental limitation highlights a recurring challenge in computational biology: the propagation of uncertainty from preprocessing steps to downstream statistical analyses. In single-cell RNA-seq, for instance, preprocessing steps that include read alignment and count normalization are necessary to obtain count matrices for statistical analyses. Recent work has focused on quantifying how uncertainty in these preprocessing steps affects downstream analyses. For instance, [265] addresses alignment uncertainty in read mapping, while [266] and [267] examine how normalization choices propagate to differential expression analysis.

Recently, prediction-powered inference (PPI) has emerged as a powerful framework for producing statistically valid inferences from machine learning predictions [48, 230, 268]. This approach addresses the challenge of uncertainty propagation by combining two complementary data sources: a small set of points with both machine learning predictions and ground truth labels to obtain statistically valid inference, and a larger set with only predictions to improve power. This framework has been applied to assess the effect of preprocessing steps of biological pipelines, with applications to proteomic analysis [48], genome-wide association studies [269], and protein language model selection [270].

6.3 Method

Our objective is to estimate reliable spatial differential expression effects from spatial transcriptomics data. Our approach, Corrected Spatial Differential Expression (CSDE), is a statistical model leveraging two data modalities: (i) a larger *ML-derived dataset*, generated through automated segmentation and classification, and (ii) a manually annotated *reference dataset*, containing ground-truth annotations for a random subset of cells from the ML-derived dataset.

CSDE combines these datasets for parameter inference. It leverages the large *ML-derived dataset* to increase statistical power by reducing the variance of estimates. Simultaneously, it uses the accurate *reference dataset* to quantify and correct for bias introduced by ML-induced errors. This combined strategy addresses the trade-off between data quantity and quality, providing reliable, and in particular, statistically valid spatial DE estimates that leverage all available data.

Figure 6.1 illustrates our approach.

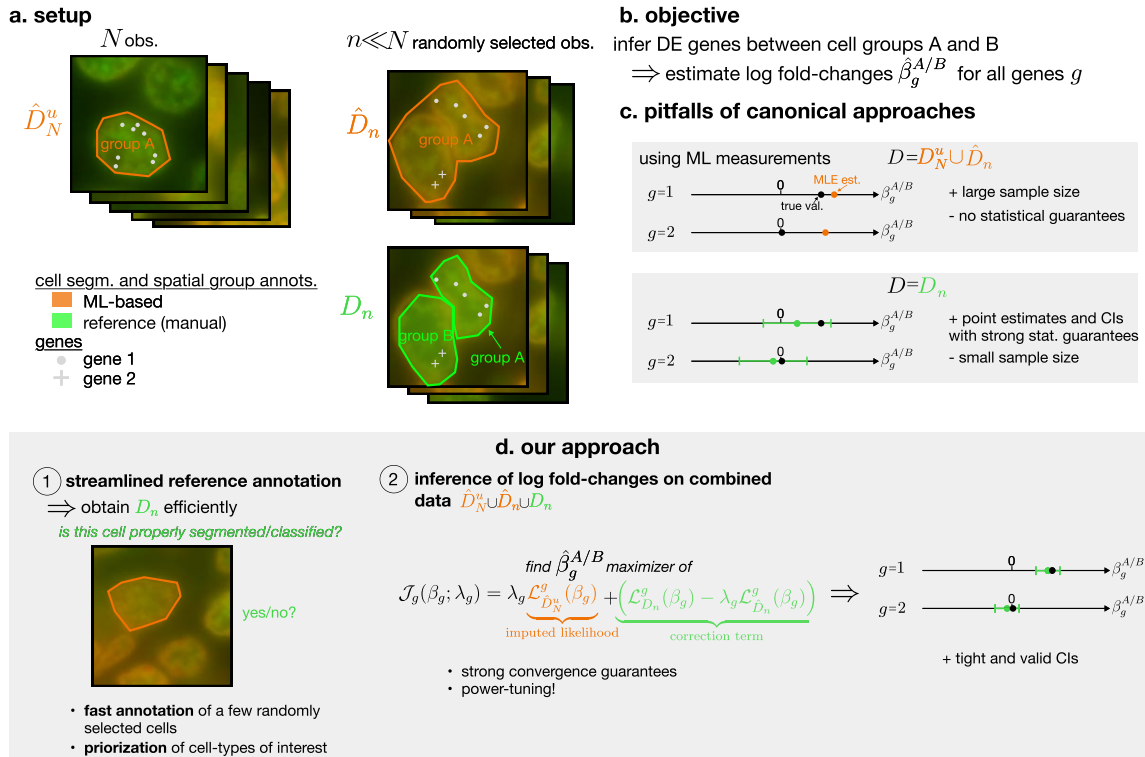


Figure 6.1: Visualization of the approach

Quantifying the impact of algorithmic errors on spatial DE analysis

We consider a spatial transcriptomics assay profiling a total of G genes in one or several tissue samples. The assay generates a series of high-resolution images that capture both the spatial organization of cells and gene expression patterns at subcellular resolution. Each image contains multiple channels encoding both morphological features (e.g., nuclei and membrane stains) useful to localize cells, and gene-specific fluorescence signals necessary to quantify gene expression.

For analytical tractability, we assume that individual cells and their approximate location are known, allowing us to extract images centered on individual cells. We let $\mathcal{I}$ denote the random variable containing all raw information captured in the spatial transcriptomics data for a single cell. This includes all gene expressions localized within the field of view, information on the cell's location in the tissue, and morphological features.

Each raw (cell-centered) observation $\mathcal{I}$ is *preprocessed* to extract the measurements necessary to perform DE analysis. We assume there is an unknown mapping f^* that represents this true measurement process, extracting the (true) gene expression X and spatial group assignment Y from raw observations as

$$f^* : \mathcal{I} \rightarrow f^*(\mathcal{I}) = (X, Y). \quad (6.1)$$

Here, X corresponds to the G -dimensional vector of gene expression levels in cell j ; in particular, X_g is the number of mRNA molecules of the g -th gene present in the cell. $Y \in \{0, \dots, K\}$ is the cell assignment to one of $K + 1$ spatial groups. A spatial group typically represents a combination of cell type and spatial region, where we assume the tissue has been pre-divided into distinct regions of interest. In the mouse brain for instance, these regions could correspond to specific cell types (e.g., neurons, glia, etc.) localized within predefined cortical layers or anatomical structures.

Since f^* is unknown, we consider two approaches to extract measurements from raw images.

Reference measurements Manual cell segmentation and classification provides the gold-standard approach to obtain cell measurements. When performed carefully, we assume this strategy provides perfect segmentation and classification, yielding ground-truth measurements. However, manual annotation is time-consuming and does not scale to the number of cells in typical ST experiments. We assume that a small set $(\mathcal{I}_j)_{j=1}^n$ of n images, selected uniformly at random, have been manually

annotated, resulting in true measurements

$$D_n = (X_j, Y_j)_{j=1}^n, \quad \text{where} \quad (X_j, Y_j) = f^*(\mathcal{I}_j) \quad \forall j \in [n]. \quad (6.2)$$

ML-derived measurements Because the manual strategy scales poorly, we also assume a ML approach can produce a mapping $\hat{f}$ that approximates f^* of Equation (6.1) to obtain cell measurements. The approximation is imperfect and can deviate from f^* due to systematic errors in cell segmentation and classification, implying that the resulting measurements cannot be assumed ground-truth. However, this strategy is scalable and allows us segment and classify all cells in the tissue.

We introduce two datasets of ML-derived measurements. $\hat{D}_n$ contains ML-derived measurements for the n images $(\mathcal{I}_j)_{j=1}^n$ with available manual measurements, and $\hat{D}_N^u$ contains ML-derived measurements for the remaining N images, that we denote as $(\mathcal{I}_j^u)_{j=1}^N$. In particular,

$$\begin{aligned} \hat{D}_n &= (\hat{X}_j, \hat{Y}_j)_{j=1}^n, \quad \text{where} \quad (\hat{X}_j, \hat{Y}_j) = \hat{f}(\mathcal{I}_j) \quad \forall j \in [n], \\ \hat{D}_N^u &= (\hat{X}_j^u, \hat{Y}_j^u)_{j=1}^N, \quad \text{where} \quad (\hat{X}_j^u, \hat{Y}_j^u) = \hat{f}(\mathcal{I}_j^u) \quad \forall j \in [N]. \end{aligned} \quad (6.3)$$

Interpretable modeling of spatial DE

We now introduce the statistical model used by CSDE for interpretable DE inference. We employ the following generalized linear model (GLM) operating on transcriptomic measurements, where draws are independent across cells and genes:

$$X_{ig} \mid Y_i = k \sim \text{Poisson}(\mu_{kg}) \quad \text{where} \quad \mu_{kg} = \begin{cases} \exp(\beta_{kg}) & \text{if } k = 0, \\ \exp(\beta_{kg} + \beta_{0g}) & \text{if } k \in \{1, \dots, K\}. \end{cases} \quad (6.4)$$

This model produces interpretable parameters that directly quantify DE effects. μ_{kg} is the mean expression level of the g -th gene in the k -th spatial group, determined by the model parameters β_{kg} . The β_{kg} parameters directly quantify the differential expression effects, representing the log-fold change (LFC) in the mean expression level of gene g between spatial group k and the reference group (group 0). A value of zero indicates equal expression, while positive and negative values indicate higher and lower expression, respectively.

Our goal is to estimate these model parameters from data, with a focus on comparing gene expression patterns between spatial groups. Without loss of generality, when comparing group k to the reference group (group 0), we aim to obtain reliable point estimates of the log fold-changes β_{kg} for all genes $g \leq G$. We also want to

quantify uncertainty on these estimates through confidence intervals, which allows us to identify genes with statistically significant DE effects by testing the null hypothesis $\mathcal{H}_0^{k,g} : \beta_{kg} = 0$.

Human-in-the-loop spatially-resolved differential expression

The model parameters in Equation (6.4) are classically estimated via maximum likelihood estimation (MLE). Applying MLE to true cell measurements (D_n) yields asymptotically normal estimates of DE effects β_{kg} . This enables the computation of valid confidence intervals and p-values for differential expression. However, the limited sample size of D_n results in wide confidence intervals, meaning that few genes might be detected as DE. Applying MLE to ML-derived measurements offers a larger sample size, but statistical guarantees cannot be easily established because these measurements are obtained from a different data generating process than the one assumed by the model.

CSDE fits Model (6.4) on the combination of true and ML-derived measurements. To do so, we introduce the log-likelihood of gene $g \leq G$ for any given dataset D of cell measurements:

$$\mathcal{L}_D(\beta^g) = \frac{1}{|D|} \sum_{(X_i, Y_i) \in D} \log p(X_i, Y_i; \beta^g)$$

where $p(X_i; Y_i, \beta^g)$ is the gene expression probability given the spatial group assignment and model parameters β^g , $|D|$ the dataset size, and $\beta^g = (\beta_{0g}, \beta_{1g}, \dots, \beta_{Kg}) \in \mathbb{R}^{K+1}$ the vector of parameters for the g -th gene. For a given gene g , and following [230], we define the prediction-powered objective function $\mathcal{J}_g(\beta_g; \lambda_g)$ as

$$\mathcal{J}_g(\beta_g; \lambda_g) = \lambda_g \underbrace{\mathcal{L}_{\hat{D}_N^u}^g(\beta_g)}_{\text{imputed likelihood}} + \underbrace{\left(\mathcal{L}_{D_n}^g(\beta_g) - \lambda_g \mathcal{L}_{\hat{D}_N^u}^g(\beta_g) \right)}_{\text{correction term}}, \quad (6.5)$$

and consider the following estimate of the DE effects:

$$\hat{\beta}_g = (\hat{\beta}_{0g}, \hat{\beta}_{1g}, \dots, \hat{\beta}_{Kg}) = \arg \max_{\beta_g \in \mathbb{R}^{K+1}} \mathcal{J}_g(\beta_g; \lambda_g). \quad (6.6)$$

Here, $\lambda \in [0, 1]$ is a tunable hyperparameter that controls the weight of the ML-derived measurements for parameter estimation, which we will discuss in more detail later.

We now unpack the rationale behind this estimator, and the objective function it optimizes. $\mathcal{J}_g(\beta_g; \lambda_g)$ contains two terms. The first term is the imputed likelihood of the (unreliable) ML-derived measurements $\hat{D}_N^u$. The second term involves the

likelihood of the data points for which both true and ML-derived measurements are available, and can be interpreted as a correction term. This correction term ensures that the objective $\mathcal{J}_g(\beta_g; \lambda_g)$ is not biased by the ML-derived measurements, as $\mathbb{E}[\mathcal{J}_g(\beta_g; \lambda_g)] = \mathbb{E}[\log p(X_1, Y_1; \beta_g)]$.

$\hat{\beta}_g$ also enjoys strong statistical guarantees. In particular, $\hat{\beta}_g$ is consistent, and asymptotically normal [230]. More specifically, for any given spatial group k , gene g , and $\lambda \in [0, 1]$, we have that

$$\sqrt{n}(\hat{\beta}_{kg} - \beta_{kg}^*) \xrightarrow{d} \mathcal{N}(0, \sigma_{kg}^2),$$

where σ_{kg}^2 is the asymptotic variance of $\hat{\beta}_{kg}$ that can easily be estimated from data, and β_{kg}^* is the true parameter value.

Selecting λ The objective function in Equation (6.6) contains a hyperparameter λ that controls the weight of the ML-derived measurements for parameter estimation. The original PPI++ approach [230] outlined a strategy to select λ to optimize the total variance of a multi-dimensional estimate of the parameters θ . This strategy is suboptimal for our problem, as we are interested in estimating a specific coordinate of the parameter vector β_g , corresponding to a specific spatial group k compared to the reference group 0. For this reason, we select λ to minimize the variance of the coordinate $\hat{\beta}_{kg}$ of $\hat{\beta}_g$.

Streamlined manual annotation of spatial transcriptomics

Here, we now describe two complementary strategies we employ to speed up the annotation of spatial transcriptomics data from raw images, allowing to construct human-labeled datasets D_n in a streamlined manner.

Streamlined manual annotation

We propose an efficient approach to cell annotation that combines the speed of automated segmentation with the reliability of human expertise. Instead of manually segmenting and classifying cells to obtain reference measurements, we use validation-based annotation, where the annotator only needs to confirm or reject the ML-derived segmentation and classification for a small subset of randomly sampled cells.

The validation process consists of two main steps, illustrated in Figure 6.2. First, the annotator assesses the quality of cell segmentation by examining the ML-derived boundaries overlaid on the raw image (Figure 6.2a). Second, if the segmentation is correct, the annotator verifies the cell type assignment using two complementary

pieces of information: (i) the list of most expressed genes in the cell, contrasted with the expected gene expression pattern for the assigned cell type (Figure 6.2b), and (ii) the spatial distribution of relevant genes in situ, which helps identify potential missegmentation artifacts. The annotator can then make an informed decision about whether the cell’s segmentation and classification are correct.

Based on these checks, the annotator simply marks the cell as *accepted* or *rejected*, a binary decision that is both faster and less prone to error than manual segmentation and classification. This process ensures proper cell classification and segmentation while identifying any spurious gene expression patterns that may arise from missegmentation.

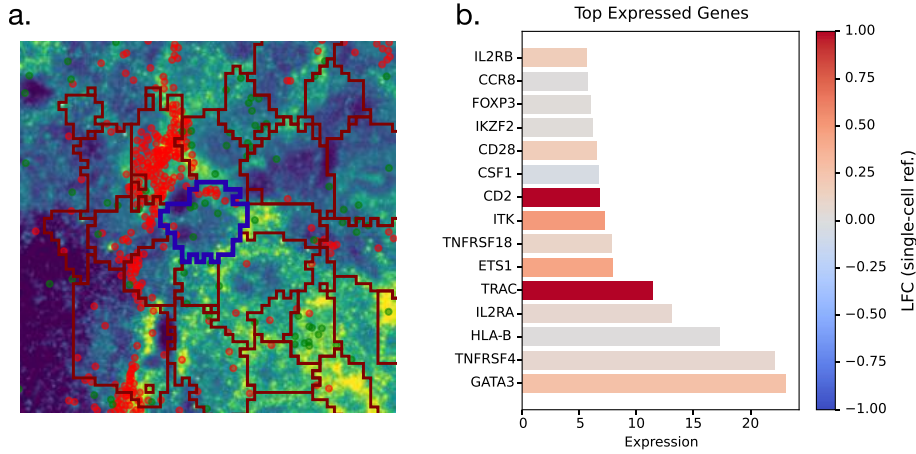


Figure 6.2: Example of images used for validation-based annotation.

Importance sampling

Typically, the types of inferences we are interested in relate to a small portion of the cells in the tissue. In the canonical case where cells of the same type are compared across spatial regions, a consequence of this is that an untargeted annotation strategy will spend a lot of energy on cells that are not likely to be of interest.

To alleviate this problem, we use importance sampling to focus manual annotation on images that are more likely to be of interest.

In particular, we assume available a weighting function $w : \mathcal{I} \mapsto \mathbb{R}^+$ measuring the importance of each image. In practice, we use custom-designed weighting functions allowing us to focus on images for which the ML annotation predicts spatial groups

of interest. Instead of sampling images uniformly at random for manual annotation, we sample images proportionally to their weight.

Because of the distributional shift induced by this weighting, our DE estimation needs to be adjusted. If w_i is the weight of the i th manually labeled cell, and using the same notations as in the unweighted case, our estimator in Equation (6.6) optimizes the modified objective:

$$\mathcal{J}_g(\beta_g; \lambda_g) = \lambda_g \mathcal{L}_{\hat{D}_N^u}^g(\beta_g) + \left(\sum_{i=1}^n \eta_i (\log p(X_i, Y_i; \beta^g) - \lambda_g \log p(\hat{X}_i, \hat{Y}_i; \beta^g)) \right), \quad (6.7)$$

where $\eta_i = \frac{1/w_i}{\sum_{j=1}^n 1/w_j}$ is the normalized weight of the i th manually labeled cell.

6.4 Results

To benchmark our approach, we focused on characterizing infiltrating T cells expression profiles in various cancer types. Concretely, our goal was to identify DE genes for the comparison of T cells inside and outside tumors in two tissues: lung and colon. Segmentation and classification errors are a critical challenge in this context, known to introduce spurious DE results that can mask true biological signals.

A total of four slides (two for lung and two for colon) were extracted from a human immuno-oncology dataset acquired using the MERSCOPE platform [271]. These assays inform on the localization of 550 targeted gene transcripts with subcellular localization precision, as well as the general tissue architecture using DAPI, PolyT and custom antibodies.

ML-derived cell segmentation & classification We used a combination of segmentation algorithms and canonical single-cell clustering and annotation methods to segment and classify cells. For segmentation, we employed Proseg [256], a transcriptomic-based cell segmentation method, to locate cells and characterize their expression profiles. After segmentation, we employed the Leiden clustering algorithm [214] on log-transformed and median-normalized expression data to cluster cells, which were then manually labeled at coarse-resolution via marker genes (Table 6.1).

Overall, although the annotations were manually curated, the obtained annotations are the direct product of an algorithmic approach to segment and cluster cells into homogeneous groups. In particular, many cells can be mislabeled during this step, making the resulting data, that we refer as the *ML-derived dataset*, less reliable than the reference annotations that we introduce now.

Manual annotation curation For each slide, we built a *reference dataset* validating or invalidating the ML-derived cell segmentation and type annotations for a random sample of 600 cells per slide. The cells of interest in this study, T cells, were relatively rare in the different slides according to the ML pipeline (Table 6.2). To ensure true T cells were sufficiently represented in the sample, we employed the importance sampling scheme mentioned in the Methods section by prioritizing cells identified as T cells by the ML pipeline. This curation step was conducted by a single annotator, who produced, for each cell, a binary annotation indicating whether the cell was properly segmented and type-classified using an image annotation tool [272]. This approach focused our manual annotation effort on T cells, which were relatively rare in the different slides (Table 6.2).

Benchmark We compared CSDE with two baselines representing current approaches for spatial DE analysis. The first baseline, *imputation*, identified DE effects using only the ML-derived dataset. This method reflects the common practice of using automated segmentation and classification with minimal manual curation. The second baseline, *classic*, identified DE effects using only the small, manually validated reference dataset. This approach represents a traditional method relying solely on high-confidence manual annotations, inherently limiting the number of cells analyzed.

Importantly, all methods employed the same underlying Poisson model. The main difference between these methods resided in the inference procedure and the data they used. The *imputation* and *classic* baselines applied standard likelihood-based inference to their respective datasets (ML-derived and manually validated reference data) to estimate log fold-changes. Our approach, CSDE, combined both datasets using the prediction-powered inference framework detailed in the Methods section. DE genes were identified for each method as those with multiplicity-corrected p-values below 0.05 [37].

Lung cancer

We first focused on characterizing T cell infiltration in lung cancer. The first slide served as our primary dataset for main analyses, while the second was used as a biological replicate to assess the reproducibility of results across different methods. After segmentation, the ML pipeline applied to the first slide returned a dataset containing 351,309 cells that were coarsely annotated using clustering (Figures 6.3a, 6.6a).

The next step in this analysis was to identify tumor regions in the slide. We employed a nearest-neighbor-based approach to delineate tumor regions based on proximity to tumor cells. Figure 6.6b shows the resulting tumor region assignments.

To apply CSDE, we manually validated the segmentation and type assignment of 600 randomly sampled cells in the slide, as previously described. This validation required a binary assessment per cell and was completed by one annotator in less than an hour, highlighting its efficiency.

Investigation of the manual annotations revealed significant segmentation and misclassification errors. Among the cells predicted as T cells by the ML pipeline and manually reviewed, only 47% were confirmed as correctly segmented and classified T cells. Cells misannotated as T cells by the ML pipeline included actual T cells with improper segmentation or classification, and non-T cells incorrectly classified as T cells (Figure 6.7).

Comparative evaluation

We first compared methods based on their ability to quantify DE effects between T cells in tumors and adjacent tissue, using two scores. The plausibility score quantifies the extent to which genes with strong DE effects are known to be expressed by T cells, calculated as the Spearman correlation between estimated absolute log fold-changes and gene expression in T cells from a reference lung cancer data. The reproducibility score measured the consistency of inferred log fold-changes between the two slides, calculated as the Spearman correlation between log fold-changes estimated independently on the two slides. Both classic and our method achieved higher plausibility scores than imputation. This indicates that manual validation of cell annotations yields more biologically relevant DE estimates. The lower performance of imputation likely is a consequence of segmentation and classification errors in the ML-derived data. Classic and our approach also yielded higher reproducibility scores, suggesting more robust log fold-change estimates. Reproducibility scores were, however, overall low, likely due to biological variability between the two slides.

We next analyzed the genes identified as differentially expressed by each method. Imputation detected the largest number of genes, which was expected given the larger sample size of the ML-derived data. CSDE identified significantly more genes than the classic approach (31 versus 21 genes). The substantial overlap between discoveries from CSDE and the classic method indicates that CSDE enhances detection power while maintaining biological validity.

To quantitatively assess the biological relevance of the discoveries, we examined the expression of the detected genes in a reference single-cell RNA-seq dataset (Figure 6.3d). As expected, genes identified by the classic method showed higher expression in T cells within the reference dataset compared to those from the imputation approach. This observation suggests that some genes detected by imputation may not be genuinely expressed in T cells but could be artifacts of the ML pipeline. CSDE

discoveries exhibited expression levels in the reference dataset similar to those from the classic approach.

We examined genes detected by the different approaches in more detail by focusing on those not typically associated with T cells (Figure 6.3e). Imputation produced a substantial number of unexpected discoveries not detected by CSDE, most likely representing artifacts introduced by the ML pipeline. These included genes characteristically expressed by stromal (*COL1A1*, *PECAM1*), endothelial (*VEGFB*), myeloid (*CSF1*), and MHC class II (*HLA-DRA*, *HLA-DRB1*, *HLA-DPB1*, *HLA-DMA*) cells.

Collectively, these findings suggest that CSDE effectively identifies a large number of biologically relevant discoveries while largely avoiding spurious findings likely resulting from ML-induced artifacts.

Characterization of infiltrating T cells

We compared discoveries from CSDE and the classic approach, which produced in the previous section the most biologically relevant discoveries to characterize infiltrating CD4 and CD8 T cells separately (Figure 6.3f).

Analysis of CD4 T cells infiltrating the tumor compared to external T cells (Figure 6.3g) revealed shared discoveries between CSDE and the classic approach. These findings indicated an activated state in intratumoral CD4 T cells, marked by upregulation of T cell receptor components (*TRAC*), co-stimulatory molecules (*ICOS*, *TNFRSF18*/GITR, *TNFRSF4*/OX40), and engagement of PI3K-AKT (*AKT1*, *PIK3CG*) and NF- κ B (*NFKBIA*, *NFKB2*) signaling pathways.

This activated state was associated with metabolic reprogramming (upregulated *PKM*), enhanced survival signals (increased *BCL2L1*), and cell cycle progression (*CDK6*). Additionally, these cells exhibited signs of exhaustion, with increased expression of inhibitory receptors (*CTLA4*, *HAVCR2*/TIM-3). Canonical Treg signatures (*FOXP3*, *IKZF2*, *GATA3*) suggested immunosuppressive capacity. Furthermore, these cells displayed a tissue-resident profile, characterized by increased *CXCR6* and downregulated recirculation markers (*CCR7*, *KLF2*, *CXCR4*).

Some detections shared by both CSDE and the classic method were unexpected. For example, the detection of MHC class I machinery components (*HLA-C*, *TAP1*) in CD4 T cells was not anticipated. The presence of these artifacts in findings from both methods suggests they originate from imperfections within the manually validated reference data, rather than reflecting methodological flaws in CSDE, whose inference framework provides theoretically consistent estimates of log fold-changes, as detailed in the Methods section. The classic method, relying solely on this validated data, also identified these genes, further supporting this conclusion.

CSDE uniquely identified additional upregulated genes that provide a more detailed view of the intratumoral CD4 T cell state. These included cell cycle regulators (*CDK2*, *CDK4*), reinforcing the proliferative phenotype. CSDE also revealed broader activation signatures, such as key transcription factors (*JUN*, *JUNB*, *NFKB1*, *STAT1*), components of Notch signaling (*NOTCH1*), and factors involved in epigenetic modulation (*HDAC1*, *BRD4*). Furthermore, CSDE refined the characterization of affected regulatory pathways. Upregulated *LGALS9* (a TIM-3 ligand) suggests increased TIM-3 inhibitory checkpoint engagement, while downregulated *EOMES* points to specific modulations in T cell differentiation programs. These CSDE-specific findings underscore a more complex landscape of enhanced proliferation, diverse signaling pathway involvement, and active stress adaptation within tumor-infiltrating CD4 T cells.

We then characterized infiltrating CD8 T cells.

These CSDE-specific findings underscore a more complex landscape of enhanced proliferation, diverse signaling pathway involvement, and active stress adaptation within tumor-infiltrating CD4 T cells.

Intratumoral CD8 T cells exhibited an activated and cytotoxic phenotype compared to the T cell population outside the tumor. Shared detections by CSDE and the classic method included upregulated effector molecules (*GNLY*, *GZMA*, *GZMB*, *CTSW*, *NKG7*, *PRF1*), the activating receptor *KLRK1* (NKG2D), the transcription factor *TBX21* (T-bet), and the co-stimulatory molecule *TNFRSF18* (GITR). This cytotoxic profile was observed in the intratumoral cells. These cells also showed metabolic changes (*PKM* upregulation), adaptation to hypoxia (*HIF1A* upregulation), potential TGF- β signaling (*TGFBR1* upregulation), and exhaustion markers (increased *HAVCR2* /TIM-3). *BAK1* upregulation suggested a balance between cell death and survival. Among the shared detections, unexpected findings included *CSF1*, typically expressed by myeloid cells, and *COL1A1*, characteristic of stromal cells and notably also detected by the classic method.

Downregulation of key T cell signaling and adhesion molecules (*IKBKB*, *IRF4*, *ITGA4*, *CD27*, *CD5*), and chemokine receptors (*CXCR4*, *CCR4*), suggested a dysfunctionally activated state.

CSDE uniquely identified additional genes that enriched the characterization of these intratumoral CD8 T cells. Upregulation of cell cycle regulators (*CDK4*, *CCND1*, *YAP1*) and downregulation of the cell cycle inhibitor *CDKN1B* strongly supported a proliferative drive. CSDE also revealed involvement of signaling pathways, including Wnt/ β -catenin (*CTNNB1*), Notch (*NOTCH1*), and responses to IFN- γ (*IFNGR2*), alongside TGF- β activity (*TGFBI*), metabolic shifts (*IDH1*), and transcription factor engagement (*CEBPB*). Unexpectedly, CSDE also identified downregulated genes such as *HLA-C*, *AKT3*, *CXCR6*, and *TRBC1*. The decreased expression of these

genes warrants consideration as potential indicators of dysfunction, immune escape, or analytical artifacts. The downregulation of *GZMK* suggested a shift towards a more terminally differentiated *GZMB*-dominant cytotoxic state.

Colon cancer

To strengthen the validity of our claims, we applied the same methodology to another cancer type, colon cancer, and for the same problem of characterizing T cell infiltration in tumors. On this problem, the ML pipeline produced a main dataset of 806,291 cells (Figure 6.4a, 6.8a). We followed the same procedure as for the lung data to annotate these segmented cells, delineate tumor regions and manually validate 600 cells that were randomly sampled from the slide (Figure 6.8b).

Comparative evaluation

Focusing on comparing log fold-change estimates, CSDE again provided more plausible and reproducible estimates than imputation (Figure 6.4b). Analyzing identified DE genes, CSDE yielded an intermediate number of discoveries, almost twice as many as the classic approach but significantly fewer than imputation. Consistent with the lung cancer results, most genes detected by CSDE were also found by the classic method. Figure 6.4d demonstrates that genes discovered by CSDE were typically expressed by T cells in the reference single-cell data, supporting the finding that CSDE identifies a larger number of biologically plausible discoveries than the classic approach. As observed previously, the imputation method produced numerous genes not typically associated with T cells, including MHC class II genes likely expressed by neighboring cells (Figure 6.4e).

Characterization of infiltrating T cells

We now focus on the characterization of intratumoral T cell populations of the colon cancer data (Figure 6.4f).

Analysis of CD4 T cells infiltrating colon tumors compared to T cells outside the tumor revealed an highly activated and proliferative state in intratumoral CD4 T cells, marked by upregulation of activation markers (*CD2*, *CD40LG*), the IL-2 receptor component (*IL2RB*), and cell cycle regulators (*PCNA*, *CDK6*, *MSH6*). This activated state was associated with metabolic reprogramming (upregulated *PKM*) and enhanced survival signals (increased *BCL2L1*, *BCL2*), alongside adaptation to the tumor microenvironment through *HIF1A* upregulation and *STAT3* engagement.

The concurrent upregulation of pro-apoptotic *BAX* suggests a complex regulatory balance or cellular heterogeneity.

CSDE uniquely identified additional upregulated genes that provide a more comprehensive picture as to what might be going on in these intratumoral CD4 T cells. These included further cell cycle regulators (*CCND1*, *CDK4*) and the Hippo pathway effector *YAP1*, reinforcing the hypothesis of a proliferative phenotype. CSDE also revealed broader engagement with key signaling pathways and TME-specific responses. Upregulated *CTNNB1* suggests Wnt/ β -catenin pathway activity, while increased *SMAD2* and *TGFBI* point to significant TGF- β signaling, indicative of CD4 T cell interaction with this major immunosuppressive cytokine. Furthermore, CSDE highlighted active adaptation to TME stressors, with upregulated *NFE2L2* (*NRF2*) signaling an antioxidant response, and *IDH1* contributing to metabolic adjustments. Increased *IFNGR2* expression also suggests heightened sensitivity or response to IFN- γ . These CSDE-specific findings underscore a more complex landscape of enhanced proliferation, diverse signaling pathway involvement, and active stress adaptation within tumor-infiltrating CD4 T cells.

Intratumoral CD8 T cells, when compared to the mixed CD4 and CD8 T cell population external to the tumor, exhibited a highly activated and cytotoxic phenotype. The detections shared by CSDE and the classical approach included upregulation of effector molecules (*GNLY*, *GZMA*, *GZMB*, *CTSW*, *NKG7*, *PRF1*), the activating receptor *KLRK1*/NKG2D, the T-box transcription factor *TBX21* (T-bet) crucial for CTL differentiation, and the co-stimulatory molecule *TNFRSF18*/GITR. These cells also displayed hallmarks of metabolic reprogramming (*PKM*) and adaptation to hypoxia (*HIF1A*), alongside potential TGF- β pathway engagement (*TGFBR1*) and signs of exhaustion (*HAVCR2*/TIM-3). *BAK1* upregulation suggested a balance between cell death and survival. Among the shared detections, unexpected findings included *CSF1*, typically expressed by myeloid cells, and *COL1A1*, characteristic of stromal cells and notably also detected by the classic method.

Downregulation of key T cell signaling and adhesion molecules (*IKBKB*, *IRF4*, *ITGA4*, *CD27*, *CD5*), and chemokine receptors (*CXCR4*, *CCR4*), suggested a dysfunctionally activated state.

CSDE uniquely identified additional genes that enriched the characterization of these intratumoral CD8 T cells. Upregulation of cell cycle regulators (*CDK4*, *CCND1*, *YAP1*) and downregulation of the cell cycle inhibitor *CDKN1B* strongly supported a proliferative state. CSDE also revealed involvement of signaling pathways, including Wnt/ β -catenin (*CTNNB1*), Notch (*NOTCH1*), and responses to IFN- γ (*IFNGR2*), alongside TGF- β activity (*TGFBI*), metabolic shifts (*IDH1*), and transcription factor engagement (*CEBPB*).

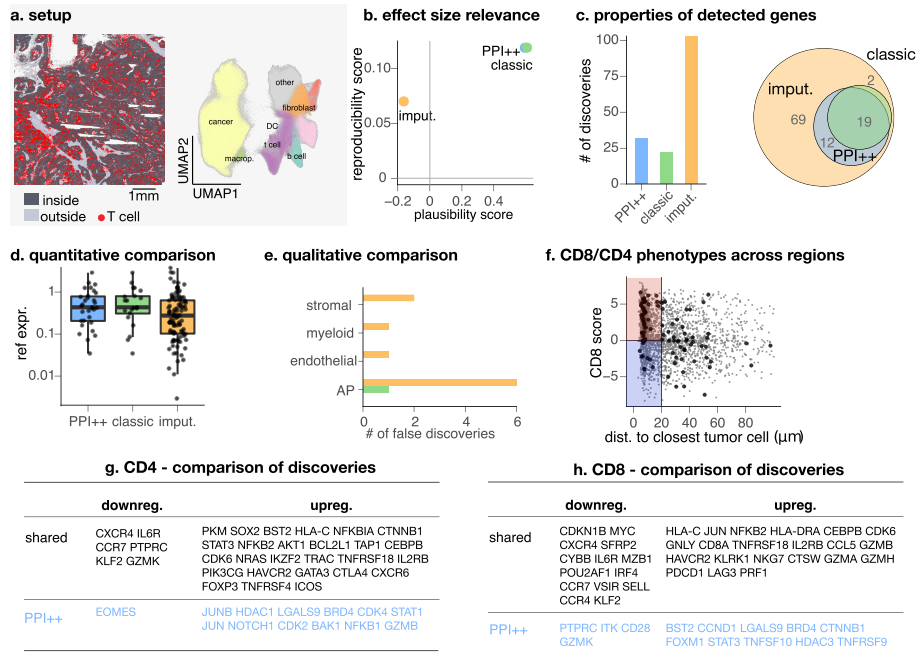


Figure 6.3: Lung experiment. Comparison of T cell gene expression profiles inside and outside tumors. **a.** *Left:* visualization of tumors in the tissue with T cells (obtained via ML pipeline) in red. *Right:* low-dimensional embeddings (UMAP) of the cells obtained by the ML pipeline. **b.** Relevance of the DE effect sizes obtained with the different methods based on reproducibility and plausibility scores. The biological plausibility score assesses the extent to which genes with large positive or negative LFCs are known to be expressed by T cells. The reproducibility score assesses the extent to which the LFC estimates are consistent between the two slides. **c.** Properties of the discoveries made by the different methods. **d.** Comparison of the discoveries using a reference single-cell RNA-seq dataset. Each point represents a gene identified as DE by one of the methods. Plotted is the expression of the gene in T cells found in lung cancer from a reference single-cell RNA-seq dataset (higher is better). **e.** Quantification of non-T cell marker hits for the discoveries made by the different methods (see Table 6.3). **f.** Visualization of T cells, where points represent individual cells (from ML pipeline or manual validation). The x-axis shows distance to the nearest tumor cell, and the y-axis shows the CD8 score (CD8A-CD4 expression). Cells are colored by CD8 score: red for positive (CD8 T cells) and blue for negative (CD4 T cells). **g.** and **h.** Comparison of the discoveries made by our method and the classic approach for the comparison of CD4 T cells infiltrating the tumor (**g**) and CD8 T cells infiltrating the tumor (**h**) against T cells outside the tumor.

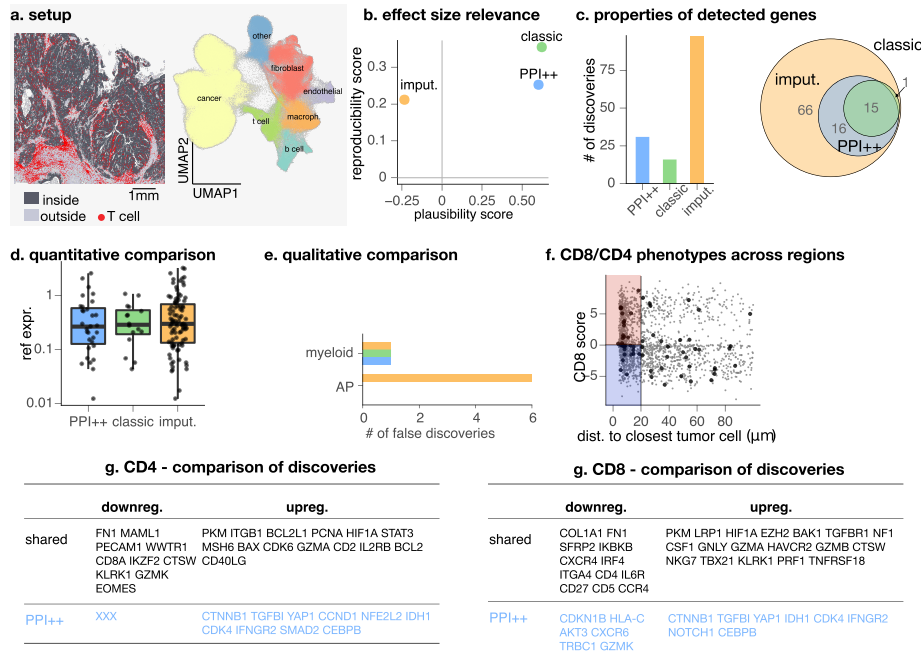


Figure 6.4: **Colon experiment.** Comparison of T cell gene expression profiles inside and outside tumors. **a.** *Left:* visualization of tumors in the tissue with T cells (obtained via ML pipeline) in red. *Right:* low-dimensional embeddings (UMAP) of the cells obtained by the ML pipeline. **b.** Relevance of the DE effect sizes obtained with the different methods based on reproducibility and plausibility scores. The biological plausibility score assesses the extent to which genes with large positive or negative LFCs are known to be expressed by T cells. The reproducibility score assesses the extent to which the LFC estimates are consistent between the two slides. **c.** Properties of the discoveries made by the different methods. **d.** Comparison of the discoveries using a reference single-cell RNA-seq dataset. Each point represents a gene identified as DE by one of the methods. Plotted is the expression of the gene in T cells found in colorectal cancer from a reference single-cell RNA-seq dataset (higher is better). **e.** Quantification of non-T cell marker hits for the discoveries made by the different methods (see Table 6.3). **f.** Visualization of T cells, where points represent individual cells (from ML pipeline or manual validation). The x-axis shows distance to the nearest tumor cell, and the y-axis shows the CD8 score (CD8A-CD4 expression). Cells are colored by CD8 score: red for positive (CD8 T cells) and blue for negative (CD4 T cells). **g.** and **h.** Comparison of the discoveries made by our method and the classic approach for the comparison of CD4 T cells infiltrating the tumor (**g**) and CD8 T cells infiltrating the tumor (**h**) against T cells outside the tumor.

Supplementary Information

6.5 Supplementary Figures

Figure 6.5: Canonical marker gene expression across cell-types in the lung dataset.

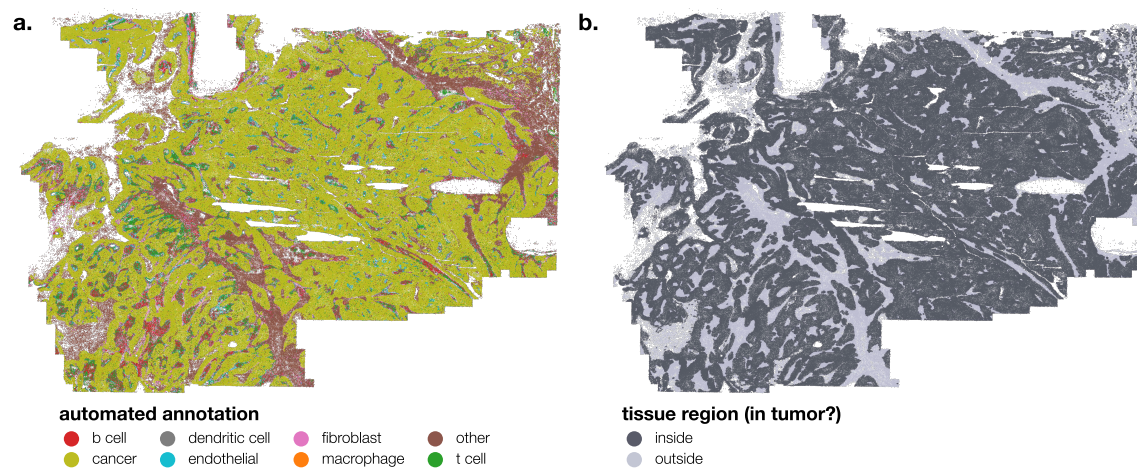


Figure 6.6: **Visualization of cells in the main lung slide**, colored by (a) ML-derived cell-type and (b) tumor region assignments.

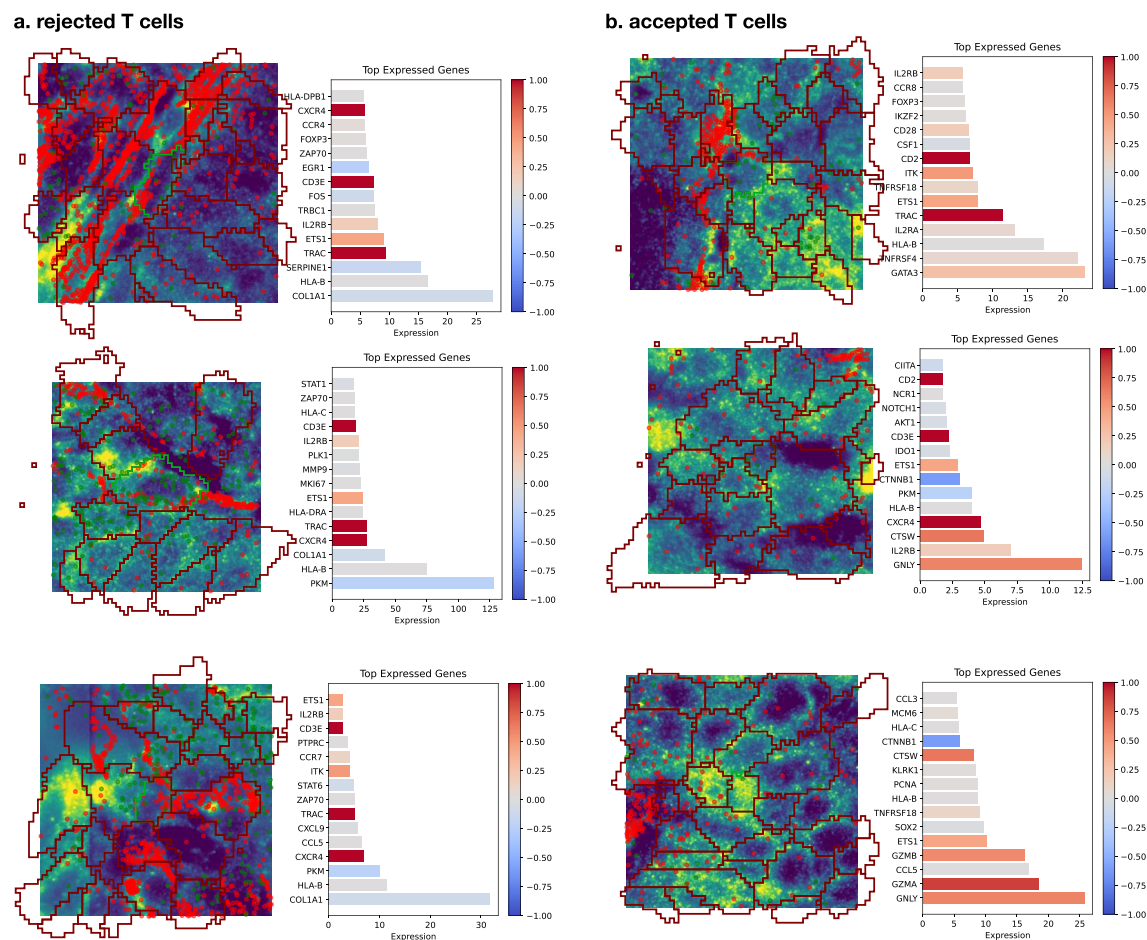


Figure 6.7: **Cells with their manual annotations in the main lung slide.** **a.** Cells annotated as T cells by the ML pipeline and rejected by the manual annotator. **b.** Cells annotated as T cells by the ML pipeline and accepted by the manual annotator. In both cases, we display three examples of cells from the same tissue slide. Each example shows cell nuclei (DAPI) along with the proposed ML segmentation overlay, as well as the top expressed genes in the cell.

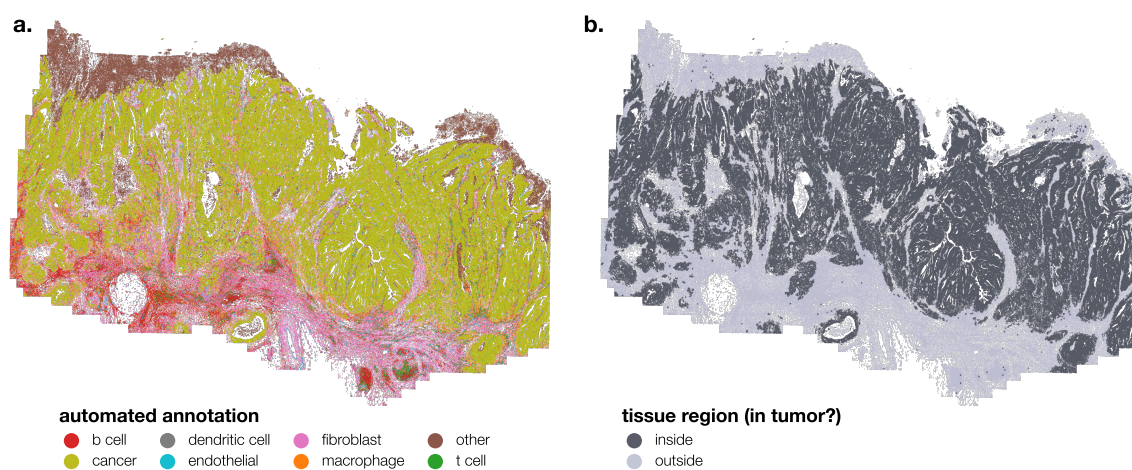


Figure 6.8: **Visualization of cells in the main colon slide**, colored by (a) ML-derived cell-type and (b) tumor region assignments.

6.6 Supplementary Tables

Table 6.1: Markers employed to label clusters.

Cell-type	Genes
Tumor	EPCAM, EGFR, VEGFA, HIF1A, MUC1, CEACAM1
T cell	CD3E, CD3D, CD3G, CD4, CD8A, CD8B
NK cell	NCAM1, NKG7, KLRK1, GNLY, PRF1
Dendritic cells	CD1C, CLEC4C, CD83, LAMP3, CD209
Fibroblast	COL1A1, FAP, PDGFRA
Neutrophil	S100A9, FCGR3A, CXCR2, ELANE, MPO
Endothelial	PECAM1, CDH5, VWF, PLVAP
Macrophage	CD68, CD163, MRC1, MSR1
B cell	CD19, CD79A, MS4A1

Table 6.2: Cell counts from ML-derived segmentation and classification. The table shows the total number of cells and the subset of T cells identified by our pipeline across four tissue slides.

Slide	Total cells	T cells (%)
Lung 1	351,309	19,867 (5.7%)
Lung 2	825,527	73,374 (8.9%)
Colon 1	671,075	5,390 (0.8%)
Colon 2	806,291	35,559 (4.4%)

Table 6.3: List of non-T cell marker genes employed in the analysis.

Category	Genes
Myeloid and DC	CD14, CD163, FCGR3A, MARCO, MRC1, TREM2, S100A9, LYZ, CYBB, CSF1, CSF1R, CD83, CD1C, CD1B, CD1E, LAMP3, CD207, CLEC4C, SIGLEC1, CD209
Stromal	COL1A1, PECAM1, COL4A1, COL5A1, COL6A3, COL11A1, FN1, LAMB3, LAMC2, FBLN1, TNC, ACTA2, FAP, DES
Endothelial	VWF, CDH5, CLDN5, PLVAP, MMRN1, MMRN2, CLEC14A, KDR, TEK, VEGFB, VEGFA, VEGFC, ANGPT1, ANGPT2, NRP1
Epithelial	EPCAM, CDH1, MUC1, MUC4
Antigen presentation	HLA-DRA, HLA-DRB1, HLA-DPB1, HLA-DMA, HLA-DPA1, HLA-DQA1, CD276, LGALS9, PDCD1LG2, NCR1, NCR3, KLRG1, XBP1

Chapter 7

Conclusion

Analyzing single-cell omics data at scale has required the development of sophisticated computational methods, often developed at the expense of direct interpretability. These methods have, however, devised efficient strategies to study biological variation under technical confounding. This dissertation presents efforts to address the challenge of deploying the strength of these models in applications requiring rigorous and interpretable biological interpretation.

The development of interpretable analytical frameworks, built upon latent variable models, to study molecular variation in various single-cell omics settings is the main contribution of this dissertation. Applied to DE analysis, we developed a statistical framework for DE analysis, producing reliable and interpretable inferences of DE effects. This tool, lvm-DE, relies on three steps. It assumes a latent variable model has been trained and produces cell-specific normalized gene expressions. To quantify a given population's differential expression, lvm-DE posits empirical Bayes priors on cell group-specific normalized gene expressions. To identify differentially expressed genes, lvm-DE compares the expressions of the different groups and provides posterior DE probabilities, which we use to identify differentially expressed genes with statistical guarantees. This framework, integrated into the `scvi-tools` software, has already been successfully applied in several single-cell studies, e.g., to study CD8 T cell residency [273], macrophage residency [274], or type 2 diabetes [275].

In parallel to multi-omics, cohort-scale single-cell omics shifted the analytical focus from individual cells to biological variation across samples. Just like early single-cell omics provided new strategies to map cell state variation, one hope for cohort-scale single-cell omics is that they will improve our understanding of sample-level variation. This hope is already a reality in oncology, for instance, where transcriptomic signatures provide powerful tools for prognosis. It is with this perspective that we designed

MrVI. MrVI is a latent variable model designed for multi-sample single-cell data. Built upon the same principles as earlier latent variable models for single-cell data, it learns two cell representations while accounting for technical confounding, e.g., coming from the experimental design. MrVI uses latent variable modeling to learn the transcriptional effects of sample contexts on continuous cell states. Once the model is fit, it can be used to characterize the effects of any sample covariate on gene expression and abundance of any given continuous cell state. Critically, it can also be used to understand what sample axes are driving gene expression variation.

The ability to study the effect of known sample covariates, and to identify relevant sample stratifications, has already supported the analysis of cohort-scale single-cell omics studies, characterizing tumor-associated macrophages [276], immune cell dynamics in glioblastoma [277], and with aging [278].

A natural extension of this work is to study cohort-scale multi-omics data, where an open question is to understand whether sample stratifications differ across omics layers, and whether these differences have biological support. Extending MrVI to such multi-omics settings is an important effort, not only to answer these questions, but also to facilitate the integration of highly heterogeneous multi-omics data, which has been a challenge due to strong confounding effects [278].

My work then focused on the challenge of evaluating model performance in single-cell omics. Selecting appropriate predictive models and understanding their expected performance is crucial for optimizing analysis pipelines. This evaluation challenge is particularly acute in single-cell omics due to the proliferation of tools and analytical tasks, made even more difficult by the scarcity of reliable ground-truth required for evaluation - a situation likely to become more complex as the field grows on both the assay and the computational side. AutoEval addresses this challenge by combining a small amount of human data and a large amount of synthetic data to get better estimates of model performance.

This framework applies beyond omics. We explored its relevance in fields like natural language processing (NLP), where model evaluation faces similar challenges regarding ground-truth data availability. A relevant observation from NLP is the increasing use of preference-based model evaluation. In this paradigm, human annotators give preferences over paired model outputs, which can be a more efficient way to compare models than constructing large and high-quality validation datasets. This paradigm seems particularly relevant in single-cell omics, where generating comprehensive validation datasets via manual cell labeling is slow, error-prone, and susceptible to annotation inconsistencies across different studies or experts. Given these challenges, adopting preference-based evaluation strategies, built upon community-driven benchmarking efforts, could constitute an efficient solution for model selection in

single-cell omics.

Finally, this dissertation addressed the challenge of performing reliable differential expression (DE) analysis between spatially-resolved cell populations in spatial transcriptomics. Because of the massive amount of data that these assays produce, black-box predictive models are taking a necessary and irreplaceable role to process data at scale, and in particular to segment cells. However, as these models are inevitably imperfect, a significant proportion of transcripts from neighboring cells are misassigned to the target cell type, leading to systematic biases that confound spatial DE analysis. The problem of understanding the impact of a preprocessing step on inference is canonical in computational biology; however, this challenge is particularly critical in spatial omics. While preprocessing typically affects the statistical guarantees of the analysis, e.g., by inflating type I error rates or effect size estimates, imperfect segmentation introduces many spurious discoveries that may dramatically misguide data interpretation. CSDE addresses this challenge by proposing a new framework to conduct spatially-resolved DE analysis while remaining robust to segmentation errors. Leveraging prediction-powered inference, it requires a small amount of ground-truth data, that is, access to manually segmented cells or properly segmented ones, to identify differentially expressed genes with statistical guarantees.

As illustrated by spatial transcriptomics, future high-throughput omics technologies will inevitably incorporate sophisticated machine learning models to perform complex processing tasks, including feature extraction, normalization, and technical correction, essential for ambitious biological discovery. This increasing reliance on black-box predictive models is an obstacle to the reliability and reproducibility of biological insights obtained from these assays. One step in this direction is to proactively rethink assay design by understanding the potential failure modes of the employed predictive models and strategically identifying where targeted human curation or validation is most critical. A statistical assessment of predictive model reliability, using strategies such as CSDE, is consequently necessary to guarantee the scientific validity of insights derived from these integrated experimental-computational approaches.

Bibliography

1. Trapnell, C. Defining Cell Types and States with Single-Cell Genomics. *Genome Research* **25**, 1491–1498 (Oct. 2015).
2. Haber, A. L. *et al.* A Single-Cell Survey of the Small Intestinal Epithelium. *Nature* **551**, 333–339 (Nov. 2017).
3. Han, X. *et al.* Mapping the Mouse Cell Atlas by Microwell-Seq. *Cell* **172**, 1091–1107.e17 (Feb. 2018).
4. Schaum, N. *et al.* Single-Cell Transcriptomics of 20 Mouse Organs Creates a Tabula Muris. *Nature* **562**, 367–372 (Oct. 2018).
5. Zeisel, A. *et al.* Cell Types in the Mouse Cortex and Hippocampus Revealed by Single-Cell RNA-seq. *Science* **347**, 1138–1142 (Mar. 2015).
6. Villani, A.-C. *et al.* Single-Cell RNA-seq Reveals New Types of Human Blood Dendritic Cells, Monocytes, and Progenitors. *Science* **356**, eaah4573 (Apr. 2017).
7. Steier, Z. *et al.* Single-cell multiomic analysis of thymocyte development reveals drivers of CD4+ T cell and CD8+ T cell lineage commitment. *Nature Immunology* **24**, 1579–1590 (2023).
8. Wagner, A., Regev, A. & Yosef, N. Revealing the vectors of cellular identity with single-cell genomics. *Nature Biotechnology* **34**, 1145–1160 (Nov. 2016).
9. The Tabula Sapiens consortium *et al.* The Tabula Sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. *Science* **376**, eabl4896 (2022).
10. Yazar, S. *et al.* Single-cell eQTL mapping identifies cell type-specific genetic control of autoimmune disease. *Science* **376**, eabf3041 (Apr. 2022).
11. Stoeckius, M. *et al.* Cell Hashing with Barcoded Antibodies Enables Multiplexing and Doublet Detection for Single Cell Genomics. *Genome Biology* **19**, 224 (Dec. 2018).

12. Stoeckius, M. *et al.* Simultaneous epitope and transcriptome measurement in single cells. *Nature Methods* **14**, 865–868 (Sept. 2017).
13. Cao, J. *et al.* Joint Profiling of Chromatin Accessibility and Gene Expression in Thousands of Single Cells. *Science* **361**, 1380–1385 (Sept. 2018).
14. Kong, L. *et al.* The landscape of immune dysregulation in Crohn’s disease revealed through single-cell transcriptomic profiling in the ileum and colon. *Immunity* **56**, 444–458 (2023).
15. Srivatsan, S. R. *et al.* Massively multiplex chemical transcriptomics at single-cell resolution. *Science* **367**, 45–51 (Jan. 2020).
16. Stephenson, E. *et al.* Single-cell multi-omics analysis of the immune response in COVID-19. *Nature Medicine* **27**, 904–916 (May 2021).
17. Yoshida, M. *et al.* Local and Systemic Responses to SARS-CoV-2 Infection in Children and Adults. *Nature* **602**, 321–327 (Feb. 2022).
18. Luecken, M. D. & Theis, F. J. Current Best Practices in Single-cell RNA-seq Analysis: A Tutorial. *Molecular Systems Biology* **15**, e8746 (June 2019).
19. Stuart, T. *et al.* Comprehensive integration of single-cell data. *Cell* **177**, 1888–1902.e21 (June 2019).
20. Haghverdi, L., Lun, A. T. L., Morgan, M. D. & Marioni, J. C. Batch Effects in Single-Cell RNA-sequencing Data Are Corrected by Matching Mutual Nearest Neighbors. *Nature Biotechnology* **36**, 421–427 (June 2018).
21. Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S. & Vert, J.-P. A General and Flexible Method for Signal Extraction from Single-Cell RNA-seq Data. *Nature Communications* **9**, 284 (Jan. 2018).
22. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nature Methods* **15**, 1053–1058 (Dec. 2018).
23. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *arXiv [Preprint]* (2013).
24. Luecken, M. D. *et al.* Benchmarking Atlas-Level Data Integration in Single-Cell Genomics. *Nature Methods* **19**, 41–50 (Jan. 2022).
25. Ding, J. & Regev, A. Deep generative model embedding of single-cell RNA-Seq profiles on hyperspheres and hyperbolic spaces. *Nature Communications* **12**, 2554 (May 2021).

26. Xu, C. *et al.* Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Molecular Systems Biology* **17**, e9620 (2021).
27. Lotfollahi, M., Wolf, F. A. & Theis, F. J. scGen predicts single-cell perturbation responses. *Nature Methods* **16**, 715–721 (July 2019).
28. Ashuach, T., Reidenbach, D. A., Gayoso, A. & Yosef, N. PeakVI: A Deep Generative Model for Single-Cell Chromatin Accessibility Analysis. *Cell Reports Methods* **2**, 100182 (Mar. 2022).
29. Lopez, R. *et al.* DestVI Identifies Continuums of Cell Types in Spatial Transcriptomics Data. *Nature Biotechnology* **40**, 1360–1369 (Sept. 2022).
30. Gayoso, A. *et al.* Joint probabilistic modeling of single-cell multi-omic data with totalVI. *Nature Methods* **18**, 272–282 (Feb. 2021).
31. Ashuach, T. *et al.* MultiVI: Deep Generative Model for the Integration of Multimodal Data. *Nature Methods* **20**, 1222–1231 (June 2023).
32. Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26**, 139–140 (2010).
33. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* **15**, 550 (2014).
34. Robins, J. M. & Rotnitzky, A. Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association* **90**, 122–129 (1995).
35. Leek, J. T. *et al.* Tackling the Widespread and Critical Impact of Batch Effects in High-Throughput Data. *Nature Reviews Genetics* **11**, 733–739 (Oct. 2010).
36. Berger, J. O. *Statistical decision theory and Bayesian analysis* (Springer Science & Business Media, 2013).
37. Benjamini, Y. & Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *The Journal of the Royal Statistical Society, Series B* **57**, 289–300 (1995).
38. Subramanian, A. *et al.* Gene Set Enrichment Analysis: A Knowledge-Based Approach for Interpreting Genome-Wide Expression Profiles. *Proceedings of the National Academy of Sciences* **102**, 15545–15550 (Oct. 2005).
39. Candès, E., Fan, Y., Janson, L. & Lv, J. Panning for gold: ‘model-X’ knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **80**, 551–577 (2018).

40. Smillie, C. S. *et al.* Intra- and Inter-cellular Rewiring of the Human Colon during Ulcerative Colitis. *Cell* **178**, 714–730.e22 (July 2019).
41. Maity, A. K. & Teschendorff, A. E. Cell-attribute aware community detection improves differential abundance testing from single-cell RNA-Seq data. *Nature Communications* **14**, 3244 (2023).
42. Hoadley, K. A. *et al.* Cell-of-origin patterns dominate the molecular classification of 10,000 tumors from 33 types of cancer. *Cell* **173**, 291–304 (2018).
43. Svensson, V., Vento-Tormo, R. & Teichmann, S. A. Exponential Scaling of Single-Cell RNA-seq in the Past Decade. *Nature Protocols* **13**, 599–604 (Apr. 2018).
44. Zappia, L. & Theis, F. J. Over 1000 Tools Reveal Trends in the Single-Cell RNA-seq Analysis Landscape. *Genome Biology* **22**, 1–18 (Dec. 2021).
45. Heumos, L. *et al.* Best Practices for Single-Cell Analysis across Modalities. *Nature Reviews Genetics* **24**, 550–572 (Aug. 2023).
46. Kharchenko, P. V. The Triumphs and Limitations of Computational Methods for scRNA-seq. *Nature Methods* **18**, 723–732 (July 2021).
47. Crowell, H. L., Morillo Leonardo, S. X., Soneson, C. & Robinson, M. D. The Shaky Foundations of Simulating Single-Cell RNA Sequencing Data. *Genome Biology* **24**, 1–19 (Dec. 2023).
48. Angelopoulos, A. N., Bates, S., Fannjiang, C., Jordan, M. I. & Zrnic, T. Prediction-Powered Inference. *Science* **382**, 669–674 (Nov. 2023).
49. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv [cs.CV]* (May 2015).
50. Pachitariu, M. & Stringer, C. Cellpose 2.0: How to Train Your Own Model. *Nature Methods* **19**, 1634–1641 (Dec. 2022).
51. Mitchel, J., Gao, T., Cole, E., Petukhov, V. & Kharchenko, P. V. Impact of Segmentation Errors in Analysis of Spatial Transcriptomics Data, 2025.01.02.631135 (Jan. 2025).
52. Lopez, R., Boyeau, P., Yosef, N., Jordan, M. & Regier, J. *Decision-Making with Auto-Encoding Variational Bayes* in *Advances in Neural Information Processing Systems* **33** (Curran Associates, Inc., 2020), 5081–5092.
53. Gayoso, A. *et al.* A Python library for probabilistic analysis of single-cell omics data. *Nature Biotechnology* **40**, 163–166 (Feb. 2022).
54. Prillo, S. *et al.* CherryML: Scalable Maximum Likelihood Estimation of Phylogenetic Models. *Nature Methods* **20**, 1232–1236 (Aug. 2023).

55. Finak, G. *et al.* MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biology* **16**, 1–13 (2015).
56. Kharchenko, P. V., Silberstein, L. & Scadden, D. T. Bayesian approach to single-cell differential expression analysis. *Nature Methods* **11**, 740–742 (2014).
57. Satija, R., Farrell, J. A., Gennert, D., Schier, A. F. & Regev, A. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology* **33**, 495–502 (2015).
58. Squair, J. W. *et al.* Confronting false discoveries in single-cell differential expression. *bioRxiv [Preprint]* (2021).
59. Johnson, W. E., Li, C. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* **8**, 118–127. eprint: <https://academic.oup.com/biostatistics/article-pdf/8/1/118/25435561/kxj037.pdf> (Apr. 2006).
60. Lopez, R., Gayoso, A. & Yosef, N. Enhancing scientific discoveries in molecular biology with deep generative models. *Molecular Systems Biology* **16**, e9198. eprint: <https://www.embopress.org/doi/pdf/10.15252/msb.20199198> (2020).
61. Fleming, S. J., Marioni, J. C. & Babadi, M. CellBender remove-background: a deep generative model for unsupervised removal of background noise from scRNA-seq datasets. *bioRxiv [Preprint]*, 791699 (2019).
62. Grün, D., Kester, L. & Van Oudenaarden, A. Validation of noise models for single-cell transcriptomics. *Nature Methods* **11**, 637–640 (2014).
63. Soneson, C. & Robinson, M. D. Bias, robustness and scalability in single-cell differential expression analysis. *Nature Methods* **15**, 255 (2018).
64. Law, C. W., Chen, Y., Shi, W. & Smyth, G. K. voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology* **15**, 1–17 (2014).
65. Ma, X., Korthauer, K., Kendzierski, C. & Newton, M. A. A compositional model to assess expression changes from single-cell RNA-seq data. *Annals of Applied Statistics* **15**, 880–901 (June 2021).
66. Qiu, P. Embracing the dropouts in single-cell RNA-seq analysis. *Nature Communications* **11**, 1–9 (2020).
67. Svensson, V., da Veiga Beltrame, E. & Pachter, L. A curated database reveals trends in single-cell transcriptomics. *Database* (2020).

68. Rashid, S., Shah, S., Bar-Joseph, Z. & Pandya, R. Dhaka: variational autoencoder for unmasking tumor heterogeneity from single cell genomic data. *Bioinformatics* (2019).
69. Wang, D. & Gu, J. VASC: dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder. *Genomics, Proteomics & Bioinformatics* **16**, 320–331 (2018).
70. Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature Communications* **10**, 390 (2019).
71. Grønbech, C. H. *et al.* scVAE: Variational auto-encoders for single-cell gene expression data. *Bioinformatics* **36**, 4415–4422 (2020).
72. Amodio, M. *et al.* Exploring single-cell data with deep multitasking neural networks. *Nature Methods* **16**, 1139–1145 (Nov. 2019).
73. Lotfollahi, M., Naghipourfar, M., Theis, F. J. & Wolf, F. A. Conditional Out-of-Distribution Generation for Unpaired Data Using Transfer VAE. *Bioinformatics* **36**, i610–i617 (Dec. 2020).
74. Clivio, O. *et al.* Detecting zero-inflated genes in single-cell transcriptomics data. *bioRxiv [Preprint]*, 794875 (2019).
75. Svensson, V. Droplet scRNA-seq is not zero-inflated. *Nature Biotechnology* **38**, 147–150 (Feb. 2020).
76. Andrews, T. S. & Hemberg, M. False signals induced by single-cell imputation. *F1000Research* **7** (2018).
77. Svensson, V., Gayoso, A., Yosef, N. & Pachter, L. Interpretable factor models of single-cell RNA-seq via variational autoencoders. *Bioinformatics* **36**, 3418–3421 (June 2020).
78. Nijkamp, E. *et al.* Learning multi-layer latent variable model via variational optimization of short run mcmc for approximate inference in *European Conference on Computer Vision* (2020), 361–378.
79. Domke, J. & Sheldon, D. R. *Importance Weighting and Variational Inference* in *Advances in Neural Information Processing Systems* (eds Bengio, S. *et al.*) **31** (Curran Associates, Inc., 2018).
80. Yao, Y., Vehtari, A., Simpson, D. & Gelman, A. Yes, but did it work?: Evaluating variational inference in *International Conference on Machine Learning* (2018), 5581–5590.

81. Newton, M. A., Kendzierski, C. M., Richmond, C. S., Blattner, F. R. & Tsui, K. W. On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data. *Journal of Computational Biology* **8**, 37–52 (2001).
82. Feng, J. *et al.* GFOLD: a generalized fold change for ranking differentially expressed genes from RNA-seq data. *Bioinformatics* **28**, 2782–2788 (2012).
83. Kapourani, C.-A., Argelaguet, R., Sanguinetti, G. & Vallejos, C. A. scMET: Bayesian modelling of DNA methylation heterogeneity at single-cell resolution. *bioRxiv [Preprint]* (2020).
84. Newton, M. A., Noueiry, A., Sarkar, D. & Ahlquist, P. Detecting differential gene expression with a semiparametric hierarchical mixture method. *Biostatistics* **5**, 155–176 (2004).
85. Lee, J., Ji, Y., Liang, S., Cai, G. & Müller, P. On differential gene expression using RNA-Seq data. *Cancer Informatics* **10**, CIN-S7473 (2011).
86. Cui, S., Guha, S., Ferreira, M. A., Tegge, A. N., *et al.* hmmSeq: A hidden Markov model for detecting differentially expressed genes from RNA-seq data. *Annals of Applied Statistics* **9**, 901–925 (2015).
87. Zhang, X., Xu, C. & Yosef, N. Simulating multiple faceted variability in single cell RNA sequencing. *Nature Communications* **10**, 1–16 (2019).
88. Korthauer, K. D. *et al.* A statistical approach for identifying differential distributions in single-cell RNA-seq experiments. *Genome Biology* **17**, 1–15 (2016).
89. Crowell, H. L. *et al.* muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data. *Nature Communications* **11**, 1–12 (2020).
90. Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nature Communications* **8**, 14049 (Jan. 2017).
91. Monaco, G. *et al.* RNA-Seq signatures normalized by mRNA abundance allow absolute deconvolution of human immune cell types. *Cell Reports* **26**, 1627–1640 (2019).
92. Abdelaal, T. *et al.* A comparison of automatic cell identification methods for single-cell RNA sequencing data. *Genome Biology* **20**, 1–19 (2019).
93. Wilk, A. J. *et al.* A single-cell atlas of the peripheral immune response in patients with severe COVID-19. *Nature Medicine* **26**, 1070–1076 (2020).

94. Rahmani, E., Jordan, M. I. & Yosef, N. Identifying systematic variation at the single-cell level by leveraging low-resolution population-level data. *bioRxiv [Preprint]* (2022).
95. Fara, A., Mitrev, Z., Rosalia, R. A. & Assas, B. M. Cytokine storm and COVID-19: a chronicle of pro-inflammatory cytokines. *Open Biology* **10**, 200160 (2020).
96. Heng, T. S. *et al.* The Immunological Genome Project: networks of gene expression in immune cells. *Nature Immunology* **9**, 1091–1094 (2008).
97. Ntranos, V., Yi, L., Melsted, P. & Pachter, L. A discriminative learning approach to differential expression analysis for single-cell RNA-seq. *Nature Methods* **16**, 163–166 (2019).
98. Quake, S. R., Consortium, T. S., *et al.* The Tabula Sapiens: a single cell transcriptomic atlas of multiple organs from individual human donors. *bioRxiv [Preprint]* (2021).
99. Schiller, H. B. *et al.* The human lung cell atlas: a high-resolution reference map of the human lung in health and disease. *American journal of respiratory cell and molecular biology* **61**, 31–41 (2019).
100. Lotfollahi, M. *et al.* Mapping single-cell data to reference atlases by transfer learning. *Nature Biotechnology* **40**, 121–130 (Aug. 2021).
101. Kriegeskorte, N., Simmons, W. K., Bellgowan, P. S. & Baker, C. I. Circular analysis in systems neuroscience: the dangers of double dipping. *Nature Neuroscience* **12**, 535–540 (2009).
102. Batson, J., Royer, L. & Webber, J. Molecular cross-validation for single-cell RNA-seq. *bioRxiv [Preprint]*, 786269 (2019).
103. Krishnan, R., Liang, D. & Hoffman, M. *On the challenges of learning with inference networks on sparse, high-dimensional data* in *International Conference on Artificial Intelligence and Statistics* (2018), 143–151.
104. Lotfollahi, M. *et al.* Biologically informed deep learning to infer gene program activity in single cells. *bioRxiv [Preprint]* (2022).
105. Stephens, M. False discovery rates: a new deal. *Biostatistics* **18**, 275–294 (Oct. 2016).
106. Lee, J., Hwang, D., *et al.* Single-cell multiomics: technologies and data analysis methods. *Experimental & Molecular Medicine* **52**, 1428–1442 (2020).
107. Clark, S. J. *et al.* scNMT-seq enables joint profiling of chromatin accessibility DNA methylation and transcription in single cells. *Nature Communications* **9**, 1–9 (2018).

108. De Cao, N. & Aziz, W. The power spherical distribution. *arXiv [Preprint]* (2020).
109. Smith, L. N. *Cyclical learning rates for training neural networks* in *2017 IEEE winter conference on applications of computer vision (WACV)* (2017), 464–472.
110. Robinson, M. D. & Oshlack, A. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology* **11**, 1–9 (2010).
111. Lareau, C. A. *et al.* Droplet-based combinatorial indexing for massive-scale single-cell chromatin accessibility. *Nature Biotechnology* **37**, 916–924 (Aug. 2019).
112. Chen, A. *et al.* Spatiotemporal transcriptomic atlas of mouse organogenesis using DNA nanoball-patterned arrays. *Cell* **185**, 1777–1792.e21 (May 2022).
113. Wang, G., Moffitt, J. R. & Zhuang, X. Author Correction: Multiplexed imaging of high-density libraries of RNAs with MERFISH and expansion microscopy. *Scientific Reports* **8**, 6487 (Apr. 2018).
114. Tornow, S. & Mewes, H. W. Functional modules by relating protein interaction networks and gene expression. *Nucleic Acids Research* **31**, 6283–6289 (Nov. 2003).
115. Moses, L. & Pachter, L. Museum of spatial transcriptomics. *Nature Methods*, 1–13 (2022).
116. Segal, E. *et al.* Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics* **34**, 166–176 (May 2003).
117. Iancu, O. D. *et al.* Utilizing RNA-Seq data for de novo coexpression network inference. *Bioinformatics* **28**, 1592–1597 (June 2012).
118. Hu, R., Qiu, X., Glazko, G., Klebanov, L. & Yakovlev, A. Detecting intergene correlation changes in microarray analysis: a new approach to gene selection. *BMC Bioinformatics* **10**, 1–9 (2009).
119. Van Dam, S., Vösa, U., van der Graaf, A., Franke, L. & de Magalhães, J. P. Gene co-expression analysis for functional classification and gene–disease predictions. *Briefings in Bioinformatics* **19**, 575–592 (Jan. 2017).
120. Yang, L. *et al.* scMAGeCK links genotypes with multiple phenotypes in single-cell CRISPR screens. *Genome Biology* **21**, 19 (Jan. 2020).
121. *Causal Inference for Statistics, Social, and Biomedical Sciences* (Cambridge University Press, 2015).

122. Gillis, J. & Pavlidis, P. “Guilt by association” is the exception rather than the rule in gene networks. *PLoS Computational Biology* **8**, e1002444 (2012).
123. Margolin, A. A. *et al.* ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* **7**, S7 (Mar. 2006).
124. Lee, H. K., Hsu, A. K., Sajdak, J., Qin, J. & Pavlidis, P. Coexpression analysis of human genes across many microarray data sets. *Genome Research* **14**, 1085–1094 (June 2004).
125. Huynh-Thu, V. A., Irrthum, A., Wehenkel, L. & Geurts, P. Inferring regulatory networks from expression data using tree-based methods. *PLoS One* **5** (Sept. 2010).
126. Moerman, T. *et al.* GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks. *Bioinformatics* **35**, 2159–2161 (June 2019).
127. Kim, S. ppcor: An R package for a fast calculation to semi-partial correlation coefficients. *Communications for Statistical Applications and Methods* **22**, 665–674 (Nov. 2015).
128. Chan, T. E., Stumpf, M. P. H. & Babbie, A. C. Gene Regulatory Network Inference from Single-Cell Data Using Multivariate Information Measures. *Cell Systems* **5**, 251–267.e3 (Sept. 2017).
129. Qiu, X. *et al.* Inferring causal gene regulatory networks from coupled Single-Cell expression dynamics using Scribe. *Cell Systems* **10**, 265–274.e11 (Mar. 2020).
130. Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society* **58**, 267–288 (Jan. 1996).
131. Meinshausen, N. & Bühlmann, P. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics* **34**, 1436–1462 (June 2006).
132. Zou, H. & Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society* **67**, 301–320 (Apr. 2005).
133. Pratapa, A., Jalihal, A. P., Law, J. N., Bharadwaj, A. & Murali, T. M. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature Methods* **17**, 147–154 (Feb. 2020).
134. Peters, J., Janzing, D. & Schölkopf, B. *Elements of Causal Inference: Foundations and Learning Algorithms* (The MIT Press, 2017).
135. Krishnaswamy, S. *et al.* Systems biology. Conditional density-based analysis of T cell signaling in single-cell data. *Science* **346**, 1250689 (Nov. 2014).

136. Melenhorst, J. J. *et al.* Decade-long leukaemia remissions with persistence of CD4+ CAR T cells. *Nature* **602**, 503–509 (2022).
137. Sacco, K. *et al.* Immunopathological signatures in multisystem inflammatory syndrome in children and pediatric COVID-19. *Nature Medicine* **28**, 1050–1062 (2022).
138. Van de Sande, B. *et al.* A scalable SCENIC workflow for single-cell gene regulatory network analysis. *Nature Protocols* **15**, 2247–2276 (2020).
139. Wasserman, L. in *All of Statistics: A Concise Course in Statistical Inference* chapter 13.5 (Springer Science & Business Media, 2013).
140. Sesia, M., Katsevich, E., Bates, S., Candès, E. & Sabatti, C. Multi-resolution localization of causal variants across the genome. *Nature Communications* **11**, 1093 (2020).
141. DeTomaso, D. & Yosef, N. Hotspot identifies informative gene modules across modalities of single-cell genomics. *Cell Systems* **12**, 446–456 (2021).
142. Schaid, D. J., Chen, W. & Larson, N. B. From genome-wide associations to candidate causal variants by statistical fine-mapping. *Nature Reviews Genetics* **19**, 491–504 (2018).
143. Liberzon, A. *et al.* The molecular signatures database hallmark gene set collection. *Cell Systems* **1**, 417–425 (2015).
144. Frangieh, C. J. *et al.* Multimodal pooled Perturb-CITE-seq screens in patient models define mechanisms of cancer immune evasion. *Nature Genetics* **53**, 332–341 (Mar. 2021).
145. Hao, Y. *et al.* Integrated analysis of multimodal single-cell data. *Cell* **184**, 3573–3587.e29 (June 2021).
146. Franceschini, A. *et al.* STRING v9. 1: protein-protein interaction networks, with increased coverage and integration. *Nucleic acids research* **41**, D808–D815 (2012).
147. Léveillé, C., AL-Daccak, R. & Mourad, W. CD20 is physically and functionally coupled to MHC class II and CD40 on human B cell lines. *European Journal of Immunology* **29**, 65–74 (Jan. 1999).
148. Zeng, J., Liu, R., Wang, J. & Fang, Y. A bispecific antibody directly induces lymphoma cell death by simultaneously targeting CD20 and HLA-DR. *Journal of Cancer Research and Clinical Oncology* **141**, 1899–1907 (Nov. 2015).

149. Claus, M., Wingert, S. & Watzl, C. Modulation of natural killer cell functions by interactions between 2B4 and CD48 in cis and in trans. *Open Biology* **6**, 160010 (May 2016).
150. Jassal, B. *et al.* The reactome pathway knowledgebase. *Nucleic Acids Research* **48**, D498–D503 (2020).
151. Camargo, J. F. *et al.* CCR5 expression levels influence NFAT translocation, IL-2 production, and subsequent signaling events during T lymphocyte activation. *The Journal of Immunology* **182**, 171–182 (Jan. 2009).
152. Oh, H.-M. *et al.* STAT3 protein promotes T-cell survival and inhibits interleukin-2 production through up-regulation of Class O Forkhead transcription factors. *Journal of Biological Chemistry* **286**, 30888–30897 (Sept. 2011).
153. Mahmud, S. A., Manlove, L. S. & Farrar, M. A. Interleukin-2 and STAT5 in regulatory T cell development and function. *JAKSTAT* **2**, e23154 (Jan. 2013).
154. He, S. *et al.* High-plex multiomic analysis in FFPE tissue at single-cellular and subcellular resolution by spatial molecular imaging. *bioRxiv [Preprint]*, 2021–11 (2021).
155. Petukhov, V. *et al.* Cell segmentation in imaging-based spatial transcriptomics. *Nature Biotechnology* **40**, 345–354 (Mar. 2022).
156. Boyeau, P. *et al.* An empirical Bayes method for differential expression analysis of single cells with deep generative models. *Proceedings of the National Academy of Sciences* **120**, e2209124120 (2023).
157. Hu, X. *et al.* ITGAE defines CD8⁺ tumor-infiltrating lymphocytes predicting a better prognostic survival in colorectal cancer. *EBioMedicine* **35**, 178–188 (Sept. 2018).
158. Garcia-Carbonero, R., Carnero, A. & Paz-Ares, L. Inhibition of HSP90 molecular chaperones: moving into the clinic. *The Lancet Oncology* **14**, e358–e369 (2013).
159. Esfahani, K. & Cohen, V. HSP90 as a novel molecular target in non-small-cell lung cancer. *Lung Cancer: Targets and Therapy* **7**, 11–17 (2016).
160. Yuan, Z., Wang, L. & Chen, C. Analysis of the prognostic, diagnostic and immunological role of HSP90 α in malignant tumors. *Frontiers in Oncology* **12**, 963719 (2022).
161. Nagy, N. *et al.* In and out of the bursa—the role of CXCR4 in chicken B cell development. *Frontiers in Immunology* **11**, 1468 (2020).

162. Li, G. *et al.* TGF- β -dependent lymphoid tissue residency of stem-like T cells limits response to tumor vaccine. *Nature Communications* **13**, 6043 (2022).
163. Bai, Y., Hu, M., Chen, Z., Wei, J. & Du, H. Single-cell transcriptome analysis reveals RGS1 as a new marker and promoting factor for T-cell exhaustion in multiple cancers. *Frontiers in Immunology* **12**, 767070 (2021).
164. Hastie, T., Tibshirani, R., Friedman, J. H. & Friedman, J. H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, 2009).
165. Li, S., Sesia, M., Romano, Y., Candès, E. & Sabatti, C. Searching for robust associations with a multi-environment knockoff filter. *Biometrika* **109**, 611–629 (2022).
166. Heinze-Deml, C., Peters, J. & Meinshausen, N. Invariant causal prediction for nonlinear models. *Journal of Causal Inference* **6**, 20170016 (2018).
167. Wagner, A. *et al.* Metabolic modeling of single Th17 cells reveals regulators of autoimmunity. *Cell* **184**, 4168–4185 (2021).
168. Cang, Z. *et al.* Screening cell–cell communication in spatial transcriptomics via collective optimal transport. *Nature Methods* **20**, 218–228 (2023).
169. Vergara, H. M. *et al.* Whole-body integration of gene expression and single-cell morphology. *Cell* **184**, 4819–4837 (2021).
170. Treppner, M. *et al.* Synthetic single cell RNA sequencing data from small pilot studies using deep generative models. *Scientific Reports* **11**, 9403 (2021).
171. Martens, L. D., Fischer, D. S., Yépez, V. A., Theis, F. J. & Gagneur, J. Modeling fragment counts improves single-cell ATAC-seq analysis. *Nature Methods* **21**, 28–31 (2024).
172. Tung, P.-Y. *et al.* Batch effects and the effective design of single-cell gene expression studies. *Scientific Reports* **7**, 39921 (2017).
173. Barry, T., Wang, X., Morris, J. A., Roeder, K. & Katsevich, E. SCEPTRE improves calibration and sensitivity in single-cell CRISPR screen analysis. *Genome Biology* **22**, 344 (Dec. 2021).
174. Rozenblatt-Rosen, O., Stubbington, M. J., Regev, A. & Teichmann, S. A. The Human Cell Atlas: from vision to reality. *Nature* **550**, 451–453 (2017).
175. Sanchez-Vega, F. *et al.* Oncogenic signaling pathways in the cancer genome atlas. *Cell* **173**, 321–337 (2018).
176. Consortium, G. Genetic effects on gene expression across human tissues. *Nature* **550**, 204–213 (2017).

177. Kang, H. M. *et al.* Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nature Biotechnology* **36**, 89–94 (Jan. 2018).
178. McGinnis, C. S. *et al.* MULTI-seq: sample multiplexing for single-cell RNA sequencing using lipid-tagged indices. *Nature Methods* **16**, 619–626 (July 2019).
179. Tran, V. *et al.* High sensitivity single cell RNA sequencing with split pool barcoding. *bioRxiv*, 2022.08.27.505512 (Aug. 2022).
180. Velmeshev, D. *et al.* Single-cell genomics identifies cell type-specific molecular changes in autism. *Science* **364**, 685–689 (May 2019).
181. Perez, R. K. *et al.* Single-cell RNA-seq reveals cell type-specific molecular and genetic associations to lupus. *Science* **376**, eabf1970 (Apr. 2022).
182. Eraslan, G. *et al.* Single-nucleus cross-tissue molecular reference maps to decipher disease gene function. *bioRxiv*, 2021.07.19.452954 (July 2021).
183. Salcher, S. *et al.* High-resolution single-cell atlas reveals diversity and plasticity of tissue-resident neutrophils in non-small cell lung cancer. *Cancer Cell* **40**, 1503–1520 (2022).
184. Mitchel, J. *et al.* Tensor decomposition reveals coordinated multicellular patterns of transcriptional variation that distinguish and stratify disease individuals. *bioRxiv*, 2022.02.16.480703 (Feb. 2022).
185. Dann, E., Henderson, N. C., Teichmann, S. A., Morgan, M. D. & Marioni, J. C. Differential abundance testing on single-cell data using k-nearest neighbor graphs. *Nature Biotechnology* **40**, 245–253 (Feb. 2022).
186. Abdollahi, A. *et al.* Inhibition of platelet-derived growth factor signaling attenuates pulmonary fibrosis. *The Journal of Experimental Medicine* **201**, 925–935 (2005).
187. Burkhardt, D. B. *et al.* Quantifying the effect of experimental perturbations at single-cell resolution. *Nature Biotechnology* **39**, 619–629 (May 2021).
188. Lotfollahi, M. *et al.* Learning interpretable cellular responses to complex perturbations in high-throughput screens. *bioRxiv*, 2021.04.14.439903 (May 2021).
189. Weinberger, E., Lin, C. & Lee, S.-I. Isolating salient variations of interest in single-cell data with contrastiveVI. *bioRxiv*, 2021.12.21.473757 (June 2022).
190. Schulte-Schrepping, J. *et al.* Severe COVID-19 is marked by a dysregulated myeloid cell compartment. *Cell* **182**, 1419–1440 (2020).
191. Martinez, F. O., Combes, T. W., Orsenigo, F. & Gordon, S. Monocyte activation in systemic Covid-19 infection: Assay and rationale. *EBioMedicine* **59**, 102964 (Sept. 2020).

192. Shojaei, M. *et al.* IFI27 transcription is an early predictor for COVID-19 outcomes, a multi-cohort observational study. *Frontiers in Immunology* **13**, 1060438 (2023).
193. De Cevins, C. *et al.* A monocyte/dendritic cell molecular signature of SARS-CoV-2-related multisystem inflammatory syndrome in children with severe myocarditis. *Med* **2**, 1072–1092 (2021).
194. Patel, A. A. *et al.* The fate and lifespan of human monocyte subsets in steady state and systemic inflammation. *Journal of Experimental Medicine* **214**, 1913–1923 (2017).
195. Subramanian, A. *et al.* A Next Generation Connectivity Map: L1000 Platform and the First 1,000,000 Profiles. *Cell* **171**, 1437–1452.e17 (Nov. 2017).
196. Jost, M. *et al.* Pharmaceutical-Grade Rigosertib Is a Microtubule-Destabilizing Agent. *Molecular Cell* **79**, 191–198.e3 (July 2020).
197. Bagratuni, T. *et al.* JQ1 inhibits tumour growth in combination with cisplatin and suppresses JAK/STAT signalling pathway in ovarian cancer. *European Journal of Cancer* **126**, 125–135 (Feb. 2020).
198. Shi, J. & Walker, M. G. Gene set enrichment analysis (GSEA) for interpreting gene expression profiles. *Current Bioinformatics* **2**, 133–137 (2007).
199. Horwitz, K., Costlow, M. & McGuire, W. L. MCF-7: a human breast cancer cell line with estrogen, androgen, progesterone, and glucocorticoid receptors. *Steroids* **26**, 785–795 (1975).
200. Grosveld, G. *et al.* The chronic myelocytic cell line K562 contains a breakpoint in bcr and produces a chimeric bcr/c-abl transcript. *Molecular and Cellular Biology* **6**, 607–616 (1986).
201. Hashash, J. G. *et al.* Altered Expression of the Epithelial Mucin MUC1 Accompanies Endoscopic Recurrence of Post-operative Crohn's disease. *Journal of Clinical Gastroenterology* **55**, 127 (2021).
202. Leal, R. F. *et al.* Identification of inflammatory mediators in patients with Crohn's disease unresponsive to anti-TNF α therapy. *Gut* **64**, 233–242 (2014).
203. Pavlidis, P. *et al.* Interleukin-22 regulates neutrophil recruitment in ulcerative colitis and is associated with resistance to ustekinumab therapy. *Nature Communications* **13**, 5820 (Oct. 2022).
204. Romano, E., Rosa, I., Fioretto, B. S. & Manetti, M. The contribution of endothelial cells to tissue fibrosis. *Current Opinion in Rheumatology* **36**, 52–60 (Jan. 2024).

205. Mukherjee, P. K. *et al.* Structuring Crohn’s disease single-cell RNA sequencing reveals fibroblast heterogeneity and intercellular interactions. *bioRxiv*, 2023–04 (2023).
206. Pilling, D., Vakil, V., Cox, N. & Gomer, R. H. TNF- α -stimulated fibroblasts secrete lumican to promote fibrocyte differentiation. *Proceedings of the National Academy of Sciences* **112**, 11929–11934 (2015).
207. Yang, K. *et al.* Transforming growth factor- β induced protein regulates pulmonary fibrosis via the G-protein signaling modulator 2/Snail axis. *Peptides* **155**, 170842 (2022).
208. Wu, F. & Chakravarti, S. Differential expression of inflammatory and fibrogenic genes and their regulation by NF- κ B inhibition in a mouse model of chronic colitis. *The Journal of Immunology* **179**, 6988–7000 (2007).
209. Rieder, F. *et al.* Inflammation-induced endothelial-to-mesenchymal transition: a novel mechanism of intestinal fibrosis. *The American Journal of Pathology* **179**, 2660–2673 (2011).
210. Siletti, K. *et al.* Transcriptomic diversity of cell types across the adult human brain. *bioRxiv*, 2022.10.12.511898 (Oct. 2022).
211. Kumar, D., Cinghu, S., Oldfield, A. J., Yang, P. & Jothi, R. Decoding the function of bivalent chromatin in development and cancer. *Genome Research* **31**, 2170–2184 (2021).
212. Takahashi, H., Iwata, T., Yamanaka, Y., Yamada, M. & Yagi, S. *Variational autoencoder with implicit optimal priors* in *Proceedings of the AAAI Conference on Artificial Intelligence* **33** (2019), 5066–5073.
213. Yao, Y., Rosasco, L. & Caponnetto, A. On Early Stopping in Gradient Descent Learning. *Constructive Approximation* **26**, 289–315 (Apr. 2007).
214. Traag, V. A., Waltman, L. & van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports* **9**, 5233 (Mar. 2019).
215. Missarova, A., Rosen, L. U., Dann, E., Satija, R. & Marioni, J. Sensitive cluster-free differential expression testing. *bioRxiv*, 2023–03 (2023).
216. Agrawal, A., Ali, A. & Boyd, S. Minimum-distortion embedding. *Foundations and Trends® in Machine Learning* **14**, 211–378 (2021).
217. Ward Jr, J. H. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association* **58**, 236–244 (1963).
218. Pilarczyk, M. *et al.* Connecting omics signatures and revealing biological mechanisms with iLINCS. *Nature Communications* **13**, 4678 (2022).

219. MacKay, D. J. *Information theory, inference and learning algorithms* (Cambridge university press, 2003).
220. Tirosh, I. *et al.* Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science* **352**, 189–196 (2016).
221. Zheng, L. *et al.* Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems* **36**, 46595–46623 (2024).
222. Corston-Oliver, S., Gamon, M. & Brockett, C. *A machine learning approach to the automatic evaluation of machine translation in Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics* (2001), 148–155.
223. Agarwal, M., Kalia, R., Bahel, V. & Thomas, A. *Autoeval: An NLP approach for automatic test evaluation system in 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)* (2021), 1–6.
224. Garg, S., Balakrishnan, S., Lipton, Z. C., Neyshabur, B. & Sedghi, H. Leveraging unlabeled data to predict out-of-distribution performance. *arXiv preprint arXiv:2201.04234* (2022).
225. Papineni, K., Roukos, S., Ward, T. & Zhu, W.-J. *Bleu: a method for automatic evaluation of machine translation in Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (2002), 311–318.
226. Lin, C.-Y. & Och, F. *Looking for a few good metrics: ROUGE and its evaluation in Ntcir workshop* (2004).
227. Wei, J. T.-Z. & Jia, R. The statistical advantage of automatic NLG metrics at the system level. *arXiv preprint arXiv:2105.12437* (2021).
228. Al Mohamad, D., van Zwet, E., Solari, A. & Goeman, J. Simultaneous confidence intervals for ranks using the partitioning principle. *Electronic Journal of Statistics* **15**, 2608–2646 (2021).
229. Chaganty, A. T., Mussman, S. & Liang, P. The price of debiasing automatic metrics in natural language evaluation. *arXiv preprint arXiv:1807.02202* (2018).
230. Angelopoulos, A. N., Duchi, J. C. & Zrnic, T. *PPI++: Efficient Prediction-Powered Inference* Nov. 2023.
231. Tsiatis, A. A. *Semiparametric Theory and Missing Data* (Springer, 2006).
232. Saad-Falcon, J., Khattab, O., Potts, C. & Zaharia, M. Ares: An automated evaluation framework for retrieval-augmented generation systems. *arXiv preprint arXiv:2311.09476* (2023).

233. Chatzi, I., Straitouri, E., Thejaswi, S. & Rodriguez, M. G. Prediction-Powered Ranking of Large Language Models. *arXiv preprint arXiv:2402.17826* (2024).
234. *Chatbot Arena: Benchmarking LLMs in the Wild* <https://chat.lmsys.org/>. Accessed: 2025-01-21.
235. Chiang, W.-L. *et al.* Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. *arXiv preprint arXiv:2403.04132* (2024).
236. Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems* **32** (2019).
237. Meier, J. *et al.* Language models enable zero-shot prediction of the effects of mutations on protein function. *bioRxiv*, 2021.07.09.450648 (Nov. 2021).
238. Laine, E., Karami, Y. & Carbone, A. GEMME: A Simple and Fast Global Epistatic Model Predicting Mutational Effects. *Molecular Biology and Evolution* **36**, 2604–2619 (Aug. 2019).
239. Riesselman, A. J., Ingraham, J. B. & Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nature Methods* **15**, 816–822 (Sept. 2018).
240. Notin, P. *et al.* ProteinGym: Large-Scale Benchmarks for Protein Design and Fitness Prediction. *bioRxiv*, 2023.12.07.570727 (Dec. 2023).
241. Olson, C. A., Wu, N. C. & Sun, R. A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain. *Current Biology* **24**, 2643–2651 (Nov. 2014).
242. Marquet, C. *et al.* Embeddings from protein language models predict conservation and variant effects. *Human Genetics* **141**, 1629–1647 (Oct. 2022).
243. Ji, Z. *et al.* Survey of hallucination in natural language generation. *ACM Computing Surveys* **55**, 1–38 (2023).
244. Li, T. *et al.* From Crowdsourced Data to High-Quality Benchmarks: Arena-Hard and BenchBuilder Pipeline. *arXiv preprint arXiv:2406.11939* (2024).
245. Zermelo, E. Die berechnung der turnier-ergebnisse als ein maximumproblem der wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift* **29**, 436–460 (1929).
246. Bradley, R. & Terry, M. E. Rank analysis of incomplete block designs: The method of paired comparisons. *Biometrika* **39**, 324–345 (1952).
247. Ford Jr, L. R. Solution of a ranking problem from binary comparisons. *The American Mathematical Monthly* **64**, 28–33 (1957).

248. Hunter, D. R. MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics* **32**, 384–406 (2004).
249. Chen, K. H., Boettiger, A. N., Moffitt, J. R., Wang, S. & Zhuang, X. RNA imaging. Spatially resolved, highly multiplexed RNA profiling in single cells. en. *Science* **348**, aaa6090 (Apr. 2015).
250. Moffitt, J. R. *et al.* Molecular, spatial, and functional single-cell profiling of the hypothalamic preoptic region. en. *Science* **362** (Nov. 2018).
251. Ståhl, P. L. *et al.* Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. en. *Science* **353**, 78–82 (July 2016).
252. Rodriques, S. G. *et al.* Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. en. *Science* **363**, 1463–1467 (Mar. 2019).
253. Zhang, M. *et al.* Spatially Resolved Cell Atlas of the Mouse Primary Motor Cortex by MERFISH. *Nature* **598**, 137–143 (Oct. 2021).
254. Allen, W. E., Blosser, T. R., Sullivan, Z. A., Dulac, C. & Zhuang, X. Molecular and Spatial Signatures of Mouse Brain Aging at Single-Cell Resolution. *Cell* **186**, 194–208.e18 (Jan. 2023).
255. Lu, Y. *et al.* Spatial Transcriptome Profiling by MERFISH Reveals Fetal Liver Hematopoietic Stem Cell Niche Architecture. *Cell Discovery* **7**, 47 (June 2021).
256. Jones, D. C. *et al.* *Cell Simulation as Cell Segmentation* Apr. 2024.
257. Palla, G. *et al.* Squidpy: A Scalable Framework for Spatial Omics Analysis. *Nature Methods* **19**, 171–178 (Feb. 2022).
258. Dries, R. *et al.* Giotto: A Toolbox for Integrative Analysis and Visualization of Spatial Expression Data. *Genome Biology* **22**, 78 (Mar. 2021).
259. Palla, G., Fischer, D. S., Regev, A. & Theis, F. J. Spatial Components of Molecular Tissue Biology. *Nature Biotechnology* **40**, 308–318 (Mar. 2022).
260. Park, J. *et al.* Cell Segmentation-Free Inference of Cell Types from in Situ Transcriptomics Data. *Nature Communications* **12**, 3545 (June 2021).
261. Stringer, C. & Pachitariu, M. Cellpose3: One-Click Image Restoration for Improved Cellular Segmentation. *Nature Methods* **22**, 592–599 (Mar. 2025).
262. Kirillov, A. *et al.* Segment Anything. *arXiv [cs.CV]* (Apr. 2023).
263. Archit, A. *et al.* Segment Anything for Microscopy. *Nature Methods* **22**, 579–591 (Mar. 2025).

264. Ergen, C. & Yosef, N. *ResolVI - Addressing Noise and Bias in Spatial Transcriptomics* Jan. 2025.
265. Van Buren, S. *et al.* Compression of Quantification Uncertainty for scRNA-seq Counts. *Bioinformatics* **37**, 1699–1707 (July 2021).
266. Cole, M. B. *et al.* Performance Assessment and Selection of Normalization Procedures for Single-Cell RNA-Seq. *Cell Systems* **8**, 315–328.e8 (Apr. 2019).
267. Ahlmann-Eltze, C. & Huber, W. Comparison of Transformations for Single-Cell RNA-seq Data. *Nature Methods* **20**, 665–672 (May 2023).
268. Zrnic, T. & Candès, E. J. *Active Statistical Inference* May 2024. arXiv: 2403.03208 [stat].
269. Miao, J. *et al.* *Valid Inference for Machine Learning-Assisted GWAS* Jan. 2024.
270. Boyeau, P., Angelopoulos, A. N., Yosef, N., Malik, J. & Jordan, M. I. *AutoEval Done Right: Using Synthetic Data for Model Evaluation* Mar. 2024.
271. Vizgen. *Vizgen MERFISH FFPE Human Immuno-oncology Data Set* Dataset (May 2022).
272. CVAT.ai Team. *Computer Vision Annotation Tool* version 2.31.0. Accessed: 2025-03-22. 2025.
273. Scott, M. C. *et al.* Deep Profiling Deconstructs Features Associated with Memory CD8⁺ T Cell Tissue Residence. *Immunity* **58**, 162–181.e10 (Jan. 2025).
274. Lehtonen, H. *et al.* Early Precursor-Derived Pituitary Gland Tissue-Resident Macrophages Play a Pivotal Role in Modulating Hormonal Balance. *Cell Reports* **44**, 115227 (Feb. 2025).
275. Khan, S. R. *et al.* Reduced Circulating Sphingolipids and CERS2 Activity Are Linked to T2D Risk and Impaired Insulin Secretion. *Science Advances* **11**, eadr1725 (Jan. 2025).
276. Sheban, F. *et al.* ZEB2 is a master switch controlling the tumor-associated macrophage program. *Cancer Cell* (2025).
277. Kirschenbaum, D. *et al.* Time-resolved single-cell transcriptomics defines immune trajectories in glioblastoma. *Cell* **187**, 149–165.e23 (2024).
278. Wells, S. B. *et al.* Multimodal profiling reveals tissue-directed signatures of human immune cells altered with age. *bioRxiv*, 2024–01 (2024).