UCLA
UCLA Electronic Theses and Dissertations

Title

From Subjective to Principled Single-Cell Data Analysis: Evaluating Annotation Behavior, Optimizing Integration, and Benchmarking Visualization Pipelines

Permalink

https://escholarship.org/uc/item/7vt623b9

ISBN

9798247933618

Author

Zhai, Zhiqian

Publication Date

2026-05-27

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA

Los Angeles

From Subjective to Principled Single-Cell Data Analysis: Evaluating Annotation Behavior, Optimizing Integration, and Benchmarking Visualization Pipelines

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Statistics

by

Zhiqian Zhai

2026

© Copyright by
Zhiqian Zhai
2026

ABSTRACT OF THE DISSERTATION

From Subjective to Principled Single-Cell Data Analysis:
Evaluating Annotation Behavior, Optimizing Integration, and Benchmarking Visualization Pipelines

by

Zhiqian Zhai
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2026
Professor Jingyi Li, Chair

Over recent years, single-cell RNA sequencing (scRNA-seq) and spatial transcriptomics have transformed transcriptomic research by enabling high-resolution gene expression profiling and making it possible to generate atlas-scale datasets efficiently and cost-effectively. Yet rigorous scRNA-seq analysis remains challenging because key tasks, including cell-type annotation, data integration, and visualization, are often affected by substantial methodological and practical limitations. Cell-type annotation relies on a flexible multi-step pipeline whose functions and parameter choices are often shaped by analysts' judgment and expertise, introducing subjectivity that can reduce reproducibility. Integration across batches is essential for large-scale and multi-condition studies, but it requires balancing the removal of between-batch variation with the preservation of true cell identity. Existing integration workflows still lack principled annotation-free strategies for feature selection and hyperparameter tuning. Visualization of atlas-scale data is further complicated by computational demands and by the strong influence of preprocessing steps such as normalization and integration, which remain insufficiently evaluated in existing benchmark studies. This dissertation addresses these challenges through three complementary projects on subjectivity in annotation, principled integration optimization, and systematic benchmarking of large-scale visualization pipelines.

My first project examines how graduate students enrolled in an advanced bioinformatics course at UCLA performed cell-type annotation. We also collected information on their pipeline-tuning practices, annotation results, and educational and research backgrounds. Our study revealed that participants performed well in identifying major cell types but often struggled with closely related cell subtypes. Participants who achieved higher cell-type annotation accuracy tended to incorporate data-quality control into analysis pipelines or have prior publications with single-cell analysis. Subjective choices of parameter values influence clustering results, and tuning beyond default settings typically improves clustering accuracy. However, because cell-type label assignment is driven largely by prespecified marker genes and subjective judgments of their expression, parameter tuning has limited impact on improving annotation accuracy. We also identified a confirmation bias: prior expectations about cell types influence subjective annotation decisions. For comparison, we evaluated an AI agent, Biomni, for automated cell-type annotation and found its performance worse than 70–90% of participants. Our findings underscore the importance of transparent reporting of analysis pipelines, including parameter choice justifications and prior expectations about cell types.

My second project presents IntegrateRigor, a data-driven, method-agnostic framework that performs batch-stable gene selection and optimizes integration hyperparameters across state-of-the-art methods, without relying on cell identity annotations. IntegrateRigor selects genes whose expression patterns are stable across batches using a gene-wise likelihood-based batch stability score, excluding batch-sensitive genes that can bias batch alignment during integration. IntegrateRigor then identifies optimal integration results across methods and hyperparameters by defining a dataset-level integration score that balances between-batch variation removal against cell identity preservation. In a colorectal cancer single-cell and spatial transcriptomics dataset, IntegrateRigor revealed previously uncharacterized cancer-immune interface states in the tumor microenvironment that were masked by both under-integration under default settings and over-integration in previous literature. Across diverse single-cell and spatial transcriptomics datasets, IntegrateRigor consistently improves integration performance by balancing over-integration and under-integration. By transforming

integration from a heuristic step into a statistically principled, dataset-adaptive procedure, IntegrateRigor improves the reproducibility and discovery power of large-scale single-cell analyses.

My third project proposes a comprehensive benchmark study evaluating 180 method pipelines that combine state-of-the-art normalization, integration, and visualization methods. We assessed performance on five semi-synthetic atlas-scale scRNA-seq datasets that mimic real data and include ground truths. Each pipeline was evaluated using ten quantitative metrics that assess visualization accuracy and scalability. This benchmark provided systematic, evidence-based guidance for selecting appropriate methods for visualizing atlas-scale single-cell data.

The dissertation of Zhiqian Zhai is approved.

Ker-Chau Li

Mark S. Handcock

Sudipto Banerjee

Jingyi Li, Committee Chair

University of California, Los Angeles

2026

To my parents.

TABLE OF CONTENTS

1 Introduction **1**

1.1 How does human subjectivity affect the reliability of single-cell RNA-seq cell-type annotation? 2

1.2 IntegrateRigor: annotation-free integration optimization for cell identity recovery reveals cancer–immune interface niches 3

1.3 Benchmarking of the single-cell cell-type visualization pipeline on atlas-scale scRNA-seq datasets 4

2 How does human subjectivity affect the reliability of single-cell RNA-seq cell-type annotation? **6**

2.1 Introduction 6

2.2 Materials and methods 9

2.2.1 Study design 9

2.2.2 Participant recruitment 11

2.2.3 AI agent for the same cell-type annotation tasks 11

2.2.4 Cell-type annotation evaluation metrics 12

2.3 Results 18

2.3.1 Participants' annotations exhibit variability and difficulty in distinguishing cell subtypes 18

2.3.2 Single-cell research experience leads to more accurate cell-type annotation 19

2.3.3 Subjectivity in label assignment reduces the benefits of parameter tuning for improved clustering 21

2.3.4 Confirmation bias deteriorates annotation accuracy 22

2.3.5 Biomni underperforms 70–90% participants in cell-type annotation accuracy . . . 24
2.4 Discussion . . . 26
2.5 Code and data availability . . . 28
2.5.1 Analysis and code . . . 28
2.5.2 Datasets . . . 28
2.6 Acknowledgements . . . 29
2.7 Figures . . . 30
2.8 Supplementary materials . . . 36
S2.8.1 Supplementary figures . . . 36
S2.8.2 Supplementary tables . . . 37
S2.8.3 Instructions of the cell-type annotation tasks for participants . . . 38
S2.8.4 Prompts of the cell-type annotation tasks for Biomni . . . 41
S2.8.5 Issues we faced when using Biomni . . . 44

3 IntegrateRigor: annotation-free integration optimization for cell identity recovery reveals cancer–immune interface niches . . . 46
3.1 Introduction . . . 46
3.2 Results . . . 50
3.2.1 Overview of IntegrateRigor: an annotation-free framework for batch-stable gene selection and integration optimization for cell identity recovery . . . 50
3.2.2 Batch stability scores identify batch-stable genes and improve cell identity recovery by excluding genes confounded by sample and technical variation . . . 53

3.2.3 Integration scores guide hyperparameter optimization to balance under-integration and over-integration for cell identity recovery 55
3.2.4 IntegrateRigor consistently improves cell identity recovery across diverse datasets and methods 56
3.2.5 IntegrateRigor uncovers cancer–immune interface niches in colorectal cancer by integrating single-cell and spatial transcriptomics data . . 58
3.3 Methods 61
3.3.1 Batch-stable gene (BSG) selection for data integration 61
3.3.2 Integration score for a given dataset and integration method 64
3.3.3 Generalization to multiple batches 68
3.3.4 Validation of the statistical models underlying IntegrateRigor 69
3.3.5 Stability and convergence 70
3.3.6 Annotation-based evaluation metrics 71
3.4 Discussion 73
3.5 Code and data availability 76
3.5.1 Data availability 76
3.5.2 Code availability 77
3.6 Acknowledgements 77
3.7 Figures 78
3.8 Supplementary materials 83
S3.8.1 Dataset Summary 83
S3.8.2 Empirical Validation of IntegrateRigor 83
S3.8.3 Genes contributing to batch effect in the unintegrated PCA overlaps with the batch-unstable genes 84
S3.8.4 Batch stability score correlates with annotation-dependent GTE scores 87

S3.8.5 IntegrateRigor identifies ribosomal genes more frequently as batch-unstable genes . . . 88
S3.8.6 Figures with complete legends for Main Figure . . . 90
S3.8.7 Harmony parameters selected by existing metrics . . . 91
S3.8.8 IntegrateRigor sensitivity and convergence to the IntegrateRigor hyperparameters . . . 92
S3.9 Complete IntegrateRigor results . . . 94
S3.9.1 Integration score across different methods . . . 101
S3.9.2 Cancer-immune niches in the colorectal cancer dataset . . . 101

4 Benchmarking of cell-type visualization pipelines with semi-synthetic atlas-scale scRNA-seq datasets . . . 103

4.1 Introduction . . . 103
4.2 Reference datasets for benchmarking . . . 105
4.2.1 Mouse cortex dataset . . . 106
4.2.2 Human prefrontal cortex dataset . . . 106
4.2.3 Human liver dataset . . . 107
4.2.4 Macaque retina fovea dataset . . . 107
4.2.5 IFALD (intestinal failure–associated liver disease) liver dataset . . . 107
4.3 Methods . . . 108
4.3.1 Selection of methods to be benchmarked . . . 108
4.3.2 Overview of methods benchmarked . . . 109
4.3.3 Benchmarking procedure . . . 114
4.3.4 Semi-synthetic data generation . . . 116
4.3.5 Benchmarking metrics . . . 118
4.4 Benchmark results . . . 127

4.5 Discussion . . . 129
4.6 Data availability . . . 131
4.7 Acknowledgements . . . 131
4.8 Figures . . . 132
4.9 Supplementary figures . . . 135

5 Summary and future directions . . . 137

5.1 How does human subjectivity affect the reliability of single-cell RNA-seq cell-type annotation? . . . 138
5.2 IntegrateRigor: annotation-free integration optimization for cell identity recovery reveals cancer–immune interface niches . . . 139
5.3 Benchmarking of cell-type visualization pipelines with semi-synthetic atlas-scale scRNA-seq datasets . . . 141

LIST OF FIGURES

1.1 **Overview of a typical single-cell RNA-seq analysis workflow.** Single-cell RNA-seq analysis in this dissertation begins with a gene-by-cell count matrix, followed by normalization to reduce technical variation such as differences in sequencing depth. A subset of informative genes is then selected for downstream analysis, typically reducing the feature space from tens of thousands of genes to a smaller set of highly variable genes. Dimension reduction or integration methods, such as PCA or batch-correction approaches, are applied to generate low-dimensional cell embeddings. These embeddings are then used for visualization, clustering, and cell-type annotation, producing interpretable cell groups and cell identity labels. The figure was adopted from [12] 5

2.1 **The cell-type annotation pipeline of Seurat V5 used by participants.** The blue oval boxes show the steps in the Seurat V5 cell-type annotation pipeline. HVG refers to the highly variable gene identification procedure. DE genes refer to the differentially expressed genes. The blue square box below shows the functions and parameters in Seurat V5 that the participants can choose or tune. The Seurat functions participants could choose to use are indicated in blue; tunable parameters for each function are shown in black; default values for each parameter are written in parentheses. 30

2.2 **Datasets and study design. a** The bone marrow complete dataset with complete 21 cell types from [73]. **b** Two variants of the bone marrow partial datasets, each with specific cell types removed (indicated by colored dashed contours in **a** and colored dashed squares in the legend). **c** Two variants of the pancreas dataset. The pancreas complete dataset contains all cell types, whereas the partial dataset excludes both pancreatic polypeptide (PP) and alpha cells, indicated by red dashed contour and squares. For the pancreas datasets, participants were specifically instructed to annotate only PP and alpha cells. **d** For each participant, each of their three assigned datasets contained 90% of cells randomly sampled from the corresponding dataset, with each participant randomly assigned one of the two bone marrow partial datasets and one of the two pancreas datasets. 31

2.3 **Reference cell-type hierarchy used to compute annotation accuracy for the bone marrow datasets.** Participant-supplied labels not in the reference were incorporated by semantic mapping to enable accuracy scoring. Grey dashed boxes mark these participant-supplied labels absent from the reference but added for accuracy scoring; white boxes mark reference labels withheld from participants but included in the scoring hierarchy; blue boxes are the 21 reference labels provided to participants for annotation. The annotation accuracy for a cell is defined as one minus the normalized shortest path length between a participant's annotated label and the reference label in the hierarchy (range: -1 to 1), where the normalization factor is the hierarchy's maximum depth of 4. For context, the shortest path length between a CD4 T cell and a memory B cell (different cell types) is 4, whereas the shortest path length between a CD4 T cell and a CD8 T cell (same cell type; different cell subtypes) is 2. If a participant's annotation for a cell is NA, then the cell's annotation accuracy is defined as -1. Participant-level annotation accuracy is defined as the mean of the cell-level annotation accuracies for all cells assigned to that participant. This metric penalizes misannotations to closely related types less than those to more distant types. 32

2.4 **Participant- and cell-level accuracy and variability on the bone marrow complete and partial datasets.** **a** Participant-level clustering accuracy and annotation accuracy for both bone marrow datasets, shown as boxplots. P-values are computed using the one-sided Mood's median test, with the alternative hypothesis that the BM complete dataset yields higher accuracies than the BM partial datasets, as the partial datasets do not contain all 21 cell types in the provided reference list. "BM complete'' refers to the bone marrow complete dataset; "BM partial'' includes both bone marrow partial 1 and partial 2 datasets. The blue dashed lines indicate baseline clustering and annotation accuracies obtained by randomly assigning one of the 21 cell types to each cell. **b** Summary statistics (mean, quantiles, and standard deviation (std)) for participant-level clustering and annotation accuracies. **c** Cell-level annotation variability, shown as boxplots; variability is defined as the number of unique participant-assigned labels per cell. **d** Cell-level annotation accuracy based on majority-vote consensus across participants, computed using the shortest path length between the majority-vote label and the reference label in the predefined hierarchy. 33

2.5 **Annotation accuracy vs. questionnaire responses.** **a** Boxplots for highlighted participant educational and research background showing significant association ($\alpha = 0.1$) with annotation accuracy, especially in the BM (bone marrow) complete dataset. **b** Summary table reporting the statistical tests used and corresponding p-values for annotation accuracy across all questionnaire response variables. Significant p-values are highlighted with underlines. Mood stands for the non-parametric Mood's median test. JT stands for the Jonckheere-Terpstra test, which assesses monotonic trends across ranked levels. 34

2.6 **Parameter choices and reference marker gene selection influence annotation accuracy on the bone marrow complete dataset.** **a** Boxplots comparing clustering accuracy based on whether the default parameter value was used. Only parameters for which at least two participants used each setting (default or non-default) are included. P-values are calculated using the one-sided Mood's median test, and significant values (p-value $\leqslant 0.1$) are highlighted in blue. QC stands for data-quality control. **b** Boxplots comparing annotation accuracy based on whether the default parameter value was used. Only parameters for which at least two participants used each setting (default or non-default) are included. P-values are calculated using the one-sided Mood's median test, and significant values (p-value $\leqslant 0.1$) are highlighted in blue. **c** Relationship between the average number of reference marker genes used per cell type and per-participant annotation accuracy. Each dot represents a participant; the average number of reference marker genes per cell type is the total number of reference markers a participant selected from literature or databases, averaged across cell types. A negative association indicates that using more reference markers can make the cell-type assignment step more challenging and increase subjectivity. 35

2.7 **Parameter choices and reference marker gene selection influence annotation accuracy on the bone marrow partial dataset. a** Boxplots comparing clustering accuracy based on whether the default parameter value was used. Only parameters for which at least two participants used each setting (default or non-default) are included. P-values are calculated using the one-sided Mood's median test, and significant values (p-value $\leqslant 0.1$) are highlighted in blue. QC stands for data-quality control. **b** Boxplots comparing annotation accuracy based on whether the default parameter value was used. Only parameters for which at least two participants used each setting (default or non-default) are included. P-values are calculated using the one-sided Mood's median test. **c** Relationship between the average number of reference marker genes used per cell type and per-participant annotation accuracy. Each dot represents a participant; the average number of reference marker genes per cell type is the total number of reference markers a participant selected from literature or databases, averaged across cell types. A negative association indicates that using more reference markers can make the cell-type assignment step more challenging and increase subjectivity. . 36

3.1 **Overview of IntegrateRigor framework. a.** Schematic of the IntegrateRigor workflow. In Stage 1, batch-stable genes (BSGs) are selected by comparing each query batch to a reference batch and computing a per-gene batch stability score (BSS), which quantifies the extent to which gene-level distributions are preserved across batches. Genes ranked above the elbow point are retained as BSGs. In Stage 2, count matrices are restricted to BSGs, and candidate hyperparameter settings for a chosen integration method are evaluated. The integrated embedding with the highest integration score is selected as optimal. **b.** Illustration of the BSS and batch alignment score (BAS). For each gene or each integrated embedding dimension, between-batch variation is modeled as a distributional shift in at least one mixture component corresponding to a shared cell identity across batches. Differences in mixture component proportions are not treated as between-batch variation. BSS is estimated on gene counts using a negative binomial mixture model, whereas per-dimension BASs are estimated on integrated embeddings using a Gaussian mixture model. **c.** Illustration of the cell identity score (CIS). For each integrated embedding dimension, preservation of cell identity is quantified by the presence of multiple Gaussian mixture components, indicating distinguishable cell identities. The per-dimension CIS is estimated using a Gaussian mixture model on the integrated embedding. The overall integration score is defined as the sum of BAS and CIS. 78

3.2 **IntegrateRigor identifies batch-unstable genes and uses integration scores to improve gene selection and hyperparameter selection while avoiding overintegration. a.** Heatmaps of the top and bottom 50 genes ranked by BSS, showing average expression across batches within each cell identity. Top-ranked genes are batch-stable, whereas bottom-ranked genes exhibit substantial batch-dependent variability. **b.** Harmony integration before and after selecting highly variable genes from the BSGs set. BSG selection improves batch mixing (higher CiLISI) while better preserving cell identity structure (higher ARI), reducing underintegration. **c.** Summary of the three IFALD batches, which differ in donors' age, conditions, and sample preservation. **d.** Comparison of mitochondrial and ribosomal gene proportions between batch-stable and batch-unstable genes, showing strong enrichment of ribosomal genes among the batch-unstable genes. Here, the batch-unstable genes are the genes with low BSS scores and are excluded from the integration. **e.** Gene Ontology enrichment analysis of the batch-unstable genes, highlighting biological processes related to coagulation, inflammatory response, lipid metabolism, and cytoplasmic translation. **f.** Visualization of Harmony dimensions with the highest and lowest batch alignment or CISs. Dimensions with high BASs show better batch mixing, whereas dimensions with high CISs better preserve biological structure. **g.** Integration scores guide hyperparameter selection for Harmony, improving both batch correction and cell identity preservation relative to the default setting. **h.** Compared with selection based on iLISI, IntegrateRigor selects a solution that avoids overintegration and better preserves cell identity, as reflected by a lower cLISI. Cell identity labels and color annotations for the UMAPs are provided in Supplementary Figures 3.11-Figure 3.13. 80

3.3 **IntegrateRigor improves integration performance across diverse datasets and methods. a.** Representative UMAPs for four datasets comparing default integration with IntegrateRigor-optimized integration. Cells are colored by batch (first column in each dataset's subfigure) or cell identity (second column in each dataset's subfigure). Across datasets and methods, IntegrateRigor improves batch mixing (CiLISI) while preserving or improving cell identity separation (ARI). **b.** Relative gains in CiLISI and ARI obtained by IntegrateRigor over default integration across six benchmark datasets and five integration methods. Results are grouped by the major source of between-batch variation. **c.** Method-level summary of average ARI and scaled CiLISI before and after IntegrateRigor optimization. Filled circles denote IntegrateRigor-selected solutions and open circles denote default settings. 82

3.4 **IntegrateRigor reveals unique cancer–immune interface niche in colorectal cancer data.** **a**, The original Crescendo integration and annotations. The top panels show the original Crescendo embedding and spatial coordinates before integration, and the bottom panel shows the Harmony-integrated embedding colored by technology. **b**, UMAPs from Harmony integration with IntegrateRigor-selected genes, colored by batch or by Crescendo cell identity annotations. Regions where originally annotated epithelial and immune cells were intermixed are highlighted with dashed circles and were further clustered and annotated as distinct cancer–immune niches based on the IntegrateRigor embedding. **c**, Refined cell identity annotations for the spatial transcriptomics samples PFA_A11 and PFA_A6. Immune cells located near epithelial cancer cells were further annotated as cancer–immune niche populations, including cancer–B niche, cancer–mast niche, cancer–myeloid niche, cancer–plasma niche, and cancer–T niche. **d**, Spatial distribution of the refined cell identities in PFA_A11. The main spatial map shows epithelial cancer cells, fibroblast and endothelial cells, immune cells, and cancer–immune niches. The inset highlights a zoomed-in region where cancer–immune niche cells are spatially intermingled with epithelial cancer cells. **e**, Dot plot of canonical marker genes across refined cell identity groups. Dot color represents average expression, and dot size represents the log fold change of each cancer–immune niche population relative to epithelial cancer cells. Cancer–immune niche groups retained expression of their corresponding immune markers while showing epithelial marker expression at levels comparable to epithelial cancer cells. **f**, Gene Ontology enrichment analysis of differentially expressed genes comparing the cancer-myeloid niche with myeloid cells after excluding epithelial marker genes. Enriched pathways include cytokine-mediated signaling, regulation of cytokine production, regulation of cell–cell adhesion, and T cell activation, supporting the biological distinction of the cancer-myeloid niche from conventional myeloid cells. 82

3.5 **Goodness-of-fit assessment using KS statistics on the IFALD dataset.** Left panel: Distribution of KS statistics for 2,000 highly variable genes fitted by the negative binomial mixture model (NBMM), showing generally small distributional discrepancies between the observed and fitted gene expression distributions. Right panel: Distribution of KS statistics across embedding dimensions for Gaussian mixture model (GMM) fits to integrated embeddings from different integration methods. Most methods show small KS statistics, whereas LIGER exhibits substantially larger KS values, indicating poorer compatibility with the GMM assumption. 84

3.6 **Observed and fitted gene-expression count distributions for randomly selected genes in the IFALD datasets.** For each selected gene, the histogram compares the empirical nonzero count distribution (Observed) with the model-implied fitted distribution (Fitted), excluding the zero-count bin to better visualize the fit among expressed cells. Each panel reports the gene Ensembl ID and the Kolmogorov–Smirnov (KS) statistic between the observed and fitted nonzero distributions. Smaller KS values indicate closer agreement between the fitted model and the observed distribution. 85

3.7 **Dimension-wise Gaussian mixture model fits for integrated embeddings on the IFALD datasets.** For each integration method, the distribution of embedding values is shown for the first five dimensions. Blue histograms represent the empirical distribution of integrated embedding values, and the overlaid colored density curves represent the fitted Gaussian mixture model with K = 5 components. Each panel reports the corresponding integration method, embedding dimension, and Kolmogorov–Smirnov (KS) statistic between the observed and fitted distributions. Smaller KS values indicate better agreement between the fitted mixture model and the empirical embedding distribution. 86

3.8 **PCA biplot for batch-stability assessment of top contributing genes in the unintegrated PCA space.** The unintegrated PCA embedding is shown for cells from three batches, with cells colored by batch. Arrows represent the top 100 genes contributing to the PCA structure, and arrow colors indicate whether each gene is classified by IntegrateRigor as batch-stable or batch-unstable. Batch-unstable genes tend to align with directions separating batches, whereas batch-stable genes are less associated with batch-driven variation, supporting the use of IntegrateRigor for identifying genes that are less affected by batch effects before integration. 87

3.9 **Correlation between gene-wise technical effects and IntegrateRigor batch-instability scores across datasets.** Each panel shows the relationship between the overall gene-level technical effect estimated by GTEs and the negative batch stability score computed by IntegrateRigor on the log scale. Each point represents one gene, and both axes are shown on the log scale. The red line indicates the fitted linear trend. Across all six datasets, genes with larger estimated technical effects tend to have higher negative batch stability scores, with significant positive Pearson correlations. This agreement suggests that IntegrateRigor effectively identifies genes whose expression variation is strongly associated with batch- or technology-driven effects. 88

3.10 **Cross-dataset overlap of ribosomal genes classified as batch-stable or batch-unstable by IntegrateRigor.** Upset plots showing the intersection patterns of ribosomal genes identified as batch-stable genes (BSGs; top or batch-unstable genes (BUSGs; bottom) across datasets. The horizontal bars indicate the total number of ribosomal genes in each dataset-specific BSG or BUSG set, while the vertical bars indicate the size of each cross-dataset intersection. Connected dots below each vertical bar denote the datasets included in the corresponding intersection. The results show that ribosomal genes classified as BUSGs are repeatedly shared across datasets, supporting their association with batch-driven technical variation. 89

3.11 **Main Figure 2b with complete batch and cell identity labels** 90

3.12 **Main Figure 2 f-g with complete batch and cell identity labels** 90

3.13 **Main Figure 2h with complete batch and cell identity labels** 91

3.14 **Relative metric trends across Harmony integration parameters.** Line plot showing how integration performance metrics change across different values of the Harmony parameter θ. In Harmony, it controls the strength of the diversity clustering penalty, with larger values encouraging stronger batch mixing during integration. All metrics are min–max normalized to the range $[0, 1]$ to facilitate comparison of relative trends across metrics. The IntegrateRigor integration score peaks at moderate θ values and decreases as θ becomes larger, suggesting that overly strong batch correction may reduce biological structure preservation. In contrast, several conventional batch-mixing metrics, including LISI, PCR, ASW, and kBET, continue to improve with larger θ, highlighting their tendency to favor stronger integration even when it may lead to over-integration. 91

3.15 **Stability of BSG selection and batch stability scores across different numbers of mixture components.** Line plots showing the robustness of IntegrateRigor outputs across different choices of K, where K = 5 is the default value. The top panel shows the Jaccard overlap between the BSGs selected at each K and those selected at K = 5. The bottom panel shows the correlation between gene-wise batch stability scores (BSSs) computed at each K and those computed at K = 5. 92

3.16 **Overlap of selected batch-stable genes across different numbers of EM algorithms' iterations per gene.** UpSet plot showing the overlap of batch-stable genes (BSGs) selected by IntegrateRigor under different maximum numbers of EM algorithms' iterations per gene. Horizontal bars indicate the number of BSGs selected under each iteration setting, vertical bars indicate intersection sizes, and connected dots denote the iteration settings included in each intersection. 92

3.17 **Stability of BSG selection and batch stability scores across different numbers of EM algorithms' iterations per gene.** Line plots showing the convergence of IntegrateRigor outputs across different maximum numbers of EM algorithms' iterations per gene, where iterations=10 is the default value. The left panel shows the Jaccard similarity between BSGs selected at each iteration setting and those selected with the default setting. The right panel shows the Pearson correlation between gene-wise batch stability scores (BSSs) computed at each iteration setting and those computed with the default setting. 93

3.18 **Stability of integration score across different numbers of Gaussian mixture components.** Line plots showing the correlation of the integration scores across different choices of K among the same set of hyperparameter values in Harmony, where K = 5 is the default value. 93

3.19 **Metrics evaluation for IntegrateRigor vs. default integration over ARI, CiLISI, and cLISI.** 94

3.20 **Comparison of default and IntegrateRigor-selected integration results on the IFN-β stimulated dataset.** UMAP visualizations of the dataset after integration using Harmony, scVI, Seurat-RPCA, FastMNN, and LIGER under either default feature/parameter settings or IntegrateRigor-selected settings. For each method and setting, cells/spots are shown twice: colored by batch on the left and by annotated cell type on the right. 95

3.21 **Comparison of default and IntegrateRigor-selected integration results on the IFALD dataset.** UMAP visualizations of the dataset after integration using Harmony, scVI, Seurat-RPCA, FastMNN, and LIGER under either default feature/parameter settings or IntegrateRigor-selected settings. For each method and setting, cells/spots are shown twice: colored by batch on the left and by annotated cell type on the right. 96

3.22 **Comparison of default and IntegrateRigor-selected integration results on the chicken heart dataset.** UMAP visualizations of the dataset after integration using Harmony, scVI, Seurat-RPCA, FastMNN, and LIGER under either default feature/parameter settings or IntegrateRigor-selected settings. For each method and setting, cells/spots are shown twice: colored by batch on the left and by annotated cell type on the right. 97

3.23 **Comparison of default and IntegrateRigor-selected integration results on the immune dataset.** UMAP visualizations of the dataset after integration using Harmony, scVI, Seurat-RPCA, FastMNN, and LIGER under either default feature/parameter settings or IntegrateRigor-selected settings. For each method and setting, cells/spots are shown twice: colored by batch on the left and by annotated cell type on the right. 98

3.24 **Comparison of default and IntegrateRigor-selected integration results on the human-mouse pancreas dataset.** UMAP visualizations of the dataset after integration using Harmony, scVI, Seurat-RPCA, FastMNN, and LIGER under either default feature/parameter settings or IntegrateRigor-selected settings. For each method and setting, cells/spots are shown twice: colored by batch on the left and by annotated cell type on the right. 99

3.25 **Comparison of default and IntegrateRigor-selected integration results on the mouse brain spatial dataset.** UMAP visualizations of the dataset after integration using Harmony, scVI, Seurat-RPCA, FastMNN, and LIGER under either default feature/parameter settings or IntegrateRigor-selected settings. For each method and setting, cells/spots are shown twice: colored by batch on the left and by annotated cell type on the right. 100

3.26 **Integration score values across different methods and datasets** 101

3.27 **Comparison of default Harmony integration and Harmony integration with IntegrateRigor for colorectal cancer dataset.** UMAP embeddings are shown for each integration strategy, with columns corresponding to default Harmony and Harmony with IntegrateRigor, and rows colored by batch or cell identity. For default Harmony, cells from different batches remain partially separated, indicating underintegration. 101

3.28 **GO enrichment analysis of genes distinguishing epithelial-niche cell states from canonical immune cell types.** Dot plots showing enriched Gene Ontology biological process terms for genes differentially enriched in epithelial-niche-associated B, T, plasma, and mast cell states compared with their corresponding canonical immune cell types. Each panel reports the top enriched GO terms for one comparison. The x-axis shows the gene ratio, dot size indicates the number of genes contributing to each GO term, and color represents the adjusted p-value. Across comparisons, epithelial-niche-associated cell states show enrichment for immune-related processes, including antigen processing and presentation, leukocyte-mediated immunity, lymphocyte-mediated immunity, and regulation of hematopoiesis, suggesting that these niche-associated populations retain immune functional programs while exhibiting distinct tissue-context-associated transcriptional features. 102

4.1 Summary of notations used in metric definitions 119

4.2 **Benchmarking framework for evaluating single-cell visualization pipelines.** The benchmarking workflow evaluates end-to-end visualization pipelines using semi-synthetic atlas-scale scRNA-seq count matrices with known ground truths. Each pipeline consists of a normalization method, an optional integration method that produces a 50-dimensional embedding, and a 2D visualization method. In total, 180 method combinations were evaluated across five semi-synthetic datasets, ranging from 0.5 to 1 million cells. Pipeline performance was assessed using ten quantitative metrics 132

4.3 **Workflow of semi-synthetic data generation.** This diagram illustrates how we generate a semi-synthetic dataset from a reference dataset for benchmarking. scDesign3 first estimates gene-specific mean and dispersion parameters, conditional on cell type and batch. Then, in combination with randomly sampled cell-type and batch labels, these parameters are used by scDesign3 to simulate atlas-scale semi-synthetic data. The randomly sampled cell-type and batch labels follow the original proportions from the reference data, subject to sampling variability. The estimated parameters and labels serve as the ground truths for evaluating pipeline performance. 132

4.4 **Examples of misleading patterns in 2D visualizations. a.** Adapted from Figure 3d in Townes et al. [19], this panel shows human PBMC monocytes, where 2D locations are visibly influenced by the fraction of zero counts after applying *log1pCPM* normalization followed by t-SNE. The blue-to-red color scale indicates the fraction of zeros per cell, illustrating how preprocessing steps can distort biological structure in visualizations. **b.** Adapted from Figure 1a in Booeshaghi et al. [15], this panel shows cells whose 2D locations are affected by library size after normalization with either *log1pCP10k* (left) or *sctransform* (right), followed by UMAP. The viridis color scale denotes log-transformed library size, highlighting how normalization artifacts can create misleading visualizations. 133

4.5 ***Seurat-UMAP* visualization of semi-synthetic cells from the pilot mouse pancreas dataset, compared to their real counterparts.** The first two columns show batch-wise UMAP visualizations of semi-synthetic and real cells, each projected into the same embedding space. The third column overlays both datasets, with semi-synthetic cells downsampled to match the number of real cells, enabling a direct visual comparison of their distributions. 133

4.6 **Ranking of visualization pipelines across normalization, integration, and visualization choices.** Pipelines were ranked by the overall score, which summarizes performance across the evaluated visualization metrics. Each row represents one complete pipeline, defined by the combination of normalization, integration, and visualization methods. The left panels show the overall score, overall score standard deviation, runtime, and peak memory usage. The right panels summarize the average metric scores and their standard deviations across datasets, with larger circles indicating higher ranks or stronger relative performance for each metric. Only the top and bottom 10 pipelines are shown here . . 134

4.7 **Comparison of our benchmark study with existing studies.** This figure compares the preprocessing steps and methods used in our benchmark study with those in two prior studies—Huang et al. [1] and Sun et al. [2]. Our analysis focuses on the joint effects of normalization, integration, and visualization methods on 2D visualization, evaluating a total of 180 pipelines. We highlight default methods used by popular scRNA-seq analysis tools, Seurat and Scanpy, which are distinguished using different font styles. Our study leverages semi-synthetic scRNA-seq datasets with unbiased ground truths, allowing for evaluation using nine quantitative metrics. In contrast, Huang et al. evaluated seven visualization methods and examined the effects of three combinations of normalization and dimensionality reduction (shown under the Preprocessing step in the figure). Notably, the output dimensions vary across dimensionality reduction methods (e.g., PCA vs. GLM-PCA), potentially limiting result comparability. Moreover, their assessment of preprocessing impacts was based solely on visual inspection, without quantitative evaluation. Sun et al. explored 11 dimensionality reduction methods combined with three normalization approaches, focusing on low-dimensional outputs beyond 2D. As a result, their study does not provide a direct assessment of 2D visualization quality. For clustering, they assessed the effects of normalization by computing normalized mutual information between known cell types and clustering results across various algorithms applied to the low-dimensional outputs. For trajectory inference, they evaluated normalization effects using Kendall correlation between known pseudotime and pseudotime inferred from the low-dimensional outputs. Both existing studies relied on cell-type labels in real datasets as ground truths, without addressing potential biases. Moreover, they examined fewer method combinations and did not account for the integration step. 135

4.8 **Complete ranking of visualization pipelines across normalization, integration, and visualization choices.** Pipelines were ranked by the overall score, which summarizes performance across the evaluated visualization metrics. Each row represents one complete pipeline, defined by the combination of normalization, integration, and visualization methods. The left panels show the overall score, overall score standard deviation, runtime, and peak memory usage. The right panels summarize the average metric scores and their standard deviations across datasets, with larger circles indicating higher ranks or stronger relative performance for each metric. 136

LIST OF TABLES

2.1 **Terminology and metric definitions used in this study.** 14

2.2 **Notations used in metric definitions (defined for one dataset: BM complete, BM partial 1 or 2, or pancreas complete or partial).** 15

2.3 **Comparison of participant-level annotation accuracy and cell-type-level average false positive rate (AFPR) between the AI agent Biomni and the participants.** * For alpha cells in the pancreas complete dataset, Biomni's detection rate was 0.000. 26

S2.4 Questionnaire for participants. 37

S3.1 Summary of datasets used in this study. 83

S3.2 **Ribosomal and mitochondrial gene enrichment among IntegrateRigor-defined batch-stable and batch-unstable genes across datasets.** Here, BSG denotes batch-stable genes. BUSG denotes batch-unstable genes. 89

4.1 **Five real datasets used as references for semi-synthetic data generation.** The semi-synthetic datasets were generated to match the cell-type and batch proportions of these real datasets, with the total number of cells randomly sampled between 0.5 and 1 million. 106

4.2 Summary of the software versions for each method 110

4.3 Normalization methods evaluated in the visualization pipeline benchmark. . . . 111

ACKNOWLEDGMENTS

This dissertation was made possible by the many people who accompanied me through the years of my doctoral study, offering their time, wisdom, encouragement, and friendship along the way.

I am most deeply indebted to my advisor, Dr. Jingyi Jessica Li. Her intellectual rigor, uncompromising standards, and belief in my work have profoundly influenced the way I think, write, and conduct research. She has taught me to ask clearer questions, pursue stronger evidence, and approach scientific problems with both discipline and imagination. Just as importantly, she has shown me what mentorship can be at its best: thoughtful, generous, principled, and deeply human. Outside of research, the moments we shared through music, art, conversation, and laughter became some of the most meaningful memories of my PhD years. The example she has set, as a scholar and as a mentor, will continue to guide me long after this dissertation is complete.

I am sincerely grateful to my doctoral committee members, Dr. Mark S. Handcock, Dr. Sudipto Banerjee, and Dr. Ker-chau Li. Their insightful comments, careful questions, and generous support helped me refine my ideas, strengthen my arguments, and see this work from broader perspectives.

I also thank the current and former members of the JSB Lab for creating an intellectually lively and supportive environment. The many discussions, suggestions, and exchanges I had with lab members played an important role in shaping this dissertation. I am particularly thankful to Qingyang Wang, Chengfeng Jiang, Dr. Changhu Wang, and Dr. Guanao Yan for their friendship, encouragement, and companionship throughout my time in the lab. I am grateful to Dr. Xinzhou Ge and Ziqi Rong for their collaboration, and to Will Boudeau and Shuyang Qian for their valuable contributions to the second chapter. I also wish to thank Dr. Yuying Xie and Dr. Rongrong Wang for their guidance and support, which helped me move forward on this journey.

Above all, I thank my parents, Ying Wang and Wenyi Zhai, and my brother, Kun Zhai.

Their unconditional love, patience, and faith in me have sustained me through every stage of this path. Finally, I thank my friends, including Zeyi Wang, Yun Cong, and Xiran Zhang, and many others, for their companionship, humor, and steady presence, which brought warmth and balance to this journey.

VITA

2017–2021 B.S. in Mathematics and Applied Mathematics, Department of Mathematical Science,
University of Science and Technology of China

PUBLICATIONS

Zhai, Z., Lei, Y.L., Wang, R. and Xie, Y., 2022. Supervised capacity preserving mapping: a clustering guided visualization method for scRNA-seq data. *Bioinformatics, 38(9), pp.2496-2503.*

Zhai, Z., Wang, C., Jiang, C., Rong, Z., Li, J. J., IntegrateRigor: annotation-free integration optimization for cell identity recovery reveals cancer–immune interface niches. bioRxiv, pp.2026-05

Wang, Q., **Zhai, Z.**, Lian, Q., Song, D., Li, J. J., 2023. Categorization and analysis of 14 computational methods for estimating cell potency from single-cell RNA-seq data. arXiv preprint arXiv:2309.13518.

Hemminger, Z., De Ocampo, H., Xie, F., **Zhai, Z.**, Li, J.J., and Wollman, R. 2026. CIPHER: An end-to-end framework for designing optimized aggregated spatial transcriptomics experiments. bioRxiv, pp.2026-01.

Gong, W., Taner, H. F., Wu, Y., He, Y., Zhou, X., Li, Z., Hu, X., Ursin, C., Debnath, K. C., Okuyama, K., Hu, Q., Donnelly, C. R., Nör, F., Perera, C. D., Bellile, E., Yunesi, A., **Zhai, Z.**, Zhao, M., Cheng, W., Fitzsimonds, Z. R., Broses, L., Li, J., Demehri, S., Nagrath, D., Wolf, G. T., Sikora, A. G., Yu, Y., Wen, H., Wei, L., Chinn, S. B., Myers, J. N., Akira, S., Xie, Y., Moon, J. J., and Lei, Y. L. 2026. BATF2 is a glutamine-responsive

tumour suppressor required for type-I interferon-dependent anti-tumour immunity. Nature Communications 17, 1271.

CHAPTER 1

Introduction

Single-cell RNA sequencing (scRNA-seq) and spatial transcriptomics have transformed modern biology by enabling gene expression measurements at single-cell and spatial resolution, making it possible to study cellular heterogeneity, tissue organization, developmental processes, and disease mechanisms at unprecedented scale and resolution [1–4]. However, these technologies do not directly produce biological interpretation. Instead, raw count matrices are converted into biological conclusions through multi-step analytical pipelines that typically include normalization [5], feature selection [6], dimensionality reduction or integration [7], clustering [8], visualization [9, 10], and cell-type annotation [11–14] (Figure 1.1).

In a typical scRNA-seq analysis workflow, normalization is first used to adjust for differences in sequencing depth and capture efficiency across cells, so that gene expression profiles become more comparable [15–17]. Feature selection identifies informative genes, such as highly variable genes, that capture major biological or technical sources of variation while reducing noise and computational burden [6, 18]. Dimensionality reduction methods, including principal component analysis, summarize high-dimensional gene expression profiles into lower-dimensional representations [19–21]. When data are collected across samples, batches, technologies, conditions, or species, integration methods are often applied to align shared cell identities while controlling unwanted between-batch variation [22–29]. These lower-dimensional representations are then used for clustering and visualization, which support the discovery of cell populations, transitional states, and relationships among cells [8, 30–36]. Finally, cell-type annotation assigns biological labels to clusters or cells based on marker genes, reference atlases, or automated annotation tools, thereby translating computational patterns into interpretable biological identities [13, 37].

Reflecting the importance and complexity of these analytical steps, many methods have been developed for each stage of the pipeline [7, 24, 38]. As a result, analysts must make a series of methodological and parameter choices, and the subjectivity in choices can propagate through the pipeline to affect downstream cell identity discovery, dataset comparison, and biological interpretation [12, 29, 39]. However, principled strategies for making these choices remain limited: many analyses still rely on default settings, heuristic decisions, or analyst experience, making it difficult to assess whether the resulting biological interpretations are robust to alternative analytical choices [25, 40, 41]. This dissertation addresses the need for more objective, transparent, and data-adaptive analytical frameworks for single-cell and spatial omics data by examining how human decisions, integration choices, and visualization pipelines affect the reliability and reproducibility of downstream biological interpretation.

1.1 How does human subjectivity affect the reliability of single-cell RNA-seq cell-type annotation?

The first part of the dissertation, described in Chapter 2, presents a human-subject study of scRNA-seq cell-type annotation. Although automated annotation methods have been proposed [42–44], expert manual annotation remains common, especially when automated methods fail or produce low-confidence results [37]. Manual annotation involves a multi-step and highly flexible pipeline, including quality control, normalization, highly variable gene selection, dimensionality reduction, clustering, visualization, differential expression analysis, and marker-gene-based label assignment [8, 10, 45–49]. To directly evaluate how subjective decisions affect annotation reliability, we designed three annotation tasks and assigned them to graduate students using the Seurat V5 pipeline [50]. By collecting participants' pipeline choices, parameter values, marker genes, annotation results, and background information, this study shows that human subjectivity substantially influences annotation outcomes. In particular, participants performed well on major cell types but struggled with finer subtypes; prior single-cell research experience was associated with higher annotation accuracy; quality-control choices were especially important; and confirmation bias could drive participants

to misannotate cells according to prior expectations. These findings highlight the need for transparent reporting of functions, parameters, marker genes, annotation evidence, and decision processes in single-cell studies.

1.2 IntegrateRigor: annotation-free integration optimization for cell identity recovery reveals cancer–immune interface niches

The second part of the dissertation, detailed in Chapter 3, presents IntegrateRigor, a data-driven and method-agnostic framework for optimizing single-cell and spatial transcriptomics integration. Current integration practice remains limited by two major issues: generic highly variable gene selection may include genes sensitive to between-batch variation, and existing evaluation metrics either require cell identity annotations or mainly assess batch mixing without detecting over-integration [6, 12, 24, 28, 51–56]. IntegrateRigor addresses these limitations by identifying batch-stable genes through an annotation-free batch stability score, evaluating integrated embeddings using complementary annotation-free metrics for batch alignment and cell identity preservation, and aggregating these metrics into an integration score for dataset-specific gene selection, hyperparameter tuning, and method selection. In an application to colorectal cancer single-cell and spatial transcriptomics, IntegrateRigor uncovered previously unannotated putative immune–epithelial niches and interface states that were obscured under default integration settings [57]. Across six integration tasks and five widely used integration methods, including Seurat-RPCA [48], Harmony [56], scVI [58], FastMNN [59], and LIGER [60], IntegrateRigor consistently improved integration performance, demonstrating the value of transforming integration from a heuristic trial-and-error step into a principled, objective, and data-adaptive procedure.

1.3 Benchmarking of the single-cell cell-type visualization pipeline on atlas-scale scRNA-seq datasets

The third part of the dissertation, described in Chapter 4, benchmarks the full scRNA-seq visualization pipeline on atlas-scale datasets. Although prior studies have benchmarked individual steps such as normalization, integration, or visualization [15, 21, 38, 61–64], evaluating each step separately does not guarantee that the best individual methods form the best overall pipeline. Existing benchmarks also often rely on real-data labels or geometry-related ground truths that may be biased by the same preprocessing steps being evaluated, and they often do not fully address the scale of modern atlas-level datasets [65, 66]. To address these limitations, this study evaluates 180 pipelines combining normalization, integration, and visualization methods, using semi-synthetic data generated by scDesign3 [67] to provide unbiased ground truths while preserving realistic single-cell structure. Each pipeline is assessed using metrics for preservation of cell-type distances and variances, removal of confounding factors, separation of distinct cell types, avoidance of within-cell-type trajectory-like artifacts, batch-effect removal, and computational efficiency. Together, these three studies move single-cell and spatial omics analysis toward more transparent, objective, and pipeline-aware practice.

In summary, my doctoral research focuses on improving the rigor of single-cell and spatial omics data analysis through objective, principled metrics and systematic benchmarking. This dissertation presents three related contributions: an investigation of the subjectivity observed in students' single-cell analysis behavior, an integration optimization framework, and a benchmarking study of visualization pipelines. These contributions are described in Chapters 2, 3, and 4, respectively.

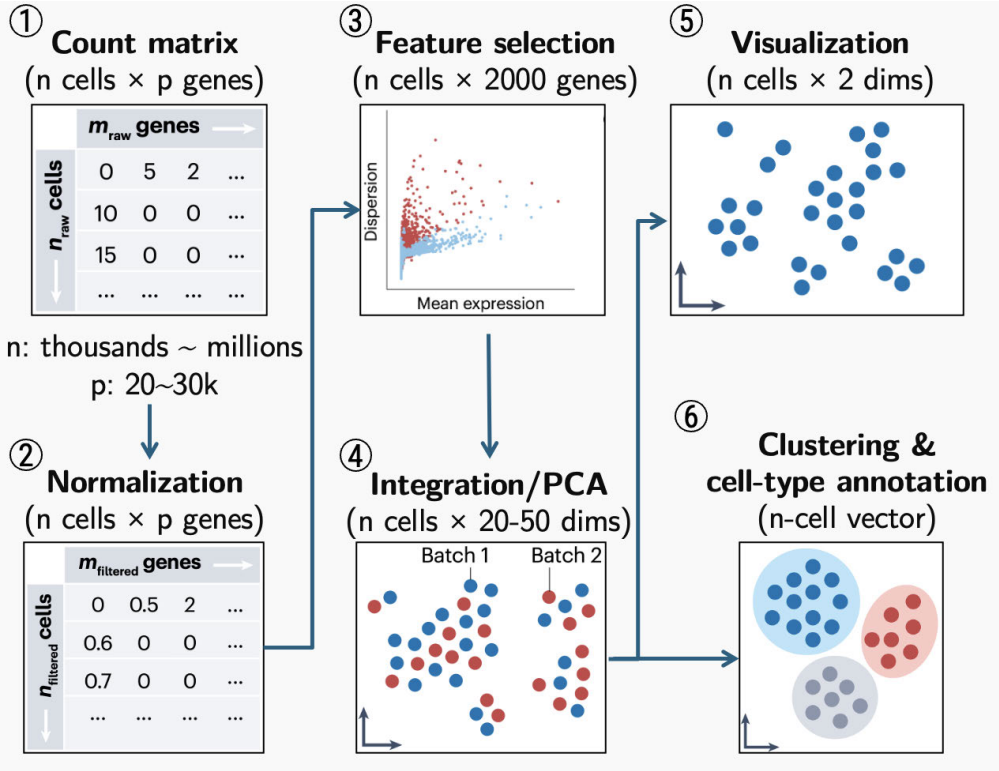


Figure 1.1: Overview of a typical single-cell RNA-seq analysis workflow. Single-cell RNA-seq analysis in this dissertation begins with a gene-by-cell count matrix, followed by normalization to reduce technical variation such as differences in sequencing depth. A subset of informative genes is then selected for downstream analysis, typically reducing the feature space from tens of thousands of genes to a smaller set of highly variable genes. Dimension reduction or integration methods, such as PCA or batch-correction approaches, are applied to generate low-dimensional cell embeddings. These embeddings are then used for visualization, clustering, and cell-type annotation, producing interpretable cell groups and cell identity labels. The figure was adopted from [12]

CHAPTER 2

How does human subjectivity affect the reliability of single-cell RNA-seq cell-type annotation?

2.1 Introduction

Single-cell RNA sequencing (scRNA-seq) has reshaped modern biology, making it possible to examine gene expression patterns in individual cells. This technology has uncovered unprecedented cellular heterogeneity, provided new insights into developmental processes and disease mechanisms, and opened avenues for precision medicine [1]. A key step in scRNA-seq analysis is cell-type annotation, where each cell is assigned a cell-type label [13, 37]. Accurate cell-type annotation is critical, as it provides transferable knowledge, i.e., cell-type labels that can be used to compare across datasets and facilitate biological interpretation, such as disease-associated changes in cell-type proportions and gene expression levels.

Although a variety of automated cell-type annotation methods have been proposed [42–44], expert manual annotation remains the common practice, especially when automated methods fail or yield low-confidence results [37]. The manual cell-type annotation process typically follows a multi-step computational pipeline. Raw count data are first preprocessed through quality control, normalization, and highly variable gene (HVG) selection, followed by dimensionality reduction (e.g., PCA) and clustering (e.g., Louvain [8, 45] or Leiden [46]) to identify similar groups of cells. Additional steps, such as visualization to refine cluster structure [10, 47] and identifying cluster-specific differentially expressed genes and matching them to prespecified marker genes, are essential for supporting subsequent annotation decisions, namely assigning cell-type labels to clusters. Popular packages such as Seurat V5 (R) [48] and Scanpy (Python) [49] provide comprehensive functions that implement each of

these steps, supporting the manual annotation process (Figure 2.1).

Subjectivity is a challenge in manual cell-type annotation for scRNA-seq data [37, 68]. Analysts must make numerous subjective choices within the analysis pipeline, including selecting functions, tuning parameters, and choosing prespecified reference marker genes [37]. As a result, the final pipeline configuration is inherently subjective and can substantially influence cell-type annotation results. Although various benchmark studies have compared computational methods in the analysis pipeline [24, 69, 70], these in-silico evaluations cannot capture the human decision-making process that introduces subjectivity into analysis results. Two important yet unexplored questions are how these subjective choices affect cell-type annotation and which factors are most influential.

To fill this gap, we designed three distinct cell-type annotation tasks and assigned them to graduate students, including both master's and PhD students. These students were enrolled in a graduate-level statistical bioinformatics course at UCLA during two quarters (Spring 2024 and Winter 2025). A total of 37 students consented to participate. The three tasks were constructed to represent varying levels of prior expectations about potential cell types: (1) a dataset containing all possible cell types from the list we provided, (2) a dataset including only a subset of those cell types, and (3) a dataset focusing exclusively on identifying two rare cell types. This design allows us to examine how participants' prior expectations about cell-type existence can influence their annotation decisions. Participants completed the three annotation tasks using the Seurat V5 pipeline, popular in the field (Figure 2.1). We collected information on their pipeline-tuning procedures, including the functions and parameter values they selected and their prespecified reference marker genes for cell-type annotation. In addition, participants completed a questionnaire to provide their educational and research background. Based on their responses, we studied how their annotation results were influenced by their educational and research background and subjective decisions.

This study provides novel insights into how analysts' decisions influence scRNA-seq cell-type annotation. It emphasizes the need for standardized definitions of cell-type marker genes and transparent reporting of analysis pipelines to improve reproducibility in single-cell research. Our analysis shows that while participants overall performed well in annotating

major cell types (e.g., T cells and B cells), they faced challenges in distinguishing among cell subtypes (e.g., CD4 T cells and CD8 T cells). In addition, annotation accuracy is strongly associated with participants' research experience: those with prior single-cell publications achieved significantly higher accuracy than others (one-sided Mood's median test, p-value = 0.01). In addition, we found that human subjectivity substantially affects cell-type annotation. Cell clustering requires subjective choices of functions and parameters, and using non-default tuned settings generally improves clustering accuracy. However, assigning cell-type labels to clusters depends heavily on prespecified marker genes and subjective judgments of their expression, so parameter tuning has far less influence on annotation accuracy than on clustering accuracy. Among the parameters within the Seurat pipeline, we observed that data-quality control (i.e., filtering low-quality cells and genes) is the most important step for achieving high annotation accuracy. Moreover, we identified a key factor—confirmation bias—which drives participants to misannotate cells in line with their prior expectations of which cell types should exist.

To further explore the potential for the automated annotation approach, we evaluated a general-purpose biomedical AI agent, Biomni [71], by applying it to the same datasets with the same tasks. This agent can automatically analyze tasks, select appropriate computational methods, and write code to complete the tasks, similar to human analysts. While it reduces manual efforts, our results indicate that it is not yet a viable replacement for human analysts.

Together, these findings highlight that human subjectivity influences scRNA-seq cell-type annotation, yet the subjective process remains indispensable. To mitigate this impact, transparency in analysis pipelines is essential. As in reporting experimental procedures in wet-lab studies, researchers should clearly document the functions, parameters, and reference marker genes used for annotation, provide evidence and explanations supporting how cell-type labels were assigned, and justify their decision processes. Such transparent reporting enables more rigorous evaluation, facilitates replication, and ultimately enhances the reliability of single-cell analyses.

2.2 Materials and methods

2.2.1 Study design

Each participant was instructed to complete (1) three cell-type annotation analysis tasks and (2) an educational and research background questionnaire. Throughout this manuscript, the terms "students" and "participants" are used interchangeably. Each participant was asked to perform the cell-type annotation task on three variant datasets (two from the bone marrow dataset and one from the pancreas dataset). Participants were required to annotate cell types using the three provided UMI count matrices, with cell barcodes anonymized using generic labels (e.g., cell 1, cell 2, etc.) to prevent information sharing across students. To ensure meaningful comparisons, participants followed the Seurat V5 [48] annotation pipeline and were only allowed to use the functions and parameter options shown in Figure 2.1.

After their analysis, participants were required to submit the following results for each dataset:

- annotated labels for all cells in the dataset;
- a record of all parameter values tuned and the final parameter values chosen for each function in the pipeline;
- the reference marker genes used for cell-type annotation.

Exact instructions for the annotation tasks are provided in the *Supplemental Material Section S2.8.3*.

2.2.1.1 Cell-type annotation tasks 1–2: bone marrow datasets

For the first two cell-type annotation tasks, participants were asked to annotate two variants of the original bone marrow datasets. We split the original bone marrow dataset into three parts and subset cell types to obtain three variants: (1) **bone marrow (BM) complete dataset**, comprising 3,093 cells representing all 21 annotated cell types; (2) **bone marrow**

(BM) partial 1 dataset, a subset of 2,982 cells with seven cell types removed; and (3) **bone marrow (BM) partial 2 dataset**, a subset of 3,184 cells with a different set of six cell types removed. These datasets are illustrated in Figure 2.2a–b, with the cell types excluded from the BM partial datasets highlighted by colored dashed circles in Figure 2.2a and the legend. To minimize information sharing among participants, we randomly sampled 90% of the cells for each variant dataset before assigning the datasets to participants.

For task 1, each participant received a downsampled BM complete dataset. For task 2, participants were randomly assigned either a downsampled BM partial 1 or 2 dataset. A brief illustration of this data assignment is shown in Figure 2.2d.

For both tasks, participants were provided with the same list of all 21 cell types from the original dataset. They were informed that some cell types might be absent but were not told which ones were missing in their assigned datasets. Because the two BM partial variants were created by selectively removing certain cell types, this design allowed us to investigate the confirmation bias in cell-type annotation, particularly in cases where the actual number of cell types was smaller than expected.

2.2.1.2 Cell-type annotation tasks 3: pancreas dataset

For the third cell-type annotation task, participants were asked to annotate one pancreas dataset. We generated two variants: (1) **pancreas complete dataset** in which all 14 cell types were retained (1,884 cells and 26,099 genes), and (2) **pancreas partial dataset** in which two rare cell types, pancreatic polypeptide (PP) and alpha cells, were removed (1,784 cells and 26,099 genes). These variants are illustrated in Figure 2.2c. Participants were randomly assigned to annotate one of these two datasets (Figure 2.2d), without being informed whether their dataset actually contained PP or alpha cells. We also randomly sampled 90% of the cells from each variant before distributing them to participants to avoid information sharing among participants.

For task 3, participants were instructed to focus on identifying two rare cell types: PP cells and alpha cells, a scenario common in practice when analysts aim to detect rare cell

types. It enabled us to assess confirmation bias—specifically, whether focusing on rare cell types increases the likelihood of false identification.

2.2.1.3 Questionnaire

In addition to the three cell-type annotation tasks, participants were asked to complete a background questionnaire. The questionnaire was designed to capture key aspects of each participant's educational and research background across three main dimensions: (1) educational background, including majors, degrees, and courses taken; (2) research background, including publications and data analyses performed in the publications; and (3) self-assessment of familiarity with scRNA-seq analysis, including confidence in interpreting results. A summary of the questionnaire items is provided in Table S2.4.

2.2.2 Participant recruitment

For this study, we obtained IRB approval and recruited graduate students enrolled in UCLA STATS M254 (cross-listed as BIOINFO 223), an advanced course on statistics and scRNA-seq analysis offered in Spring 2024 and Winter 2025. Participation in the study was entirely voluntary and had no impact on their course grades. Informed consent was obtained from all participants, including explicit permission to use their annotation results and questionnaire responses for research purposes.

Of the 37 students who agreed to participate, 28 provided completed and valid results (i.e., cell-type annotation results for all provided cells) we included in the study. One additional participant provided incomplete results for task 2. As a result, we analyzed results from 28 participants for task 2 and results from 29 participants for tasks 1 and 3.

2.2.3 AI agent for the same cell-type annotation tasks

Biomni [71] is an AI agent designed to automatically complete biomedical-related tasks, including single-cell data analysis. It will find and execute established single-cell analysis tools for tasks such as cell-type annotation. It integrates curated marker gene databases

and can retrieve additional information from online sources when it feels necessary or the information is explicitly requested in the prompt. Without human involvement other than the prompt, Biomni can automatically handle parameter tuning and marker-gene selection when these steps are involved, and provide a transparent process. We accessed Biomni through its official web application (https://app.biomni.stanford.edu; access date: Oct. 20th, 2025). We gave Biomni the same prompt as human participants but without the Seurat V5 requirement, as its web application did not consistently support Seurat installation, and allowed it to use any programming language or pipeline that completed the annotation task. The exact prompt we used is included in the *Supplementary Materials Section S2.8.4*.

2.2.4 Cell-type annotation evaluation metrics

We collected the original cell-type annotations from the publications of the bone marrow and pancreas datasets [72, 73] and treated them as expert reference annotations, against which each participant's annotations were compared.

For the bone marrow datasets, we evaluated participants' performance using four metrics: two at the participant level (clustering accuracy and annotation accuracy) and two at the cell level (majority-vote annotation accuracy and annotation variability). We only calculated these metrics on the bone marrow datasets (tasks 1 and 2), as task 3 on the pancreas dataset involved annotating only two cell types and was specifically designed to examine confirmation bias. The participant-level metrics are (1) **clustering accuracy**, quantified by the Adjusted Rand Index (ARI), and (2) **annotation accuracy**. The annotation accuracy was evaluated based on a predefined cell-type hierarchy for the bone marrow dataset; this hierarchy was not provided to participants, who only received a flattened list of 21 potential cell types corresponding to the hierarchy's leaf nodes. For each participant, annotation accuracy was defined as the average of one minus the normalized path length between the participant's annotated and reference cell types across all cells in their dataset. This metric penalized misannotations between closely related cell types less than those between more distant types. Each participant received one clustering accuracy score and one annotation accuracy score.

The cell-level metrics are (1) **majority-vote annotation accuracy**, defined as a cell's annotation accuracy using the same hierarchy based on the cell's majority-vote annotation across participants (i.e., the cell-type label most frequently assigned by participants); for this metric, the definition mirrors the participant-level version and was computed as one minus the normalized path length between the majority-vote cell type and the reference cell type, and (2) **annotation variability**, the number of distinct cell types assigned to the same cell across participants, reflecting the degree of disagreement or uncertainty in annotation. Each cell in the three datasets (BM complete, BM partial 1 or 2, and pancreas complete or partial), prior to the 90% downsampling for participant assignments, received one majority-vote annotation accuracy score and one annotation variability score.

To assess confirmation bias in Task 3, we evaluated participants' annotations for the pancreas dataset using a cell-type-level metric, the **average false positive rate (AFPR)**. For each cell type, AFPR was defined as the ratio of the number of cells incorrectly annotated as that type to the total number of cells that did not belong to that type (i.e., the numerator is a subset of the denominator), averaged across participants. We computed this quantity separately for participants who received the pancreas complete dataset and those who received the pancreas partial dataset, yielding one AFPR score per cell type for each group. This design enables direct comparison between participants whose datasets contained the two rare target cell types (PP and alpha cells) and those whose datasets did not. As a baseline, we also computed AFPR for the bone marrow dataset, where no specific target types were emphasized and thus no confirmation bias is expected. Elevated AFPR values for PP and alpha cells, especially in the partial-dataset group, whose datasets lacked these two rare types, relative to the bone marrow baseline or the complete-dataset group would indicate a tendency for participants to over-assign cells to these targeted types.

The terminologies and explanations of the metrics are summarized in Table 2.1. And the notations used throughout are summarized in Table 2.2.

Terminology	Level	Explanation
Reference annotations	Cell level	Expert cell clusters with cell-type labels from each dataset's original publication.
Participant annotations	Cell level	Cell clusters and cell-type labels annotated by each participant for each assigned dataset.
Clustering accuracy	Participant level	Adjusted Rand Index (ARI) measuring agreement between each participant's clustering and the reference clustering (cluster structure only; labels not used).
Annotation accuracy	Participant level	Accuracy comparing each participant's annotated cell types with the reference labels, defined as one minus the normalized path length in the cell-type hierarchy and averaged across all cells.
Majority-vote annotation accuracy	Cell level	Accuracy comparing each cell's majority-vote annotation with its reference label, computed using the same hierarchy-based definition.
Annotation variability	Cell level	Number of distinct cell types assigned to the same cell by different participants, reflecting annotation disagreement.
Average false positive rate (AFPR)	Cell-type level	For each cell type, the ratio of cells incorrectly annotated as that type to all cells not belonging to it, averaged across participants; used to detect confirmation bias in Task 3.

Table 2.1: Terminology and metric definitions used in this study.

Symbol	Description
S_i	Set of cells assigned to participant i (after 90% downsampling).
$N_i = \|S_i\|$	Total number of cells in the dataset assigned to participant i.
n_{ijk}	Number of cells in the dataset that participant i annotated using the j-th cell-type index and that have the k-th reference cell-type index; the indices j and k are independent indexing positions with no correspondence to actual cell-type labels.
ℓ_{ic}	Cell-type label assigned by participant i to cell c, defined only for cell $c \in S_i$.
ℓ_c^0	Reference cell-type label for cell c.
ℓ_c^{m}	Majority-vote label for cell c, computed from the labels assigned by all participants who received cell c in their dataset.
$\boldsymbol{\ell}_c$	Set of all cell-type labels assigned to cell c by different participants who received that cell.

Table 2.2: Notations used in metric definitions (defined for one dataset: BM complete, BM partial 1 or 2, or pancreas complete or partial).

2.2.4.1 Participant-level clustering accuracy

We used the ARI to evaluate the accuracy of participants' clustering results. It measures how closely each participant's clusters align with the reference clusters from the original publication, comparing only the cluster structure and not cell-type labels. Let

$$N_i = \text{total number of cells in the dataset assigned to participant } i,$$

n_{ijk} = number of cells in the dataset that participant i annotated using the j-th cell-type index and that have the k-th reference cell-type index,

$$a_{ij} = \sum_k n_{ijk}, \quad b_{ik} = \sum_j n_{ijk}.$$

Then the ARI for the i-th participant is defined as

$$\mathrm{ARI}_i = \frac{\sum_{jk}\binom{n_{ijk}}{2} - \frac{\sum_j \binom{a_{ij}}{2}\sum_k \binom{b_{ik}}{2}}{\binom{N_i}{2}}}{\frac{1}{2}\left[\sum_j \binom{a_{ij}}{2} + \sum_k \binom{b_{ik}}{2}\right] - \frac{\sum_j \binom{a_{ij}}{2}\sum_k \binom{b_{ik}}{2}}{\binom{N_i}{2}}} \in [-1, 1].$$

A high ARI value indicates that the participant's clusters closely match the reference clusters, reflecting high clustering accuracy.

2.2.4.2 Participant-level annotation accuracy

In addition to clustering accuracy, which compares only cluster structure, the hierarchy-based annotation accuracy directly evaluates cell-type labels in a participant's annotation while accounting for biological similarity among cell types. Mismatches between closely related types (e.g., CD4 T cells vs. CD8 T cells) are penalized less than errors across distant lineages (e.g., CD4 T cells vs. memory B cells). We used a cell-type hierarchy, where nodes represent cell types and edges encode lineage relationships, as shown in Figure 2.3. For each cell c in each dataset before 90% downsampling, let ℓ_c^0 denote its reference label and ℓ_{ic} the label assigned by participant i, which is defined only for cell $c \in S_i$, where S_i is the set of cells assigned to participant i. The annotation accuracy for participant i is defined as

$$A_i = 1 - \frac{1}{N_i}\sum_{c \in S_i} \frac{d(\ell_c^0, \ell_{ic})}{D} \in [-1, 1],$$

where $d(\cdot, \cdot)$ is the shortest path length between two nodes in the cell-type hierarchy and $D = 4$ is the hierarchy depth used for normalization. If a participant's annotation for a cell is NA, we assign $d(\ell_c^0, \ell_{ic})$ a value of $2D$ for that cell. A higher annotation accuracy value indicates stronger agreement between a participant's cell-type labels and the reference.

2.2.4.3 Cell-level majority-vote annotation accuracy

For each cell in each dataset (before the 90% downsampling), we defined a majority-vote annotation accuracy to quantify the agreement between the reference label and the label

most frequently assigned by participants. Let ℓ_c^0 denote the reference cell-type label for cell c, and let $\boldsymbol{\ell}_c$ be the set of labels assigned to c by all participants who received that cell. The majority-vote label, ℓ_c^{m}, is the label with the highest frequency in $\boldsymbol{\ell}_c$.

We computed the majority-vote annotation accuracy using the same cell-type hierarchy as in the participant-level annotation accuracy, shown in Figure 2.3. The accuracy for cell c is defined as

$$1 - \frac{d\big(\ell_c^0,\, \ell_c^{\mathrm{m}}\big)}{D},$$

where $d(\cdot, \cdot)$ denotes the shortest path length between two cell-type nodes in the hierarchy, and $D = 4$ is the hierarchy's maximum depth used for normalization.

2.2.4.4 Cell-level annotation variability

To quantify the variability between participants' annotations, we also computed the number of distinct cell types assigned to the same cell by participants who received that cell in their assigned datasets, i.e., the cardinality of the unique label set $|\mathrm{unique}(\boldsymbol{\ell}_c)|$. A higher value indicates that participants assigned many distinct labels to the same cell, reflecting greater annotation variability.

2.2.4.5 Cell-type-level average false positive rate

We first defined the false positive rate (FPR) for cell type k at the participant level and then averaged these values across participants as the cell-type-level average false positive rate (AFPR). The participant-level FPR for cell type k is

$$\mathrm{FPR}_i(k) = \frac{\sum_{c \in S_i} \mathbf{1}\{\ell_{ic} = k,\ \ell_c^0 \neq k\}}{\sum_{c \in S_i} \mathbf{1}\{\ell_c^0 \neq k\}} \in [0, 1].$$

For each cell type k, we further split participants into two groups:

- P_{k+}: participants whose datasets contain cell type k,
- P_{k-}: participants whose datasets do not contain cell type k.

The corresponding AFPRs are

$$\frac{1}{|P_{k+}|}\sum_{i\in P_{k+}}\mathrm{FPR}_i(k), \qquad \frac{1}{|P_{k-}|}\sum_{i\in P_{k-}}\mathrm{FPR}_i(k).$$

2.3 Results

Because participants were only asked to annotate two rare cell types in the pancreas complete and partial datasets, we only analyzed these datasets for the confirmation bias analysis presented in *Sections 2.3.4*. For the other analyses in *Sections 2.3.1–2.3.3*, we focused solely on the bone marrow datasets.

2.3.1 Participants' annotations exhibit variability and difficulty in distinguishing cell subtypes

The two participant-level metrics revealed the difficulty of accurately annotating the cell subtypes. The first metric, clustering accuracy, assesses whether the cell clusters identified by a participant align with the cell partition determined by the reference cell-type labels, without considering agreement between the participant's labels and the reference labels. As illustrated in Figure 2.4a–b, across participants, the clustering accuracy (range: $[-1, 1]$) was 0.59 ± 0.11 (mean $\pm$ standard deviation) for the BM complete dataset and 0.54 ± 0.18 for the BM partial dataset. The second metric, annotation accuracy, measures the agreement in cell-type labeling between a participant and the reference. The annotation accuracy (range: $[-1, 1]$) was 0.53 ± 0.31 for the BM complete dataset and 0.47 ± 0.34 for the BM partial dataset. Because annotation accuracy is defined using the shortest path between participant and reference cell-type labels in the cell-type hierarchy, we report the average shortest path length (maximum possible length: 8), which was 1.88 for the BM complete dataset and 2.12 for the BM partial dataset. According to the hierarchy used to define the annotation accuracy (Figure 2.3), these values were close to the path length between cell subtypes (e.g., CD4 T vs. CD8 T, which is 2) and smaller than the path length between different cell types (e.g., T vs. B, which is 4). This indicated that participants were generally accurate at identifying major

cell types but struggled to distinguish among closely related subtypes. To create a baseline for the two metrics, we generated annotations by randomly assigning one of the 21 cell types to each cell. Notably, 2–3 participants had clustering and annotation accuracies comparable to, or even worse than, those produced by random assignment. This effect was particularly pronounced in the BM partial datasets, where annotation was more challenging because the number of cell types was smaller than expected. These observations raise concerns about the reliability and consistency of single-cell data analysis when annotation tasks deviate from analysts' expectations.

In addition, the annotation variability is shown in Figure 2.4c. This metric quantifies, for each cell, how many distinct cell types were assigned to it across all participants who were assigned to this cell. The annotation variability had a median of 4 or 5 distinct cell-type labels per cell, reflecting substantial disagreement among participants. Together with the large standard deviations in clustering and annotation accuracies, these results highlight substantial variability in cell-type annotations across analysts.

When annotations were aggregated using a majority-vote consensus across participants, the resulting cell-level annotation accuracy (range: $[-1, 1]$, averaged across cells) increased to 0.78 for the BM complete dataset and 0.74 for the BM partial dataset (Figure 2.4d), compared with participant-level accuracies of 0.59 and 0.54, respectively (Figure 2.4b). The median majority-vote accuracy was 1.00 (the maximum possible value) for both datasets (Figure 2.4d). These results suggest that, despite the variability noted above, majority voting can be an effective strategy when multiple analysts are available, although such a setup is typically not feasible for most single-cell data analyses.

2.3.2 Single-cell research experience leads to more accurate cell-type annotation

We next examined how participants' educational and research backgrounds affected annotation results, using questionnaire data that captured their educational background, publication record, and self-assessed familiarity and confidence in single-cell analysis (Table S2.4).

We identified three main binary factors that influenced clustering and annotation ac-

curacies, evaluated using one-sided Mood's median tests for each factor (p-value $\leqslant 0.1$): one related to educational background ("Study area": whether the participant's field was in the biological sciences) and two related to research background ("Publication": whether the participant had prior publications, and "Single-cell experience": whether they had prior publications involving single-cell analysis). For the study-area factor, the alternative hypothesis was that participants from non-biological fields would achieve higher accuracies than those from biological sciences, reflecting our view that computational training is especially important for single-cell data analysis. This is consistent with the context of this bioinformatics course, where students from other fields often possess stronger computational skills and high motivation to learn biological concepts. The Mood's median test [74] is robust to outliers, and does not rely on distributional assumptions.

As shown in Figure 2.5a, all three factors were significantly associated with annotation accuracy for the BM complete dataset, with one-sided Mood's median test p-values of 0.04 for study area, 0.08 for publication record, and 0.01 for single-cell experience. These factors reflect a gradient of increasingly relevant computational and research experience, ranging from having a biological academic background (often associated with less training in computation), to having prior publications (indicating formal research experience), to having prior single-cell analysis experience (the most directly relevant expertise). This gradient was reflected in performance, with median annotation accuracies of 0.63, 0.64, and 0.77, respectively.

Figure 2.5a also shows that for the BM partial dataset, which contains fewer cell types than the reference list and therefore presents a more challenging annotation task, only prior single-cell experience remained significantly associated with annotation accuracy (p-value = 0.01), whereas study area and publication record were not significant (p-values = 0.50 and 0.34). Nevertheless, the medians still increased along the same training gradient—0.59, 0.63, and 0.76—mirroring the trend observed in the BM complete dataset.

Beyond the three primary factors, we also examined the effects of additional background factors on both annotation and clustering accuracies, as summarized in Figure 2.5b. Seven binary factors were evaluated using the one-sided Mood's median test ("Mood''), and six

multi-level factors were evaluated using the Jonckheere–Terpstra test ("JT") [75], which assesses monotonic trends across ordered levels. For the BM complete dataset, six factors were significantly associated with annotation accuracy (p-value $\leqslant 0.1$). Three were binary factors reflecting prior research experience—having previous publications ("Publication"), having prior single-cell analysis experience ("Single-cell experience"), and having prior genetics/genomics analysis experience ("Genetics experience"). Two additional binary factors reflected educational background—whether the participant had previously taken a graduate-level course ("GradCourse") and whether their study area was in the biological sciences ("Study area"). For the BM partial dataset, which represents a more challenging annotation setting, only prior single-cell analysis experience ("Single-cell experience") remained significantly associated with annotation accuracy.

Overall, the questionnaire responses indicate that prior training, especially single-cell analysis experience, is crucial for achieving accurate annotation.

2.3.3 Subjectivity in label assignment reduces the benefits of parameter tuning for improved clustering

Beyond participants' backgrounds, we examined the effects of parameter choices and reference marker gene specification. We found that subjective parameter choices made before clustering mainly influenced the clustering itself, whereas subjectivity in the later cell-type label assignment had a much stronger effect on the final annotations.

As shown in Figure 2.6a, modifying nine of the thirteen parameters, ranging from quality control (QC) to visualization, to non-default values resulted in higher clustering accuracy. This effect was particularly significant, as indicated by a one-sided Mood's median test (p-value $\leqslant 0.1$), for parameters in QC ("min.cells:QC" and "min.features:QC"), the number of dimensions used in `FindNeighbors` ("dims:FindNeighbors"), and both the number of dimensions and the distance metric in `RunUMAP` ("dims:RunUMAP" and "metric:RunUMAP"). The influence of the UMAP parameters also suggests that visualization can guide the clustering process, since analysts often rely on visual inspection to judge whether their clustering re-

sults are reasonable. However, these improvements diminished once cell types were assigned to clusters. As shown in Figure 2.6b, only non-default QC-related parameters improved annotation accuracy. We argue that the gains from parameter tuning were largely masked by human subjectivity in assigning cell types to clusters.

A major source of subjectivity in label assignment is the choice of reference cell-type marker genes. One hypothesis is that using a larger set of marker genes makes cell-type assignment more challenging, as analysts must consider many genes simultaneously when deciding which cell type best matches each cluster. Consistent with this hypothesis, we observed a clear negative association between the number of reference markers used for a given cell type and annotation accuracy (Spearman correlation = –0.49, p-value = 0.0066, computed using Algorithm AS 89; Figure 2.6c). In our study, reference marker genes were those selected by participants from published literature or databases to guide cluster-to-cell-type assignment. The observed trend suggests that expanding the marker list may overwhelm analysts and exacerbate subjective variation in their labeling decisions.

Together, these observations highlight the substantial impact of human subjectivity on cell-type annotation. Prior benchmarking efforts, which focused primarily on comparing methods or tuning parameters, did not account for the uncertainty introduced by subjective human decisions during the cell-type assignment step, and therefore may overestimate the reliability of their benchmarking conclusions.

2.3.4 Confirmation bias deteriorates annotation accuracy

Finally, we evaluated how confirmation bias influenced students' annotation results in two scenarios: (1) when the dataset contained fewer cell types than expected, and (2) when participants intended to identify rare cell types that might not actually be present.

2.3.4.1 Case I: Inaccurate prior expectations about cell-type presence reduce annotation accuracy

We compared annotation results between the BM complete and partial datasets to assess how inaccurate prior expectations about cell-type presence influence annotation performance. For both datasets, participants were informed that each dataset contained cell types selected from a list of 21 possible types. However, only the complete dataset included all 21 cell types, whereas the partial dataset contained only a subset. Thus, the key distinction between the two tasks was whether participants' prior expectations about which cell types were present aligned with the dataset's actual cell-type composition.

As shown in Figure 2.4a–b, clustering accuracy did not differ significantly between the two tasks (medians of 0.594 for BM complete and 0.592 for BM partial; p-value = 0.55, one-sided Mood's median test). In contrast, annotation accuracy differed significantly between datasets (medians of 0.63 for BM complete and 0.56 for BM partial; p-value = 0.07, one-sided Mood's median test). These results indicate that inaccurate prior expectations about cell-type presence primarily affect the cell-type assignment step.

2.3.4.2 Case II: Intention to discover rare cell types increases false identification

We next examined annotation results on the pancreas dataset to illustrate how confirmation bias arises when analysts intend to discover specific rare cell types. This expectation to find a target cell type can increase the likelihood of falsely identifying cells as that type, even when the cell type is actually present.

The key distinction between the bone marrow and pancreas datasets is whether participants were focused on detecting particular rare cell types. In the pancreas dataset, participants concentrated on identifying PP cells and alpha cells, both of which are relatively rare. In contrast, in the BM partial datasets, although some cell types were absent, participants were not instructed to search for any specific type but instead to consider all possible cell types from a provided list. Our objective was therefore to assess whether focused attention on rare cell types increases the likelihood of labeling cells as those types.

To quantify the tendency to falsely annotate cells as a target type, we used the average false positive rate (AFPR) at the cell-type level (see Materials and methods). We compared the AFPRs for PP and alpha cells (2.5 percent and 2.8 percent of the pancreas dataset) with the AFPRs for the 13 cell types removed from the BM partial 1 or 2 dataset, which had comparable proportions (0.5–5 percent) in the BM complete dataset. In the pancreas complete dataset, where PP and alpha cells truly exist, their AFPRs were 0.078 and 0.077, respectively, whereas the mean AFPR for comparably sized cell types in the BM partial datasets was 0.028. In other words, when PP and alpha cells were present, their AFPRs in the pancreas dataset were roughly 2.75 times higher than those for cell types of similar abundance in the BM partial dataset. This indicates that targeted searching inflates false identification even when the target cell types genuinely exist.

A more severe issue emerged when the relevant cell types were absent. In the pancreas partial datasets, where PP and alpha cells are absent, their AFPRs increased to 0.078 and 0.157, respectively, while the mean AFPR for the 13 BM cell types when absent in the BM partial datasets was 0.019. Thus, when PP and alpha cells were absent, their AFPRs were approximately 4 to 8 times higher than those for absent cell types in the BM partial dataset. These findings demonstrate that analysts' focused intention to identify specific rare cell types substantially increases the likelihood of false identification, especially when those cell types are not actually present.

Overall, these results underscore how subjective expectations, particularly the belief that a rare cell population should exist, can amplify confirmation bias or unconscious data-driven justification during annotation. This highlights the need for careful, objective, and bias-aware decision-making when evaluating the presence of rare or novel cell types.

2.3.5 Biomni underperforms 70–90% participants in cell-type annotation accuracy

As shown in Table 2.3, the AI agent Biomni achieved substantially lower annotation accuracies than most human participants on the bone marrow datasets. Across the BM complete

and partial datasets, Biomni's annotation accuracy ranged from 0.27 to 0.31, whereas the mean accuracy among human participants ranged from 0.45 to 0.53. Overall, 70 to 90 percent of participants outperformed Biomni on these tasks.

On the pancreas datasets, where the focus was on identifying rare PP and alpha cells, Biomni showed mixed performance. In the pancreas partial dataset, where both PP and alpha cells were absent, Biomni did not label any cells as PP and had an AFPR of 0.064 for alpha cells, both lower than the participants' average AFPRs. This indicates that Biomni was less likely than participants to falsely identify missing rare cell types. However, at the individual participant level, Biomni outperformed only 78.6 percent of participants in avoiding false identification of alpha cells, suggesting that Biomni did not consistently surpass human annotators.

When PP and alpha cells were present in the pancreas complete dataset, Biomni's performance declined. Although its AFPR for alpha cells was zero, this was due not to accurate annotation but to its failure to identify any alpha cells at all, despite their presence in the data. Biomni also showed a relatively high AFPR of 0.093 for PP cells compared with the participants' average of 0.076, with 80 percent of participants performing better.

Taken together, these results suggest that despite the subjectivity inherent in human annotation, human analysts currently remain more accurate and reliable than Biomni in our study. Biomni's difficulties in detecting rare but real cell types and its inconsistent performance relative to human participants indicate that AI-based annotation tools require further development before they can replace human expertise in cell-type annotation.

Tasks 1–2 datasets	Biomni annotation accuracy	Participants' average annotation accuracy	
Bone marrow complete	0.29	0.53	89.7% participants better
Bone marrow partial 1	0.27	0.45	71.4% participants better
Bone marrow partial 2	0.31	0.48	85.7% participants better
Cell types in the task 3's datasets	**Biomni's AFPR**	**Participants' average AFPR**	
PP in pancreas complete	0.093	0.076	80.0% participants are better
Alpha in pancreas complete	0.000*	0.078	13.3% participants' AFPR = 0
PP in pancreas partial	0.000	0.157	35.7% participants's AFPR = 0
Alpha in pancreas partial	0.064	0.078	78.6% participants are better

Table 2.3: Comparison of participant-level annotation accuracy and cell-type-level average false positive rate (AFPR) between the AI agent Biomni and the participants. * For alpha cells in the pancreas complete dataset, Biomni's detection rate was 0.000.

2.4 Discussion

This study provides a systematic, human-participated evaluation of how analysts' backgrounds and subjective analytical choices shape scRNA-seq cell-type annotation results. Using a cohort of graduate students, we found that even when the analysis pipeline and available functions were standardized, participant-specific factors, including prior educational and research experience, parameter tuning, and prior expectations about which cell types were present in the datasets, introduced substantial variability into annotation outcomes.

Participants were generally able to identify major cell types correctly, but considerable variability emerged in the finer-grained assignment of cell subtypes. A majority-voting consensus across participants substantially improved annotation accuracy, suggesting that consensus-based approaches may help mitigate individual-level variability. Relevant analytical experience also played an important role: participants with prior single-cell analysis

experience achieved higher annotation accuracy.

Our investigation demonstrated that assigning cell types to clusters involves substantial subjectivity and that prior expectations can further distort annotation decisions. Although using non-default parameter values improved clustering accuracy, this benefit largely disappeared once cell-type labels were assigned to clusters, indicating that subjectivity in the cell-type assignment step is a major source of variability. In addition, increasing the number of reference marker genes used for assigning labels was associated with lower annotation accuracy, likely because analysts must evaluate many genes simultaneously without clear guidance on their relative importance. These findings highlight that treating all reference markers as equally informative is suboptimal and underscore the need for clearer, evidence-based guidelines for reference marker selection and weighting.

Prior expectations also exerted a strong influence on annotation behavior. Participants performed worse when the datasets did not match their expectations about which cell types were present, and their intention to discover rare cell types increased false identification of those types, particularly when they were absent. These observations underscore the importance of approaching annotation objectively and being aware of how prior expectations may influence labeling decisions. Clear and transparent reporting of annotation choices, including how reference markers and prior expectations were used, is essential and should be required by journals.

Although human annotation is subjective, our evaluation of Biomni, a recent AI agent, showed that its performance still lagged behind that of most human participants. Biomni frequently failed to detect rare but true cell types and exhibited inconsistent behavior across datasets, suggesting that current AI-based annotation tools, while promising, are not yet reliable enough to replace human expertise.

This study has several limitations. The cohort size was modest and consisted of graduate students, which limits statistical power and may affect the generalizability of our findings to broader or more experienced research communities. Additionally, although the analysis pipeline and available tools were standardized, other potential sources of variability, such

as differences in time spent on the task, were not controlled. Future studies with larger and more diverse participant pools, including professional researchers, will be essential to validate and extend these findings.

Despite these limitations, our conclusions highlight the substantial impact of analysts' backgrounds, parameter-tuning decisions, and confirmation bias on the reproducibility and robustness of scRNA-seq annotation. The subjectivity inherent in the cell-type assignment step underscores the need for more standardized, transparent, and objective practices, including principled reference marker gene selection and clear reporting of annotation decisions, to support consistent and reliable cell-type identification in scRNA-seq research.

2.5 Code and data availability

2.5.1 Analysis and code

The processed student annotations, questionnaire responses, and the code used for analysis are accessible at Zenodo https://doi.org/10.5281/zenodo.17795761.

2.5.2 Datasets

2.5.2.1 Bone marrow dataset

The first dataset we used is a bone marrow dataset [72]. It was sequenced from the bone marrow tissue of one healthy donor (PT15) by the 10X Chromium technology and included 21 annotated cell types (it can be accessed in https://cellxgene.cziscience.com/collections/26b5b4f6-828c-4791-b4a3-abb19e3b1952). It comprised 9,259 cells and 26,832 genes. We downsampled this dataset to create three variant datasets, each containing a different number of cell types. Each participant was randomly assigned two of these variants. Because all variants were accompanied by the same list of potentially existing cell types, this design allowed us to examine how prior expectations about existing cell types influence annotation decisions.

2.5.2.2 Pancreas dataset

The second dataset is a pancreas dataset [73]. It was sequenced from the pancreas tissue of one healthy donor (HPAP024) by the 10X Chromium technology and also had cell-type labels available (it is publicly available in https://cellxgene.cziscience.com/collections/51544e44-293b-4c2b-8c26-560678423380). It comprised 1,884 cells and 26,099 genes. We downsampled this dataset to create two variant datasets, differing in whether two rare cell types, pancreatic polypeptide (PP) cells and alpha cells, were included. Each participant was randomly assigned one of the two variants. Having participants focus only on annotating these two rare cell types, the variants were designed to examine how the tendency to search for rare cell types influences annotation decisions.

2.6 Acknowledgements

This chapter is based on my joint work with Chengfeng Jiang, Ziqi Rong, Shuyang Qian, Will Bodeau, Xinzhou Ge, the students from course M254, and my Ph.D. advisor, Dr. Jingyi Jessica Li. We are grateful for the comments and feedback from the members of the Junction of Statistics and Biology at UCLA (`http://jsb.ucla.edu`) and thank the students who agreed to participate.

2.7 Figures

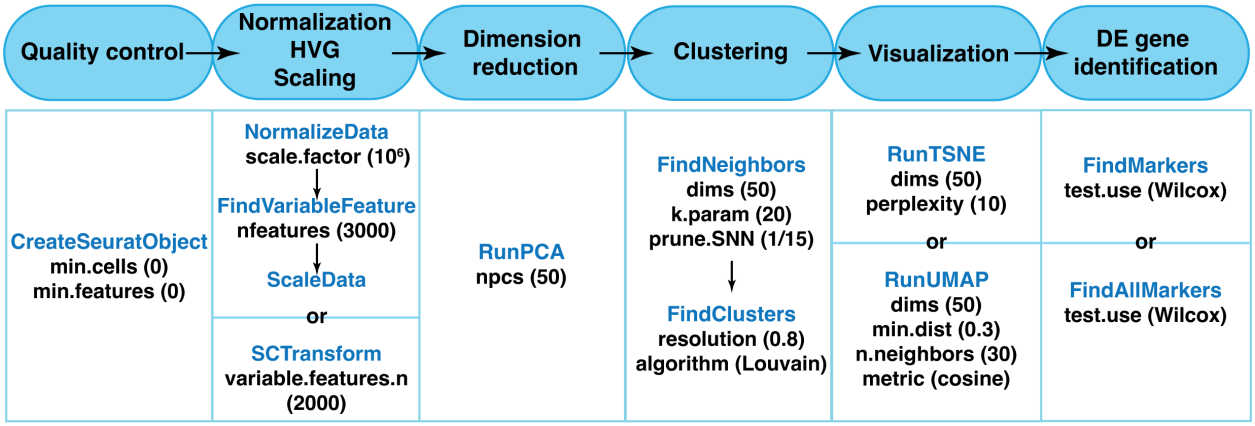


Figure 2.1: The cell-type annotation pipeline of Seurat V5 used by participants. The blue oval boxes show the steps in the Seurat V5 cell-type annotation pipeline. HVG refers to the highly variable gene identification procedure. DE genes refer to the differentially expressed genes. The blue square box below shows the functions and parameters in Seurat V5 that the participants can choose or tune. The Seurat functions participants could choose to use are indicated in blue; tunable parameters for each function are shown in black; default values for each parameter are written in parentheses.

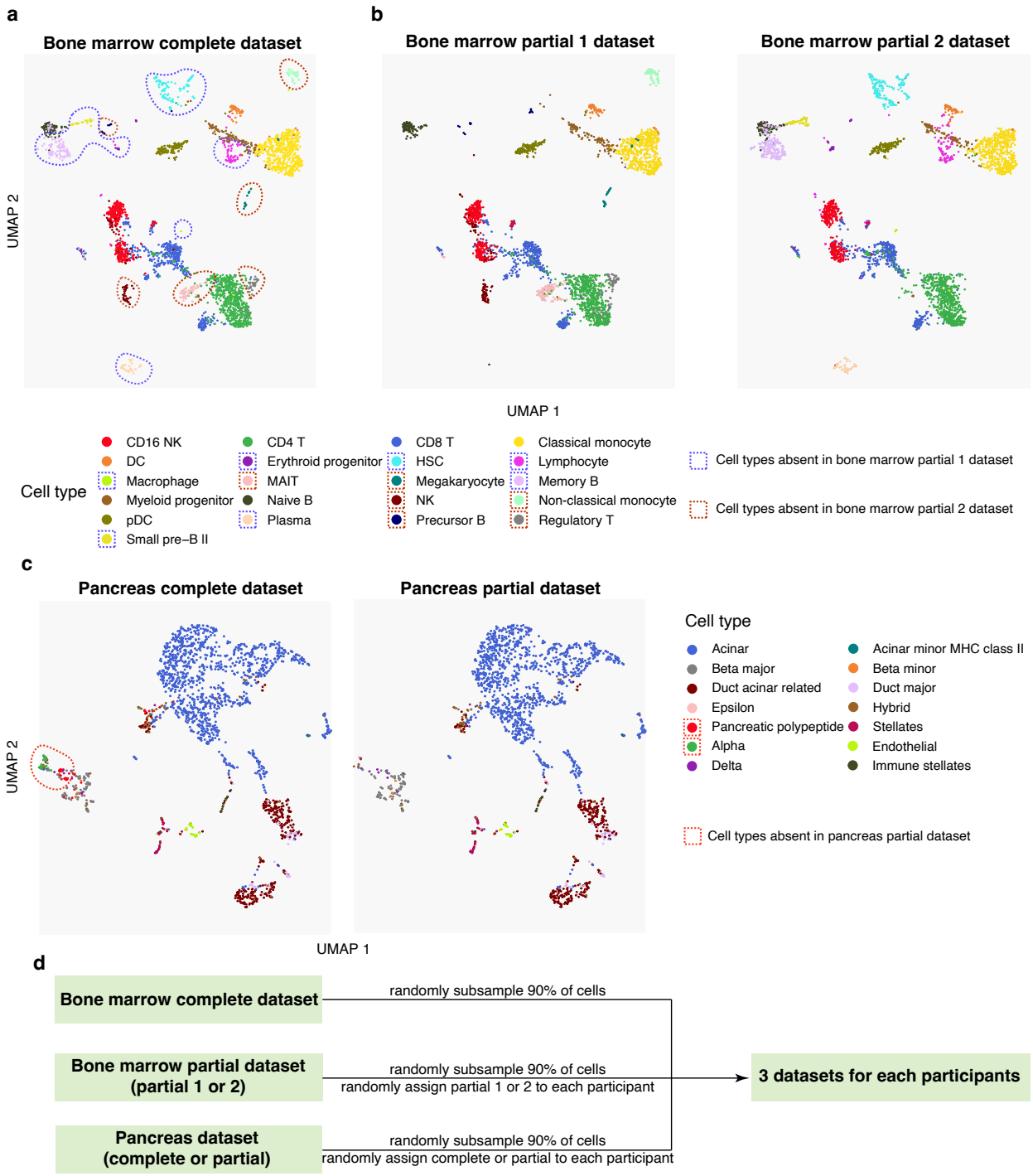


Figure 2.2: Datasets and study design. **a** The bone marrow complete dataset with complete 21 cell types from [73]. **b** Two variants of the bone marrow partial datasets, each with specific cell types removed (indicated by colored dashed contours in **a** and colored dashed squares in the legend). **c** Two variants of the pancreas dataset. The pancreas complete dataset contains all cell types, whereas the partial dataset excludes both pancreatic polypeptide (PP) and alpha cells, indicated by red dashed contour and squares. For the pancreas datasets, participants were specifically instructed to annotate only PP and alpha cells. **d** For each participant, each of their three assigned datasets contained 90% of cells randomly sampled from the corresponding dataset, with each participant randomly assigned one of the two bone marrow partial datasets and one of the two pancreas datasets.

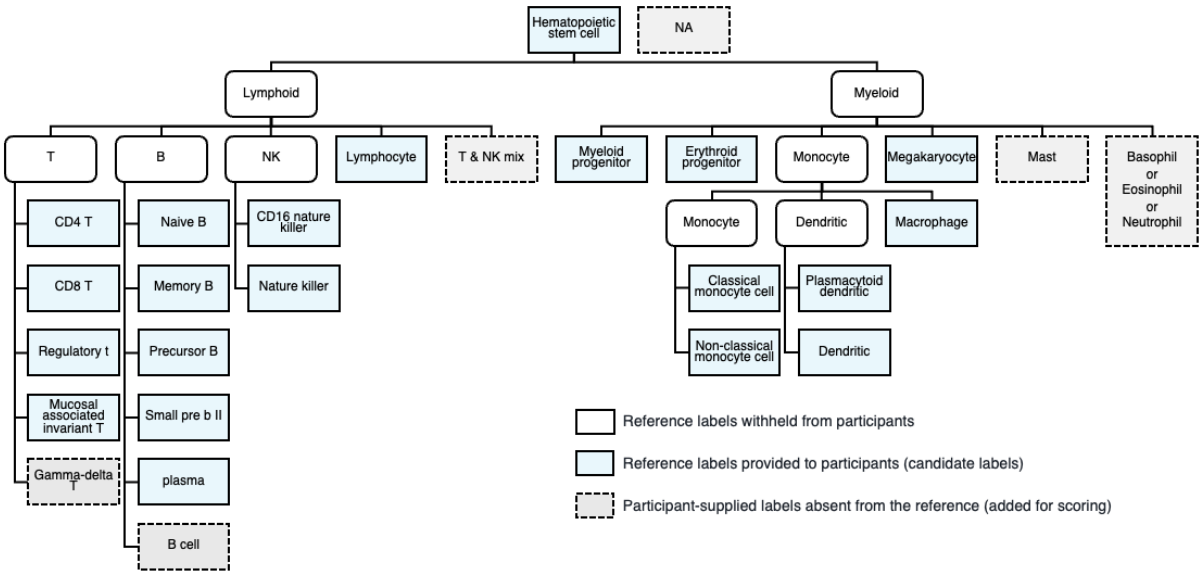
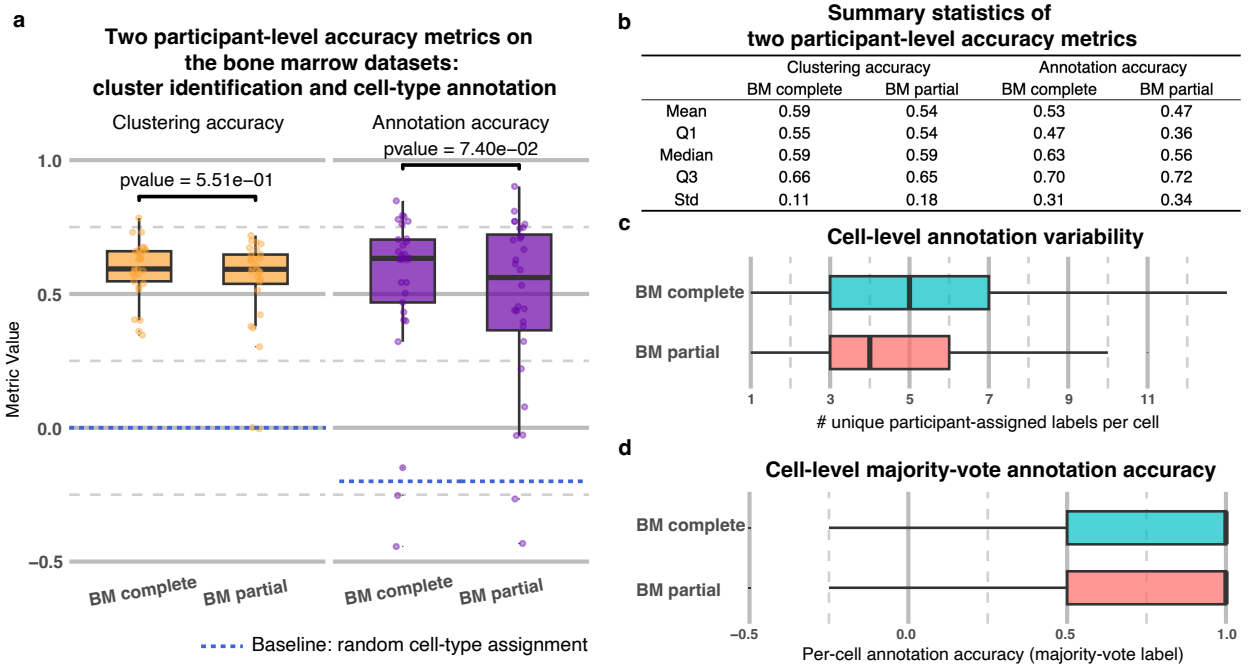


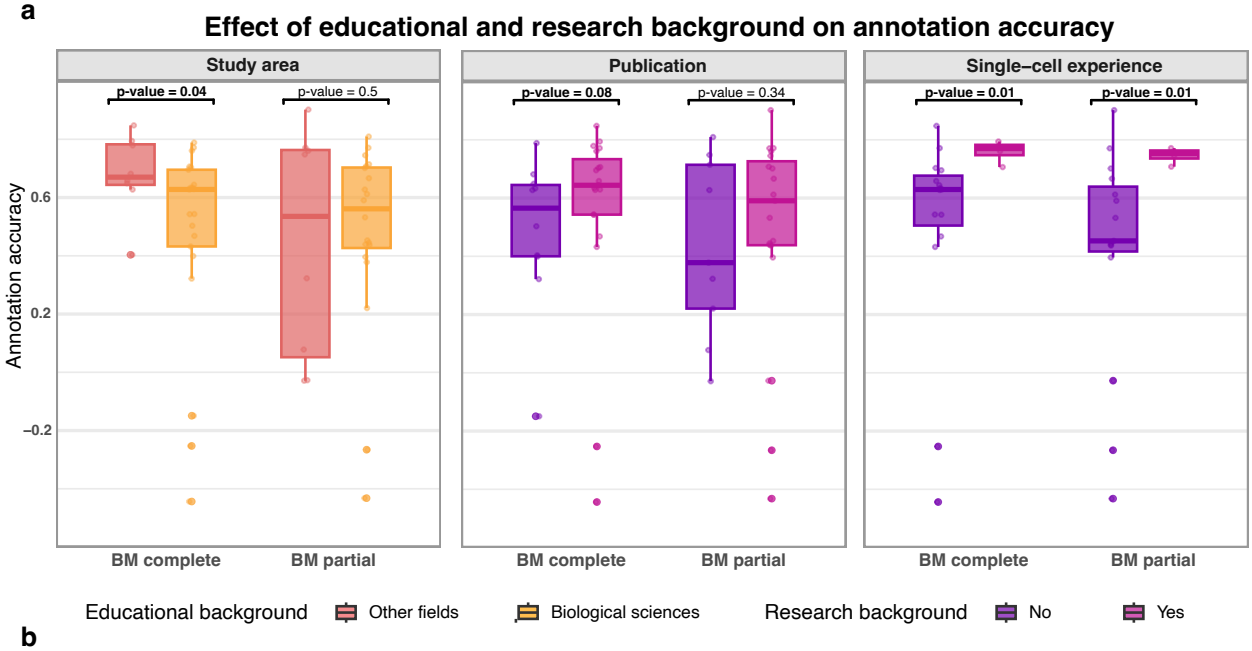
Figure 2.3: Reference cell-type hierarchy used to compute annotation accuracy for the bone marrow datasets. Participant-supplied labels not in the reference were incorporated by semantic mapping to enable accuracy scoring. Grey dashed boxes mark these participant-supplied labels absent from the reference but added for accuracy scoring; white boxes mark reference labels withheld from participants but included in the scoring hierarchy; blue boxes are the 21 reference labels provided to participants for annotation. The annotation accuracy for a cell is defined as one minus the normalized shortest path length between a participant's annotated label and the reference label in the hierarchy (range: -1 to 1), where the normalization factor is the hierarchy's maximum depth of 4. For context, the shortest path length between a CD4 T cell and a memory B cell (different cell types) is 4, whereas the shortest path length between a CD4 T cell and a CD8 T cell (same cell type; different cell subtypes) is 2. If a participant's annotation for a cell is NA, then the cell's annotation accuracy is defined as -1. Participant-level annotation accuracy is defined as the mean of the cell-level annotation accuracies for all cells assigned to that participant. This metric penalizes misannotations to closely related types less than those to more distant types.



	Clustering accuracy		Annotation accuracy	
	BM complete	BM partial	BM complete	BM partial
Mean	0.59	0.54	0.53	0.47
Q1	0.55	0.54	0.47	0.36
Median	0.59	0.59	0.63	0.56
Q3	0.66	0.65	0.70	0.72
Std	0.11	0.18	0.31	0.34

Figure 2.4: Participant- and cell-level accuracy and variability on the bone marrow complete and partial datasets. a Participant-level clustering accuracy and annotation accuracy for both bone marrow datasets, shown as boxplots. P-values are computed using the one-sided Mood's median test, with the alternative hypothesis that the BM complete dataset yields higher accuracies than the BM partial datasets, as the partial datasets do not contain all 21 cell types in the provided reference list. "BM complete'' refers to the bone marrow complete dataset; "BM partial'' includes both bone marrow partial 1 and partial 2 datasets. The blue dashed lines indicate baseline clustering and annotation accuracies obtained by randomly assigning one of the 21 cell types to each cell. **b** Summary statistics (mean, quantiles, and standard deviation (std)) for participant-level clustering and annotation accuracies. **c** Cell-level annotation variability, shown as boxplots; variability is defined as the number of unique participant-assigned labels per cell. **d** Cell-level annotation accuracy based on majority-vote consensus across participants, computed using the shortest path length between the majority-vote label and the reference label in the predefined hierarchy.

a



b

	Levels	Test name	Clustering accuracy		Annotation accuracy	
			p-value BM complete	p-value BM partial	p-value BM complete	p-value BM partial
Publication	No/Yes	Mood	0.55	0.02	0.08	0.35
GradCourse	No/Yes	Mood	0.52	0.84	0.08	0.84
Stats experience	No/Yes	Mood	0.55	0.16	0.55	0.55
Genetics experience	No/Yes	Mood	0.35	0.35	0.08	0.35
Single-cell experience	No/Yes	Mood	0.45	0.45	0.01	0.01
Study area	Biological sciences/Others	Mood	0.18	0.79	0.04	0.50
Current degree	Master/PhD	Mood	0.58	0.62	0.24	0.16
nPublications	0, 1-2, 3+	JT	0.13	0.01	0.14	0.28
nGradCourses	0, 1-2, 3-5, 6+	JT	0.19	0.56	0.3	0.79
Single-cell familarity level	0, 1, 2, 3, 4, 5	JT	0.11	0.22	0.22	0.12
Confidence level (Result)	0, 1, 2, 3, 4, 5	JT	0.23	0.41	0.27	0.11
Confidence level (SingleCell)	0, 1, 2	JT	0.07	0.09	0.05	0.20
Confidence level (Seurat)	0, 1, 2	JT	0.82	0.52	0.46	0.58

Figure 2.5: Annotation accuracy vs. questionnaire responses. a Boxplots for highlighted participant educational and research background showing significant association ($\alpha = 0.1$) with annotation accuracy, especially in the BM (bone marrow) complete dataset. **b** Summary table reporting the statistical tests used and corresponding p-values for annotation accuracy across all questionnaire response variables. Significant p-values are highlighted with underlines. Mood stands for the non-parametric Mood's median test. JT stands for the Jonckheere-Terpstra test, which assesses monotonic trends across ranked levels.

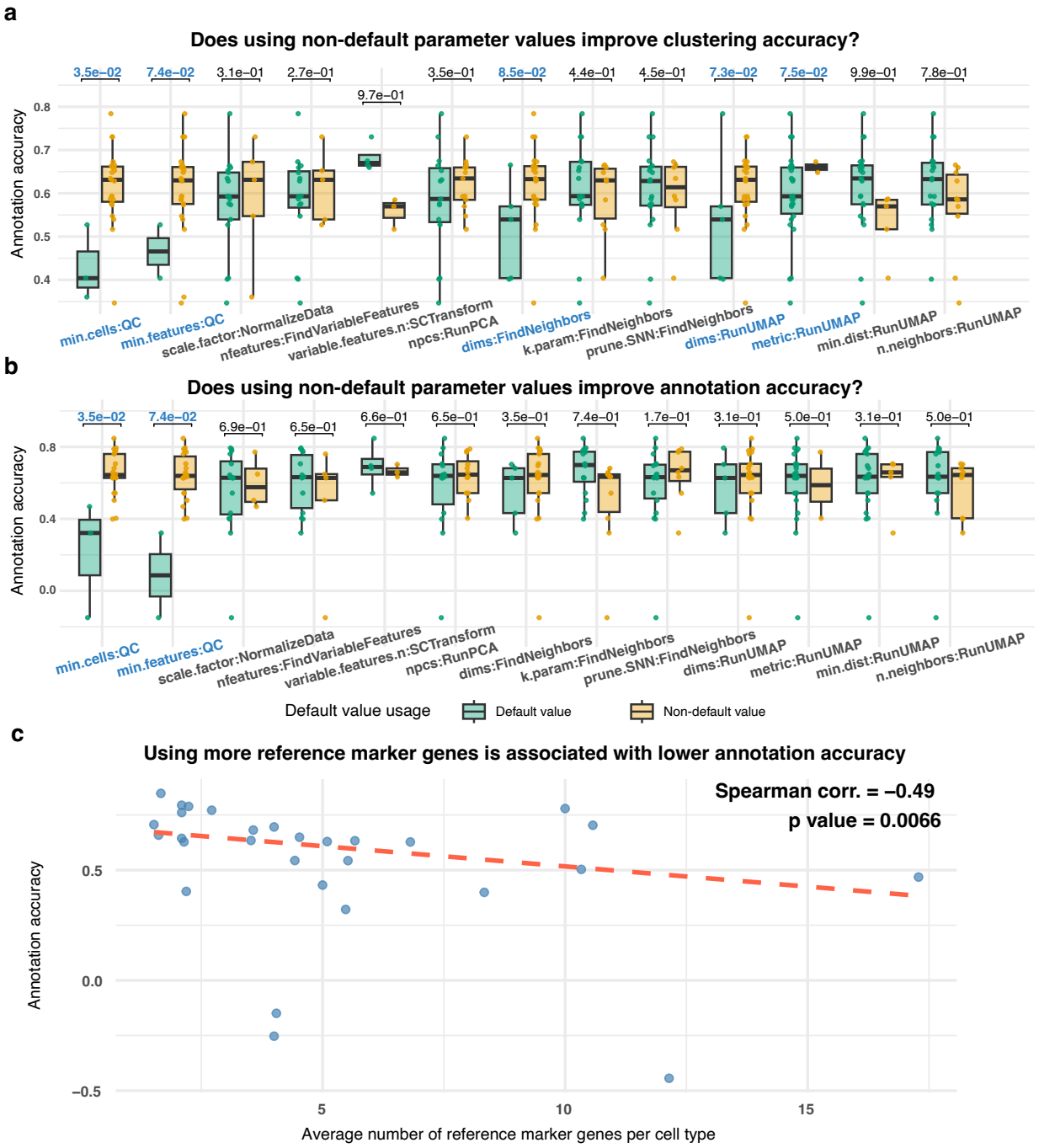


Figure 2.6: Parameter choices and reference marker gene selection influence annotation accuracy on the bone marrow complete dataset. **a** Boxplots comparing clustering accuracy based on whether the default parameter value was used. Only parameters for which at least two participants used each setting (default or non-default) are included. P-values are calculated using the one-sided Mood's median test, and significant values (p-value $\leqslant 0.1$) are highlighted in blue. QC stands for data-quality control. **b** Boxplots comparing annotation accuracy based on whether the default parameter value was used. Only parameters for which at least two participants used each setting (default or non-default) are included. P-values are calculated using the one-sided Mood's median test, and significant values (p-value $\leqslant 0.1$) are highlighted in blue. **c** Relationship between the average number of reference marker genes used per cell type and per-participant annotation accuracy. Each dot represents a participant; the average number of reference marker genes per cell type is the total number of reference markers a participant selected from literature or databases, averaged across cell types. A negative association indicates that using more reference markers can make the cell-type assignment step more challenging and increase subjectivity.

2.8 Supplementary materials

S2.8.1 Supplementary figures

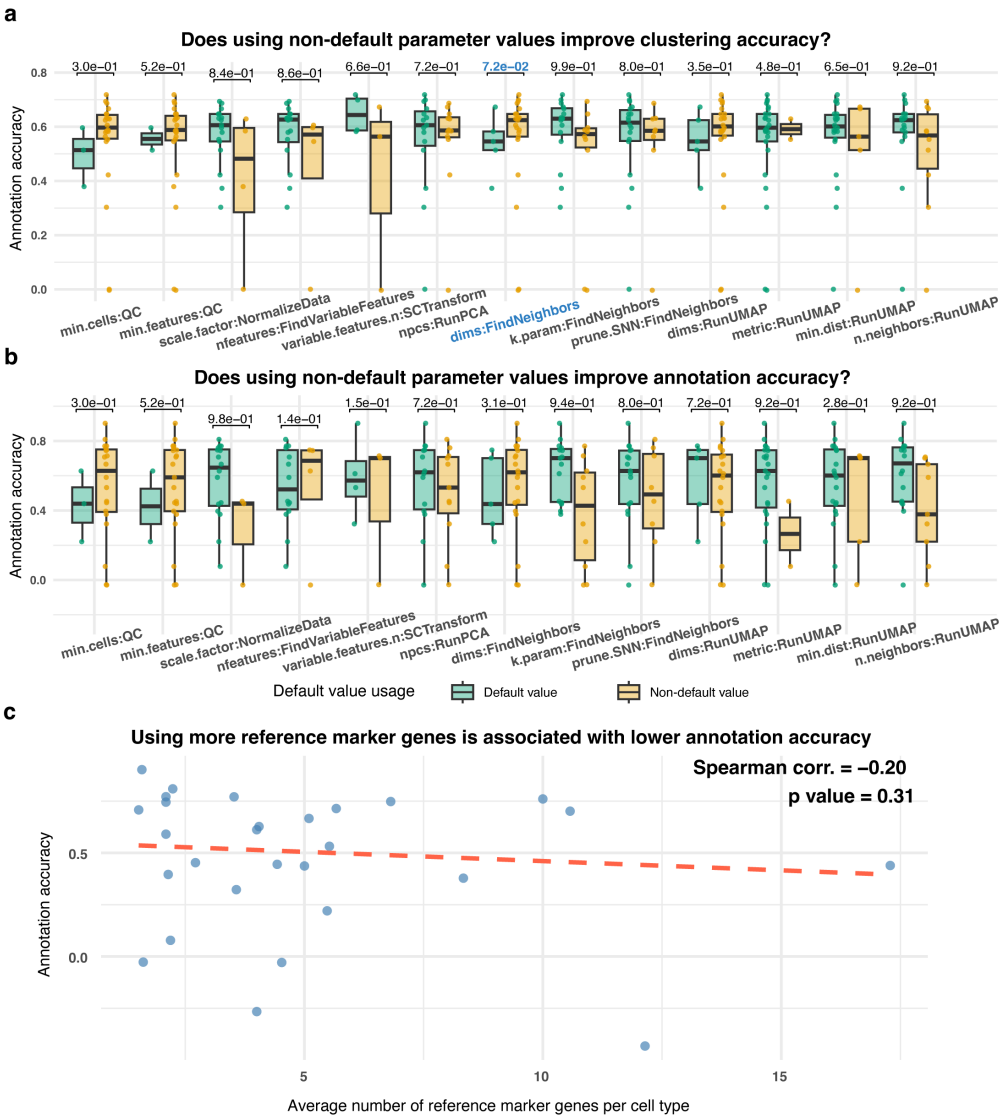


Figure 2.7: Parameter choices and reference marker gene selection influence annotation accuracy on the bone marrow partial dataset. a Boxplots comparing clustering accuracy based on whether the default parameter value was used. Only parameters for which at least two participants used each setting (default or non-default) are included. P-values are calculated using the one-sided Mood's median test, and significant values (p-value $\leqslant 0.1$) are highlighted in blue. QC stands for data-quality control. **b** Boxplots comparing annotation accuracy based on whether the default parameter value was used. Only parameters for which at least two participants used each setting (default or non-default) are included. P-values are calculated using the one-sided Mood's median test. **c** Relationship between the average number of reference marker genes used per cell type and per-participant annotation accuracy. Each dot represents a participant; the average number of reference marker genes per cell type is the total number of reference markers a participant selected from literature or databases, averaged across cell types. A negative association indicates that using more reference markers can make the cell-type assignment step more challenging and increase subjectivity.

S2.8.2 Supplementary tables

Question
1. What is your highest degree?
2. What area did you receive your highest degree?
3. What degree are you currently pursuing?
4. What is your current area of focus?
5. How many peer-reviewed publications have you co-authored in any area?
6. In your publication(s), did you perform any statistical analysis?
7. In your publication(s), did you perform any genetic or genomic data analysis?
8. In your publication(s), did you perform any single-cell data analysis?
9. How many undergraduate-level courses in statistics/ quantitative data analysis have you taken?
10. How many graduate-level courses in statistics/ quantitative data analysis have you taken?
11. From 0 (low) to 5 (high), how familiar are you with single-cell data analysis before taking the course?
12. From 0 to 5, how familiar are you with single-cell data analysis after taking the course?
13. From 0 to 5, how much do you believe that single-cell RNA-seq can capture previously known cell types in a biological sample?
14. From 0 to 5, how much do you believe in the reliability of Seurat, the single-cell analysis pipeline?
15. From 0 to 5, how accurate do you believe your final project results are?

Table S2.4: Questionnaire for participants.

S2.8.3 Instructions of the cell-type annotation tasks for participants

Final Project

For this project, you are required to use Seurat version 5.0.3 in R. Please submit 1) your code, 2) Seurat objects that store the annotated cell type labels [see below for details], and 3) the Excel file with your answers to each question.

Note that you should make sure your submitted code can run without errors and reproduce the same results you submitted. Please use `set.seed(2024)` at the beginning of your code in R.

Part 1

In this part, you are given two UMI count matrices with genes as rows and cells as columns [first one: BoneMarrow_dataset1.rds; second one: BoneMarrow_dataset2.rds]. Both these datasets come from the human bone marrow tissue from a healthy donor, which is sequenced by 10X 5' technology. Each dataset may contain some or all cell types in the following list (21 cell types in total):

- Classical monocyte cell
- Non-classical monocyte cell
- Plasmacytoid dendritic cell
- Dendritic cell
- Precursor B cell
- small pre-B-II cell
- Memory B cell
- Naive B cell
- CD16 natural killer cell

- Natural killer cell
- Plasma cell
- Macrophage cell
- Megakaryocyte cell
- Myeloid progenitor
- Erythroid progenitor
- Hematopoietic stem cell
- Lymphocyte cell
- CD8 T cell
- CD4 T cell
- Regulatory T cell
- Mucosal-associated invariant T cell

However, it is unknown how many and which cell types each of the two datasets contain. You need to determine which cell types exist at your discretion. Please use the Seurat pipeline and search for cell type marker genes to manually annotate the cell types for each dataset separately. You may search online for databases or relevant literature for cell type marker genes if needed.

Additionally, please answer the following questions and fill out the Excel file with your answers.

1. (1) What functions in Seurat did you use to annotate cell types? Use Y for Yes, and N for No to indicate this in the Excel sheets Part1-Dataset1-Q1 and Part1-Dataset2-Q1. (2) Please record the parameter values you tried for each function in the Excel sheets Part1-Dataset1-Q1 and Part1-Dataset2-Q1 as well. Note that the Excel sheet lists the parameters

you are allowed to change for each function. You could try any legit values for most parameters, but for a few parameters, you can only try certain values. If you only used the default values without tuning or you didn't use the function, fill out NA.

2. What are the parameter values for each function you decided to use for your final cell type annotations? Fill out them in the sheets Part1-Dataset1-Q2 and Part1-Dataset2-Q2. [See hint 1: you should choose one way for normalization. (If you choose sctransform, then you should not use ScaleData() and FindVariableFeatures().)]

3. What cell type marker genes did you use in your attempt to identify each cell type? List the marker genes in the sheets Part1-Q3. Please use Ensembl IDs as marker gene names [for example, ENSG00000203747].

4. What are your final annotation results? Please add your annotated cell type label for each cell to the submitted Seurat objects as a part of the `seurat.obj@meta.data`, named `cell_type`. And please use exactly the same cell type name listed above as your cell type labels.

Part 2

In the second part, you are given one UMI count matrix with genes as rows and cells as columns [Pancreas.rds]. The dataset comes from the healthy donor's pancreas tissue. The human pancreas dataset usually contains pancreatic polypeptide (PP) and alpha cells, which we are interested in. Please use the Seurat pipeline and search for cell-type marker genes to manually annotate these two cell types if they exist. You may search online for databases or relevant literature for cell type marker genes if needed. Additionally, please answer the following questions and fill out the Excel file with your answers.

1. 1) What functions in Seurat did you use to annotate cell types? Use Y for Yes, and N for No to indicate this in the Excel sheet Part2-Q1. 2) Please record the parameter values you have tried for each function in the Excel sheet Part2-Q1 as well. Note that the Excel sheet lists the parameters you are allowed to change for each function. You could try any legit values for most parameters, but for a few parameters, you could only try certain values.

If you only used the default values without tuning or you did not use the function, fill out NA.

2. What are the parameter values for each function you decided to use for your final cell type annotations? Fill them out in the sheets Part2-Q2. [See hint 1: you should choose one way for normalization. (If you choose sctransform, then you should not use ScaleData() and FindVariableFeatures().)]

3. What cell type marker genes did you use in your attempt to identify PP and alpha cells? List the marker genes in the sheet Part2-Q3. Please use the Ensembl IDs as the gene names.

4. What are your final annotation results? Please make sure that you add your annotated cell type label to each cell in the submitted Seurat objects as a part of the seurat.obj@meta.data, named 'cell_type'. For PP cells and alpha cells you find, please use 'PP' and 'alpha' as their cell type labels. For cells that you think are neither PP nor alpha cells, use NA as their cell type labels.

5. If PP cell or/and alpha cell exists, can you find new marker genes besides what you used to annotate the cell types? Fill your answers in the sheet Part2-Q5. If you did not find PP/alpha cells, or you don't think there are any additional cell type marker genes, please fill out NA in the sheet.

S2.8.4 Prompts of the cell-type annotation tasks for Biomni

For this project, you are required to complete Part 1 and Part 2 to analyze the provided datasets. Please submit 1) your code, 2) the full annotated cell type labels [see below for details], and 3) the Answer Sheet with your answers to each question.

Note that you should make sure your submitted code can run without errors and reproduce the same results you submitted.

Part 1

In this part, you are given three UMI count matrices with genes as rows and cells as columns [first one: BoneMarrow_dataset1.h5ad; second one: BoneMarrow_dataset2.h5ad; third one: BoneMarrow_dataset3.h5ad]. All these datasets come from the human bone marrow tissue from a healthy donor, which is sequenced by 10X 5' technology. Each dataset may contain some or all cell types in the following list (21 cell types in total):

- Classical monocyte cell
- Non-classical monocyte cell
- Plasmacytoid dendritic cell
- Dendritic cell
- Precursor B cell
- small pre-B-II cell
- Memory B cell
- Naive B cell
- CD16 natural killer cell
- Natural killer cell
- Plasma cell
- Macrophage cell
- Megakaryocyte cell
- Myeloid progenitor
- Erythroid progenitor
- Hematopoietic stem cell

- Lymphocyte cell
- CD8 T cell
- CD4 T cell
- Regulatory T cell
- Mucosal-associated invariant T cell

However, it is unknown how many and which cell types each of the two datasets contain. You need to determine which cell types exist at your discretion. Please search for cell type marker genes to manually annotate the cell types for each dataset separately. You may search online for databases or relevant literature for cell type marker genes if needed.

Additionally, please answer the following questions for each dataset.

1. 1) What functions did you use to annotate cell types? 2) Please record the parameter values you tried for each function.

2. What are the parameter values for each function you decided to use for your final cell type annotations?

3. What cell type marker genes did you use in your attempt to identify each cell type? List the marker genes. Please use Ensembl IDs as marker gene names [for example, ENSG00000203747].

4. What are your final annotation results? Please use exactly the same cell type name listed above as your cell type labels.

Part 2

In the second part, you are given two UMI count matrices with genes as rows and cells as columns [Pancreas_dataset1.h5ad and Pancreas_dataset2.h5ad]. The datasets come from the healthy donors' pancreas tissue. The human pancreas dataset usually contains pancreatic polypeptide (PP) and alpha cells, which we are interested in. Please search for cell type

marker genes to manually annotate these two cell types if they exist. You may search online for databases or relevant literature for cell type marker genes if needed.

Additionally, please answer the following questions for each dataset.

1. 1) What functions did you use to annotate cell types? 2) Please record the parameter values you have tried for each function. You could try any legit values for most parameters, but for a few parameters, you could only try certain values. If you only used the default values without tuning or you did not use the function, fill out NA.

2. What are the parameter values for each function you decided to use for your final cell type annotations?

3. What cell type marker genes did you use in your attempt to identify PP and alpha cells? List the marker genes. Please use the Ensembl IDs as the gene names.

4. What are your final cell type annotation results? For PP cells and alpha cells you find, please use 'PP' and 'alpha' as their cell type labels. For cells that you think are neither PP nor alpha cells, use NA as their cell type labels.

5. If PP cell or/and alpha cell exists, can you find new marker genes besides what you used to annotate the cell types? If you did not find PP/alpha cells, or you don't think there are any additional cell type marker genes, please fill out NA in the sheet.

S2.8.5 Issues we faced when using Biomni

During our evaluation using the official Biomni web application, we encountered several issues that might affect the user experience.

The first issue concerned software environment configuration. Initially, we required the use of the R programming language and a specific version of Seurat (v5.0.3). However, the agent failed to correctly install the specified version, resulting in a recursive installation failure loop. Consequently, we removed the constraint on the analysis environment, allowing the agent to use any programming language and software package version. Given that Python is the predominant language for most AI agents, Biomni defaulted to using Scanpy

for subsequent analyses.

The second issue is the inconsistency when downloading results from the Biomni web application. Upon downloading the final analysis results (comprising code and output files), we observed discrepancies between the workflow displayed on the web interface and the contents of the downloaded archive. Specifically, in the text-based analysis sections, the "User Prompts" recorded in the downloaded Jupyter notebooks did not match our current prompts; instead, they reflected prompts from our prior experimental iterations, even though the web interface displayed the correct version. This discrepancy in the downloaded logs posed a challenge. Nonetheless, we believe this behavior reflects the actual user experience, and we confirmed that the analysis in the notebooks did not rely on any Seurat or R implementation. Therefore, we still used the downloaded annotation results as the final Biomni output.

While a local deployment of Biomni might resolve these discrepancies, we were unable to pursue this option due to computational resources and time constraints. We remain grateful to the original authors of Biomni for generously providing the free web-based execution environment and LLM API support.

CHAPTER 3

IntegrateRigor: annotation-free integration optimization for cell identity recovery reveals cancer–immune interface niches

3.1 Introduction

Large-scale single-cell RNA sequencing (scRNA-seq) and spatial transcriptomics (ST) are transforming biology by enabling systematic measurements of gene expression at single-cell and spatial resolution [2–4]. The increasing availability of large reference atlases [76–78], together with the rapid accumulation of datasets generated across diverse experimental and biological sources, has greatly expanded opportunities to study cellular heterogeneity, tissue organization, and disease processes at scale. Realizing this potential, however, increasingly requires integrating data across multiple batches to recover comparable cell identities, a step that remains technically and conceptually challenging.

Here, a *batch* refers broadly to a sample or group of cells collected as a replicate or under a distinct condition, such as from different donors, time points, laboratories, sequencing platforms, or even species [29]. Throughout this manuscript, *cell identity* denotes the transcriptomic characteristics of a cell, including its type, subtype, or state [79]. Between-batch differences can arise from two distinct sources: differences in cell identity composition across batches, and differences in measured gene expression for the same cell identity across batches. The former does not hinder cell identity identification, as cells with the same identity are still comparable; the latter, which we term *between-batch variation*, obscures true cell identity and is the primary challenge for data integration. Importantly, between-batch variation is

not limited to technical artifacts but can also reflect biological differences between batches, such as cell-type-specific differential expression. Simply pooling cells across batches before identifying cell identities is not a valid option, as it fails to account for between-batch variation and thereby compromises cell identity recovery, as well as all downstream analyses that depend on it, such as differential expression testing, trajectory and lineage inference, and cell–cell communication analysis [57, 80].

Integration is therefore essential for constructing a shared space in which cell identities can be consistently recovered and compared across batches [22, 23]. Without proper integration, cells with the same identity but from different batches may remain separated due to between-batch variation [24, 25]. Crucially, cell identity recovery is a prerequisite for any cell-identity–based downstream analysis; these analyses should be performed only after reliable cell identities have been established across batches, not entangled with the integration step itself. Effective integration must therefore balance two competing objectives: preserving cell identity structure while minimizing between-batch variation. Failure to achieve this balance leads to two opposite failure modes. *Under-integration* leaves residual between-batch variation, causing cells to group by batch rather than by true identity. *Over-integration*, in contrast, removes biologically meaningful between-cell-identity variation and artificially merges distinct cell identities [26–29]. Both distort cell identity recovery and undermine the rigor and reproducibility of any downstream analyses. A wide range of computational methods has been developed for single-cell data integration, including Harmony [56], Seurat-RPCA [48], scVI [58], FastMNN [59], and LIGER [60], among many others [57, 81–83]. These methods are primarily designed to align cells with the same biological identity across batches into a shared low-dimensional embedding. Other approaches, such as Crescendo [57] and CellANOVA [84], take the integrated embedding as input to perform count-level batch correction and cell-type-specific differential expression analysis, respectively; their outputs therefore depend directly on the quality of the integrated embedding. However, despite substantial methodological progress, current integration practice remains limited in two key respects, especially in the common setting where accurate cell identity annotations are unavailable before integration [29].

First, the choice of input genes is a key but underappreciated determinant of integration quality for cell identity recovery [6, 29]. Most integration methods begin with generic gene selection, typically using highly variable genes (HVGs) selected either within each batch and then pooled, or from cells pooled across batches [6, 12]. However, these HVGs may be sensitive to between-batch variation; that is, their expression levels for the same cell identity may differ substantially across batches. For example, many HVGs primarily reflect activity or stress-response programs rather than intrinsic cell identity [51, 52], and are therefore highly sensitive to between-batch variation (for example, mitochondrial and ribosomal genes) [6, 53]. Using such HVGs introduces between-batch variation into the integration input, which hinders reliable cell identity alignment across batches. In contrast, genes tied to conserved cell-identity programs show relatively stable expression patterns across donors, tissues, and even species, providing a biologically grounded pool of candidate genes [85–88]. A recent benchmarking study [6] showed that gene selection can be as consequential as method choice for integration quality. Although a cell-type-annotation-dependent metric, group technical effects (GTE), has been proposed to quantify gene-wise between-batch variation [54], reliable annotations are often unavailable before integration, and using them for gene selection introduces circularity into the workflow.

Second, existing metrics for evaluating the trade-off between batch mixing and cell identity preservation fall into two categories, each with important limitations. First, annotation-dependent metrics—such as cell-type local inverse Simpson's index (cLISI) [24, 55] and adjusted Rand index (ARI) [55, 89] for cell identity preservation, and cell-type-aware integration LISI (CiLISI) [24] for batch mixing—are informative when accurate cell identity labels are available. However, they create a circular dependency: integration is performed precisely to recover cell identities across batches, yet these metrics presuppose that reliable annotations already exist. Second, annotation-free metrics—such as integration LISI (iLISI), k-nearest-neighbor batch-effect test (kBET), principal component regression (PCR), and batch average silhouette width (batch ASW) [24, 28, 55, 56]—avoid this dependency but measure only batch mixing without assessing cell identity preservation, and are therefore unable to detect over-integration. Although silhouette score is an annotation-free metric

that can assess cell cluster separation and thus cell identity preservation, it depends on the performance of the clustering algorithm and has been shown to be unreliable in single-cell integration benchmarking [55]. As a result, no existing annotation-free approach jointly assesses batch mixing and cell identity preservation to guide integration decisions for a specific dataset. In practice, researchers often resort to subjective visual inspection of UMAP plots [10] and trial-and-error hyperparameter tuning (e.g., the diversity clustering penalty in Harmony), making integration difficult to reproduce and to optimize for a given dataset.

Together, these limitations reveal a critical gap: no existing approach jointly addresses gene selection, hyperparameter optimization, and method selection in a statistically principled, annotation-free, dataset-specific manner with the explicit goal of reliable cell identity recovery. To address this gap, we propose IntegrateRigor, a data-driven, method-agnostic framework for optimizing single-cell and spatial transcriptomics integration for cell identity recovery. IntegrateRigor is not a standalone integration method; rather, it provides a principled wrapper for evaluating and tuning existing integration methods. It first identifies *batch-stable genes (BSGs)*—genes whose expression patterns across cell identities remain consistent across batches—and selects HVGs within this batch-stable set as cell-identity-informative input features for integration. Gene selection is achieved through a likelihood-based, annotation-free per-gene *batch stability score (BSS)*, which quantifies gene-wise stability across batches without requiring cell identity annotations. After integration, IntegrateRigor evaluates the resulting cell embedding using two annotation-free metrics: the *batch alignment score (BAS)*, which measures residual between-batch variation, and the *cell identity score (CIS)*, which measures preservation of cell identity structure. These two scores are aggregated into a dataset-level *integration score* that identifies the optimal configuration (i.e., method and hyperparameters) for reliable cell identity recovery on a given dataset. This shifts the focus from global method benchmarking to dataset-specific optimization: although benchmarking studies provide valuable overall rankings of integration methods [24, 25, 53], they typically use fixed parameter settings across datasets and therefore may not reflect each method's best achievable performance on any given dataset. By jointly addressing gene selection, hyperparameter tuning, and method selection, IntegrateRigor transforms

integration from a heuristic, trial-and-error workflow into a rigorous, annotation-free, and data-adaptive procedure for cell identity recovery.

In an application to colorectal cancer single-cell and spatial transcriptomics [57], IntegrateRigor enabled reliable cell identity recovery that revealed previously unannotated cancer–immune interface niches in the tumor microenvironment, structure that was obscured under both default integration settings and the over-integrated embedding reported in previous literature. Across six integration tasks spanning diverse sources of between-batch variation (i.e., sequencing platforms, sample conditions, developmental stage, species, and spatial tissue sections), IntegrateRigor consistently improved cell identity recovery of five widely used methods (Seurat-RPCA, Harmony, scVI, FastMNN, and LIGER). Notably, Harmony, which showed only moderate performance under default settings, achieved one of the best overall results when paired with IntegrateRigor, demonstrating that dataset-specific optimization can unlock the full potential of existing methods for cell identity recovery.

3.2 Results

3.2.1 Overview of IntegrateRigor: an annotation-free framework for batch-stable gene selection and integration optimization for cell identity recovery

IntegrateRigor provides an annotation-free, data-adaptive wrapper that optimizes data integration for reliable cell identity recovery while minimizing human subjectivity in hyperparameter tuning. By combining BSG selection with data-driven hyperparameter optimization, IntegrateRigor improves cell identity recovery by mitigating both under-integration and over-integration, without requiring prior cell identity annotations. This design makes IntegrateRigor broadly applicable across datasets and easy to use as a plug-in within existing integration pipelines. Details of the IntegrateRigor method are described in the **Methods Section**.

As illustrated in Figure 3.1, IntegrateRigor introduces annotation-free statistical metrics

at three levels: (1) a likelihood-based per-gene BSS that selects batch-stable, cell-identity-informative genes as input features; (2) likelihood-based embedding-level metrics, BAS and CIS, that jointly quantify batch mixing and cell identity preservation; and (3) a dataset-level integration score that identifies the optimal method and hyperparameter setting for a given dataset.

In Stage 1, IntegrateRigor identifies a set of BSGs to use as input features for downstream integration (Figure 3.1a). Here, between-batch variation refers to differences in measured gene expression for the same cell identity across batches, as distinct from differences in cell identity composition. We assume that a subset of genes carries cell identity information while showing minimal between-batch variation; that is, their expression within each cell identity does not shift substantially across batches (e.g., they are not cell-identity-specific differentially expressed genes across batches), and IntegrateRigor aims to identify and select these genes. Intuitively, such a gene should show the same relative expression patterns across cell identities in every batch, even if the proportions of cell identities differ between batches.

To formalize this intuition, we construct a likelihood-based statistic to quantify the degree to which a gene's expression distribution is preserved across batches. The idea is to leverage a *reference batch*—either user-defined (e.g., an atlas or a well-annotated anchor dataset) or, if not specified, automatically selected as the batch with the greatest cell-identity diversity, approximated by the largest number of cell clusters within a batch—to guide recovery of shared cell identity structure across batches. We refer to the remaining batches as *query batches*. Specifically, for each gene and each query batch, we fit two mixture models: a reference-constrained model that fixes the mixture components (corresponding to cell identities) as estimated in the reference batch while allowing only component proportions to differ, and an unconstrained model in which both components and proportions are free to vary from the reference batch. The difference in model fit between these two models, measured by the normalized log-likelihood, quantifies the degree of between-batch variation for that gene in that query batch; averaging this difference across all query batches yields the per-gene BSS, which captures how consistently a gene's expression pattern is maintained across batches (Figure 3.1b). Genes with higher scores are selected as BSGs using a data-adaptive thresh-

old determined by elbow-point detection, yielding input features that preferentially retain cell identity signals while reducing between-batch variation. HVGs are then selected within the BSGs as final input features, ensuring that they are both cell-identity-informative and stable across batches.

In Stage 2, IntegrateRigor identifies the optimal integration result among candidate embeddings generated under different hyperparameter settings (Figure 3.1a). Each embedding is evaluated using two complementary annotation-free metrics that correspond to the two objectives of integration for cell identity recovery. The BAS quantifies the degree of between-batch alignment in each embedding dimension, where higher values reflect better batch mixing. The CIS measures how strongly each embedding dimension preserves biologically meaningful cell identity structure, quantified by the degree to which the distribution of embedding values in that dimension reflects distinct cell subpopulations rather than a unimodal background. Specifically, for each embedding dimension, we compute the log-likelihood difference between a constrained and an unconstrained model separately for the reference batch and each query batch, then average across batches, with each per-batch score normalized by cell number to ensure comparability. These per-dimension scores are aggregated across all embedding dimensions to yield the BAS and CIS (Figure 3.1b–c). The dataset-level integration score is defined as the sum of BAS and CIS, directly capturing the trade-off between batch mixing and cell identity preservation: BAS penalizes under-integration, whereas CIS penalizes over-integration. Because both scores are defined as normalized log-likelihood differences, they are on a comparable scale and can be summed directly, without requiring an additional weighting parameter. The hyperparameter setting that maximizes the integration score is then selected as the optimal configuration for cell identity recovery on the given dataset.

Empirical validation of the statistical models underlying IntegrateRigor is in the **Methods Section**.

3.2.2 Batch stability scores identify batch-stable genes and improve cell identity recovery by excluding genes confounded by sample and technical variation

The BSS, computed in Stage 1 of IntegrateRigor, quantifies how stably a gene's expression pattern is maintained across batches within the same cell identity. Genes with high BSS show consistent expression patterns across batches within the same cell identity; genes with low BSS show substantial cross-batch shifts, indicating that their expression is strongly influenced by between-batch variation and may obscure shared cell identity structure, thereby hindering reliable cell identity recovery. We validated the BSS and evaluated the downstream benefit of BSS-based gene selection using the intestinal failure-associated liver disease (IFALD) dataset comprising three donors [90].

Genes ranked highly by the BSS showed more concordant expression patterns across batches, whereas low-ranked genes displayed substantial between-batch variation (Figure 3.2a). Principal component analysis of the pooled cells confirmed this separation: genes with large loadings on the batch-dominated principal component 1 (PC1) tended to have lower BSS and were not classified as batch-stable, whereas genes with large loadings on the cell-type-dominated PC2 tended to have higher BSS and were classified as batch-stable (Figure 3.8). To benchmark the BSS against an existing annotation-dependent metric, we compared the BSS with the GTE score [54], which requires cell-type annotations. BSS was linearly correlated with GTE on the log scale across all datasets (Pearson $r = 0.50$–0.69; all $p < 2.2\times10^{-16}$; Figure 3.9), confirming that BSS captures a similar notion of between-batch variation as this annotation-dependent metric, while requiring no cell-type labels.

IntegrateRigor ranks genes by BSS and uses a data-adaptive elbow-point threshold [91] to exclude genes that are clearly not batch-stable, retaining the remainder as BSGs for downstream HVG selection. This approach is deliberately conservative: rather than aggressively selecting only the most stable genes, it removes the clearly problematic ones while preserving a broad pool for HVG selection. On the IFALD dataset, only 331 of 14,349 genes were excluded as batch-unstable, leaving 14,018 BSGs. In both the default and BSG-restricted settings, the same number of 2,000 HVGs were used as input features; the only difference

was whether those HVGs were drawn from the full gene set or restricted to BSGs. When Harmony was applied to the default 2,000 HVGs, the integrated embedding showed clear under-integration, with cells of the same type remaining separated by batch. Restricting HVG selection to BSGs substantially improved cell identity recovery, increasing CiLISI from 1.33 to 1.47 and ARI from 0.77 to 0.89 (Figure 3.2b). Consistent improvements were observed across all five integration methods evaluated (FastMNN, Harmony, LIGER, scVI, and Seurat-RPCA), demonstrating that BSG selection is a modular, method-agnostic improvement that can be plugged into existing pipelines.

To understand why batch-unstable genes hinder cell identity recovery, we examined their biological and technical content. The three IFALD donors differ in age, tissue preservation method, and disease condition (Figure 3.2c). In both the unintegrated PCA and default Harmony UMAP, donor C54 (a healthy adult, frozen) was clearly separated from the other two donors, suggesting that the dominant between-batch variation reflects sample-level characteristics rather than disease status per se (Figure 3.2b; Figure 3.8). Consistent with this, batch-unstable genes were enriched for gene ontology terms related to coagulation and fibrinolysis, acute-phase and inflammatory responses, and lipid and metabolic processes (Figure 3.2e)—processes that are likely associated with age and preservation rather than shared cell identities [92–94]. Batch-unstable genes were also enriched for mitochondrial and ribosomal genes, which are known to reflect sample quality, stress, and global transcriptional activity rather than cell identity; conversely, these genes were underrepresented among BSGs (Table S3.2). Notably, not all ribosomal genes were uniformly unstable: a subset with specialized regulatory functions were consistently classified as batch-stable across datasets, consistent with prior gene-wise analyses [54] (Supplementary Material; Supplementary Figure 3.10).

Together, these results demonstrate that BSG selection improves cell identity recovery by filtering out genes dominated by sample-specific effects or technical noise, while retaining genes that encode shared cell identity structure across batches.

3.2.3 Integration scores guide hyperparameter optimization to balance under-integration and over-integration for cell identity recovery

State-of-the-art integration methods (e.g., Seurat-RPCA, Harmony, scVI, and FastMNN) produce a low-dimensional cell embedding as their integration output, and IntegrateRigor evaluates each dimension of this embedding using two complementary annotation-free metrics: BAS, which quantifies the degree of between-batch alignment, and CIS, which measures how strongly the dimension preserves biologically distinct cell identity structure. To calculate BAS and CIS, IntegrateRigor uses a Gaussian mixture model to evaluate the distribution of values in each embedding dimension. However, LIGER embeddings were not well captured by this model (see **Methods Section**), so LIGER was excluded from hyperparameter tuning and method selection analyses.

In the IFN-β-stimulated PBMC dataset [95] (13,999 cells, two batches), BAS and CIS were directly interpretable at the dimension level on the Harmony embedding: dimensions with high BAS showed strong batch mixing, whereas those with high CIS clearly resolved major immune cell types (Figure 3.2f). This confirms that the two scores capture complementary aspects of integration quality and can support exploratory analysis of individual dimensions of the integrated embedding.

Building on this, IntegrateRigor averages the dimension-wise BAS and CIS across all embedding dimensions and defines the integration score as their sum. On the IFN-β-stimulated PBMC dataset, default Harmony showed clear under-integration, with cells remaining partially separated by batch, compromising cell identity recovery (Figure 3.2g). In contrast, the hyperparameter value selected by the integration score, even without BSG selection, improved batch mixing (CiLISI increased from 1.54 to 1.65) while substantially improving cell identity recovery (ARI increased from 0.73 to 0.94) (Figure 3.2g). Crucially, this optimization required no cell identity annotations and relied entirely on the intrinsic structure of the integrated embedding.

A key advantage of jointly scoring batch mixing and cell identity preservation is the ability to guard against over-integration, a failure mode that annotation-free batch-mixing metrics

alone cannot detect. Existing annotation-free metrics such as iLISI, PCR, batch ASW, and kBET reward stronger batch mixing but do not assess whether cell identity structure is preserved. To illustrate this limitation, we analyzed a human immune cell dataset with two batches (one derived from bone marrow tissue and the other from PBMC) [24]. We then evaluated Harmony embeddings across a range of hyperparameter values. The four batch-mixing metrics showed similar trends across hyperparameters and, although kBET selected a slightly different hyperparameter value than iLISI, PCR, and batch ASW, all of these settings favored stronger batch mixing and diverged from the hyperparameter at which the IntegrateRigor integration score peaked. Specifically, along the hyperparameter path, BAS increased with the hyperparameter value, whereas CIS decreased (Figure 3.14), demonstrating the tradeoff between batch mixing and cell identity preservation—a consideration absent from existing batch-mixing metrics (Figure 3.14). At the hyperparameter selected by maximizing the integration score, cells from different batches were well aligned while major immune populations remained clearly separated (CiLISI = 1.43, cLISI = 1.16, ARI = 0.65). In contrast, the hyperparameter selected by iLISI, PCR, ASW, and kBET achieved a similar batch mixing (CiLISI = 1.40) but exhibited over-integration: several distinct cell identities became blended (cLISI increased to 1.26 and ARI dropped to 0.61) (Figure 3.2h). Here, a higher cLISI value indicates worse cell identity preservation.

Together, these results demonstrate that the integration score identifies hyperparameter settings that achieve a principled balance between batch mixing and cell identity preservation for reliable cell identity recovery, mitigating both under-integration and over-integration without requiring prior cell identity annotations.

3.2.4 IntegrateRigor consistently improves cell identity recovery across diverse datasets and methods

To assess the generalizability of IntegrateRigor, we evaluated it across a diverse collection of integration tasks spanning multiple sources of between-batch variation: sample conditions (IFN-β-stimulated versus control PBMCs [95]; IFALD donors with distinct age, preservation,

and disease context [90]), sequencing platforms (human immune cells [24]), developmental stages (chicken heart [96]), tissue slices (mouse brain spatial transcriptomics [97]) and species (human–mouse pancreas [98]). This diversity allows us to test whether IntegrateRigor can improve recovery of comparable cell identities across batches in settings that go well beyond routine single-cell batch correction, including cases where technical and biological differences make integration especially challenging.

Across all these tasks, IntegrateRigor consistently improved cell identity recovery relative to the default pipeline, as evidenced by better batch mixing (CiLISI) together with maintained or improved cell identity preservation (ARI) (Figure 3.3a). In the IFN-β-stimulated PBMC dataset, applying IntegrateRigor with scVI improved batch mixing (CiLISI increased from 1.69 to 1.78) while maintaining high cell identity recovery (ARI from 0.82 to 0.83), reflecting better mixing of stimulated and control cells without loss of cell-type structure. In IFALD, where default Harmony left cells of the same type partially separated by donor, IntegrateRigor substantially improved both batch mixing and cell identity recovery (CiLISI: 1.36 $\rightarrow$ 1.47; ARI: 0.77 $\rightarrow$ 0.88). In the chicken heart dataset, FastMNN with IntegrateRigor improved from CiLISI = 2.49 to 2.63 and ARI from 0.62 to 0.63. In the human–mouse pancreas dataset, Seurat-RPCA batch mixing increased from CiLISI = 1.88 to 2.07 while ARI remained high at 0.95. Complete metric values and UMAP visualizations for all datasets are provided in Supplementary Figures 3.19–3.25. Together, these results demonstrate that IntegrateRigor improves cell identity recovery across a wide range of dataset scenarios, from cases where the default embedding is already reasonable to examples with pronounced under-integration.

The gains are summarized in Figure 3.3b, which shows the relative improvement of IntegrateRigor over default integration across methods and datasets. Batch mixing (CiLISI) improved in nearly all settings, with the largest gains in datasets with strong between-batch variation such as IFN-β-stimulated PBMC, IFALD, and human–mouse pancreas. The strong improvement on the human–mouse pancreas dataset shows that IntegrateRigor can also optimize cross-species integration, where the differences between batches are not limited to technical noise but include species-level transcriptomic differences. This result supports the

broader applicability of IntegrateRigor: by selecting batch-stable genes and tuning integration strength, the framework can recover comparable cell identities even when datasets are separated by large biological and technical gaps. Crucially, these improvements in batch alignment were accompanied by maintained or improved ARI, confirming that increased mixing did not come at the cost of losing cell identity structure (Figure 3.3c). Notably, Harmony showed the largest improvement and became the best-performing method overall when optimized by IntegrateRigor, despite its moderate default performance, demonstrating how dataset-specific optimization can enable better cell identity recovery.

We further examined whether the integration score, in addition to guiding hyperparameter selection within each method, can help distinguish among integration methods. This is inherently difficult because integration has no absolute gold standard and different metrics capture different aspects of performance [55, 99]. We therefore do not claim that IntegrateRigor selects a uniquely optimal method, but instead ask whether its ranking agrees with established annotation-dependent metrics of batch alignment and cell identity preservation.

In five of the six datasets, Seurat-RPCA and Harmony were the two methods with the highest integration scores (Figure 3.26). The exception was the chicken heart dataset, for which Seurat-RPCA and FastMNN ranked highest, with FastMNN achieving the top integration score (Figure 3.26). This pattern is broadly consistent with the overall ranking in Figure 3.3c, where Seurat-RPCA and Harmony cluster in the upper-right region (high CiLISI and high ARI), and with the strong cell-type separation observed for FastMNN in the chicken heart dataset (Figure 3.22). These observations support the integration score as a useful annotation-free proxy for method-level cell identity recovery and batch alignment.

3.2.5 IntegrateRigor uncovers cancer–immune interface niches in colorectal cancer by integrating single-cell and spatial transcriptomics data

Principled integration for reliable cell identity recovery can enable biological discoveries that remain hidden under suboptimal hyperparameter choices. We illustrate this in a colorectal cancer dataset profiled with both single-cell and spatial transcriptomics [57] (192,166

cells; two MERSCOPE spatial tissue slices and one 10x Genomics single-cell batch). The original study [57] used Harmony as an upstream step to obtain integrated embedding for Crescendo, a downstream count correction method, and provided cell-type annotations based on the Harmony embedding (Figure 3.4a). However, neither the code nor the integrated embedding used in that study was publicly released. We therefore reproduced the Harmony embedding under default settings and found it markedly different from the published embedding: the original embedding showed extensive batch mixing (Figure 3.4a), whereas our default Harmony embedding showed clear residual batch separation, particularly among epithelial cancer cells (Figure 3.27). This discrepancy highlights a common challenge: without a principled criterion, it is unclear which embedding better preserves biologically distinct cell identities.

Applying IntegrateRigor to this dataset yielded an optimized Harmony embedding that differed from both the original and the default result, achieving a better balance between batch mixing and cell identity preservation (Figure 3.4a). When we examined whether the cell-type annotations from the original study remained consistent with the optimized embedding, we found substantial discordance: a subset of cells originally annotated as epithelial cancer cells clustered among immune populations in the IntegrateRigor embedding. This motivated us to perform independent cell-type annotation on the optimized embedding using Louvain clustering and marker gene expression.

In our reannotation, cells that shifted from epithelial cancer cells to immune cells were mapped back to their physical locations in the spatial transcriptomics data (Figure 3.4b). These cells localized specifically to the boundary between canonical epithelial cancer cells (annotated as epithelial cancer in both the original and revised annotations) and canonical immune cells (annotated as immune in both annotations). We refer to these spatially localized boundary populations as *cancer–immune interface niches*.

We confirmed their identity as interface populations by differential expression analysis using observed gene expression from a single batch before integration, to avoid integration artifacts (Figure 3.4c). When compared with canonical epithelial cancer cells, each niche was enriched for markers of its corresponding immune cell type; when compared with canon-

ical immune cells, each niche retained strong epithelial marker expression (*KRT8*, *MUC13*, *PERP*). This dual signature—immune-like relative to epithelium, epithelial-like relative to immune—defines the interface niche identity. Specifically, the cancer–B niche expressed *CD19*, *MS4A1*, and *CXCR5*; the cancer–mast niche expressed *CPA3*, *MS4A2*, and *KIT*; the cancer–T niche expressed *CCL5*, *CD3D*, and *KLRB1*; the cancer–plasma niche expressed *MANF*, *CCR10*, and *MZB1*; and the cancer–myeloid niche expressed *CSF3R*, *FCGR3B*, and *IL1A*. Such mixed transcriptional profiles are expected in spatial transcriptomics data, where direct cell–cell contact, imperfect cell segmentation, and local transcript contamination can all contribute.

Functional enrichment analysis of genes differentially expressed between each niche and its corresponding canonical immune population (excluding shared epithelial markers) revealed immune cell-type-specific programs consistent with active roles at the cancer–immune interface (Figure 3.4d; Figure 3.28) [100–102]. The cancer–myeloid niche was enriched for "cytokine-mediated signaling pathway," "positive regulation of cytokine production," "leukocyte cell–cell adhesion," and "regulation of T cell activation," consistent with a program of immune communication and local coordination in the tumor microenvironment [103, 104]. The cancer–B niche was dominated by antigen-processing and antigen-presentation terms, consistent with a state specialized for antigen handling at the epithelial cancer interface [105–107]. The cancer–T niche was enriched for leukocyte-mediated cytotoxicity, interleukin-10 production, and cell–cell adhesion, suggesting an immunoregulatory T-cell state shaped by epithelial contact [104, 108]. The cancer–plasma niche was enriched for antigen-presentation and activated immune-response programs beyond canonical plasma-cell function [109]. The cancer–mast niche was enriched for immunological memory formation and antigen-presentation-related terms, suggesting involvement in broader adaptive immune regulation [110–112]. Across all niches, the enriched programs reflect the characteristic functions of the corresponding immune cell type, consistent with their positions at the cancer–immune interface.

These interface niches were not detectable under either the original or the default Harmony embedding. The original embedding merged them into a broad cluster, masking their

intermediate identity and leading to their annotation as epithelial cancer cells only. The default Harmony embedding failed to align them consistently across batches, preventing their recognition as specific types of niches. In contrast, the IntegrateRigor-optimized embedding revealed these cancer–immune interface niches and supported their interpretation through spatial localization, dual cancer–immune marker signatures, and immune cell-type-specific functional enrichment. This example demonstrates that the choice of integration hyperparameters is not merely a technical detail: by optimizing integration for cell identity recovery, IntegrateRigor can reveal cell populations of biological interest that would otherwise remain obscured.

3.3 Methods

We first consider a two-batch setting consisting of a reference batch and a query batch, and subsequently extend the framework to multiple batches. The reference batch serves as an anchor for shared cell identity structure and is assumed to contain a more diverse set of cell identities, while the query batch may contain only a subset of cell identities.

Let the gene expression count matrix of the reference batch be denoted by $\mathbf{X}^0 = [x_1^0, \ldots, x_p^0] \in \mathbb{N}^{n_0 \times p}$ and that of the query batch by $\mathbf{X}^1 = [x_1^1, \ldots, x_p^1] \in \mathbb{N}^{n_1 \times p}$, where n_0 and n_1 denote the numbers of cells in the reference and query batches, respectively, and p denotes the number of genes.

3.3.1 Batch-stable gene (BSG) selection for data integration

3.3.1.1 Statistical modeling for gene expression in reference and query batches

We focus on a single gene and suppress the gene index for notational simplicity. Let $\mathbf{x}^b = (x_1^b, \ldots, x_{n_b}^b)^\top$ denote the expression counts in batch $b \in \{0, 1\}$, with corresponding library sizes $(l_1^b, \ldots, l_{n_b}^b)^\top$. Here, $b = 0$ refers to the reference batch, and $b = 1$ refers to the query batch.

To identify genes that preserve cell identity across batches, we model gene expression us-

ing a negative binomial (NB) mixture model, with mixture components representing distinct cell identities. This model is motivated by extensive prior evidence that the NB distribution effectively models overdispersed gene-expression count data within a homogeneous cell identity [113]. Validation of this model assumption is provided in **Methods Subsection "Validation of the statistical models underlying IntegrateRigor."** Specifically, we assume that for any given gene

$$x_i^0 \sim \sum_{k=1}^{K_0} \pi_k^0 \,\mathrm{NB}\big(l_i^0 \mu_k^0, \phi_k^0\big)\,, \quad i = 1, \ldots, n_0, \tag{3.1}$$

and

$$x_i^1 \sim \sum_{k=1}^{K_1} \pi_k^1 \,\mathrm{NB}\big(l_i^1 \mu_k^1, \phi_k^1\big)\,, \quad i = 1, \ldots, n_1, \tag{3.2}$$

where $K_1 \leqslant K_0$. Here, $\pi_k^b > 0$ denotes the mixing proportion of component k in batch $b \in \{0, 1\}$, with $\sum_{k=1}^{K_b} \pi_k^b = 1$, and $\mu_k^b > 0$ and $\phi_k^b > 0$ denote the corresponding mean and dispersion parameters for that NB component, respectively. We assume that these component-specific parameters are distinct across mixture components; that is,

$$(\mu_k^b, \phi_k^b) \neq (\mu_{k'}^b, \phi_{k'}^b), \quad \text{for all } k \neq k',\ k, k' = 1, \ldots, K_b,\ b \in \{0, 1\}.$$

Intuitively, if a gene reflects cell identity rather than between-batch variation, its expression pattern across cell identities should be preserved across batches, even when the proportions of those cell identities differ. In this case, the gene expression distribution in the query batch is assumed to share a subset of the mixture components in the reference batch, while allowing only the mixing proportions to differ.

Specifically, in the absence of between-batch variation, the mixture components $\{\mu_k^1, \phi_k^1\}_{k=1}^{K_1}$ in the query batch model (3.2) is a subset of the mixture components $\{\mu_k^0, \phi_k^0\}_{k=1}^{K_0}$ in the reference batch model (3.1). Then the gene expression distribution of the query batch can be

re-parameterized using $\{\mu_k^0, \phi_k^0\}_{k=1}^{K_0}$, while allowing only the component proportions to vary:

$$x_i^1 \sim \sum_{k=1}^{K_0} \tilde{\pi}_k^1 \, \mathrm{NB}(\ell_i^1 \mu_k^0, \phi_k^0), \quad i = 1, \ldots, n_1, \tag{3.3}$$

which we refer to as the *reference-constrained model*. Here, $\tilde{\pi}_k^1$ denotes the mixing proportion of the reference component k in the reference-constrained model for the query batch. We distinguish it from π_k^1, which refers to the mixing proportion of the query component k in model (3.2), and the reference components and query components may not share the same indices.

In contrast, between-batch variation induces shifts in the component parameters, such that $\{\mu_k^1, \phi_k^1\}_{k=1}^{K_1} \notin \{\mu_k^0, \phi_k^0\}_{k=1}^{K_0}$. Consequently, the query distribution can no longer be adequately explained by the reference mixture structure alone and instead requires the *full model* (3.2). We therefore define the BSS for the gene based on how well the query batch can be explained by the reference-constrained model relative to the full model.

3.3.1.2 Batch stability score (BSS) definition

To distinguish these two scenarios, we compare two models, the reference-constrained model (3.3) and the full model (3.2), for the query batch, both fitted using the Expectation–Maximization (EM) algorithm [**dempster1977maximum**]. Under the reference-constrained model (3.3), the component parameters $\{\mu_k^0, \phi_k^0\}_{k=1}^{K_0}$ are first estimated from the reference batch via EM and then treated as fixed when evaluating the query batch, with only the mixing proportions $\{\tilde{\pi}_k^1\}_{k=1}^{K_0}$ re-estimated by EM. Under the full model (3.2), all parameters $\{\pi_k^1, \mu_k^1, \phi_k^1\}_{k=1}^{K_1}$ are estimated directly from the query batch using EM.

Let ℓ_0^{BSS} denote the log-likelihood of the reference-constrained model and ℓ_1^{BSS} of the log-likelihood of the full model, both evaluated based on the parameter estimates from the query batch. Then we define the batch stability score (BSS) as

$$\mathrm{BSS} = \frac{1}{n_1}(\ell_0^{\mathrm{BSS}} - \ell_1^{\mathrm{BSS}}) \in (-\infty, 0),$$

which takes a negative value as ℓ_0^{BSS} is strictly smaller than ℓ_1^{BSS} as the parameter space of the reference-constrained model is a strict subspace of that of the full model.

A high (less negative) BSS indicates that the reference-constrained model explains the query data well, suggesting that the gene's expression pattern transfers between batches and is therefore batch-stable. Conversely, a low (more negative) BSS indicates that the full model provides a substantially better fit, suggesting between-batch variation. BSGs are selected based on the empirical distribution of BSS values using a data-adaptive threshold determined by an elbow criterion [91].

3.3.2 Integration score for a given dataset and integration method

3.3.2.1 Statistical modeling for integrated embedding

Let $\mathbf{E}^0 = [e_1^0, \ldots, e_d^0] \in \mathbb{R}^{n_0 \times d}$ and $\mathbf{E}^1 = [e_1^1, \ldots, e_d^1] \in \mathbb{R}^{n_1 \times d}$ denote the low-dimensional embeddings of the reference and query batches from an integrated embedding produced by a given integration method, where $e_j^b = (e_{1j}^b, \ldots, e_{n_b j}^b)^\top$ denotes the jth embedding dimension for batch $b \in \{0, 1\}$, where $b = 0$ denotes the reference batch, and $b = 1$ denotes the query batch.

For each embedding dimension $j = 1, \ldots, d$, we assume that

$$e_{ij}^0 \sim \sum_{m=1}^{M_0} \pi_{mj}^0 \mathcal{N}\left(\mu_{mj}^0, (\sigma_{mj}^0)^2\right), \quad i = 1, \ldots, n_0,$$

and

$$e_{ij}^1 \sim \sum_{m=1}^{M_1} \pi_{mj}^1 \mathcal{N}\left(\mu_{mj}^1, (\sigma_{mj}^1)^2\right), \quad i = 1, \ldots, n_1, \tag{3.4}$$

where $M_1 \leqslant M_0$. Here, $\pi_{mj}^b > 0$ denotes the mixing proportion of component m in embedding dimension j for batch $b \in \{0, 1\}$, with $\sum_{m=1}^{M_b} \pi_{mj}^b = 1$, and μ_{mj}^b and $(\sigma_{mj}^b)^2$ denote the corresponding mean and variance parameters for that Gaussian component, respectively. Given our observation that the Gaussian mixture model reflects distinct cell identities in some dimensions of the integrated embedding (see **Methods Subsection "Validation of the**

statistical models underlying IntegrateRigor."), we assume that the mixture components in this model correspond to cell identities and that the component-specific parameters are distinct; that is,

$$\left(\mu^b_{mj}, (\sigma^b_{mj})^2\right) \neq \left(\mu^b_{m'j}, (\sigma^b_{m'j})^2\right), \text{ for all } m, m' \in \{1, \ldots, M_b\},\ m \neq m',\ j = 1, \ldots, d,\ b \in \{0, 1\}.$$

3.3.2.2 Batch alignment score (BAS) definition

The BAS quantifies batch mixing in the integrated embedding space. For each embedding dimension $j = 1, \ldots, d$, following the same rationale as the BSS definition, we compare a reference-constrained model,

$$e^1_{ij} \sim \sum_{m=1}^{M_0} \tilde{\pi}^1_{mj} \mathcal{N}\left(\mu^0_{mj}, (\sigma^0_{mj})^2\right), \quad i = 1, \ldots, n_1,$$

with the full model (3.4) for the query batch embeddings in the same dimension. Under the reference-constrained model, the component parameters $\{\mu^0_{mj}, (\sigma^0_{mj})^2\}_{m=1}^{M_0}$ are estimated from the reference batch and then treated as fixed when evaluating the query batch, with only the mixing proportions $\{\tilde{\pi}^1_{mj}\}_{m=1}^{M_0}$ re-estimated. Under the full model, all parameters $\{\pi^1_{mj}, \mu^1_{mj}, (\sigma^1_{mj})^2\}_{m=1}^{M_1}$ are estimated directly from the query batch.

Let ℓ^{BAS}_{0j} denote the log-likelihood of the reference-constrained model and ℓ^{BAS}_{1j} denote the log-likelihood of the full model, both evaluated on the query embeddings in dimension j. We define the dimension-wise BAS as

$$\text{BAS}_j = \frac{1}{n_1}\left(\ell^{\text{BAS}}_{0j} - \ell^{\text{BAS}}_{1j}\right) \in (-\infty, 0).$$

The overall BAS is obtained by averaging over all embedding dimensions:

$$\text{BAS} = \frac{1}{d}\sum_{j=1}^{d} \text{BAS}_j.$$

A higher (less negative) BAS indicates that the reference-constrained model explains the

query embeddings well, suggesting better alignment between batches and more effective removal of between-batch variation. Conversely, a lower (more negative) BAS indicates that the full model provides a substantially better fit, suggesting residual between-batch variation and potential under-integration.

3.3.2.3 Cell identity score (CIS) definition

The CIS quantifies the extent to which biologically distinct cell identities are preserved in the integrated embedding. Intuitively, an embedding dimension that captures cell identities should exhibit a multimodal distribution, whose modes (i.e., mixture components) correspond to distinct cell identities [114]. In contrast, if an embedding dimension exhibits a unimodal distribution, it does not reveal cell identities.

Thus, for each embedding dimension j, we compare a Gaussian mixture model with a single Gaussian model, both fitted within each batch. Let ℓ_{bj}^{mix} denote the log-likelihood of the Gaussian mixture model and ℓ_{bj}^{null} denote the log-likelihood of the single Gaussian model evaluated on batch $b = 0, 1$ and dimension j. We define the dimension-wise CIS as

$$\text{CIS}_j = \frac{1}{2}\left[\frac{1}{n_0}\left(\ell_{0j}^{\text{mix}} - \ell_{0j}^{\text{null}}\right) + \frac{1}{n_1}\left(\ell_{1j}^{\text{mix}} - \ell_{1j}^{\text{null}}\right)\right] \in (0, +\infty).$$

The overall CIS is obtained by averaging over all dimensions of the integrated embedding:

$$\text{CIS} = \frac{1}{d}\sum_{j=1}^{d} \text{CIS}_j.$$

A higher CIS indicates that the Gaussian mixture model explains the embedding values well across dimensions, suggesting stronger preservation of cell identity structure, whereas a low CIS suggests loss of cell identities and potential over-integration.

3.3.2.4 Integration score definition

The dataset-level integration score is defined as

$$\text{Integration score} = \text{BAS} + \text{CIS},$$

where BAS penalizes under-integration, and CIS penalizes over-integration. Hence, the integration score balances the trade-off between batch mixing and cell identity preservation, providing a principled, annotation-free criterion. Maximizing the integration score, IntegrateRigor optimizes the hyperparameter setting of a given integration method on a given dataset.

3.3.2.5 Hyperparameter for integration methods

On the six integration datasets we analyzed in the **Subsection "IntegrateRigor consistently improves integration across diverse datasets and methods"**, we evaluated the following hyperparameters and values for each integration method.

For Harmony, we performed a grid search over two hyperparameters: `theta` $\in \{2, 4, 8\}$, which controls the strength of the diversity clustering penalty and `nclust` $\in \{80, 100, 120\}$, which controls the number of clusters used in soft clustering. This grid search resulted in $3 \times 3 = 9$ parameter settings for each dataset. For Seurat-RPCA, we searched across the hyperparameter, `k.weight` $\in \{20, 60, 80, 100, 140\}$, which determines the number of neighbors considered when weighting anchors and is associated with the degree of between-batch variation removal. For scVI, we searched across the hyperparameter, `n_hidden` $\in \{64, 128, 256\}$, which specifies the number of nodes in each hidden layer of the neural network. For FastMNN, we searched across the hyperparameter, `k` $\in \{10, 20, 60, 100\}$, which specifies the number of nearest neighbors used for mutual nearest-neighbor matching across batches.

3.3.3 Generalization to multiple batches

Suppose there are $B > 1$ query batches in addition to one reference batch. Let

$$\mathbf{X}^b = (\mathbf{x}_1^b, \ldots, \mathbf{x}_{n_b}^b)^\top \in \mathbb{N}^{n_b \times p}, \quad b = 0, 1, \ldots, B,$$

denote the gene-count matrix for batch b, and let

$$\mathbf{E}^b = (\mathbf{e}_1^b, \ldots, \mathbf{e}_{n_b}^b)^\top \in \mathbb{R}^{n_b \times d}, \quad b = 0, 1, \ldots, B,$$

denote the corresponding low-dimensional embedding matrix. Here, $b = 0$ refers to the reference batch, whereas $b = 1, \ldots, B$ refer to the query batches.

For each query batch $b \in \{1, \ldots, B\}$, comparing it with the reference batch yields a gene-level BSS and a dimension-level BAS, denoted by BSS^b and BAS^b, respectively. We then define the overall BSS for each gene and the overall BAS for each dimension by averaging across query batches:

$$\mathrm{BSS} = \frac{1}{B} \sum_{b=1}^{B} \mathrm{BSS}^b, \qquad \mathrm{BAS} = \frac{1}{B} \sum_{b=1}^{B} \mathrm{BAS}^b.$$

The CIS generalizes naturally to the multi-batch setting by averaging the per-batch log-likelihood improvement over all batches, including the reference batch:

$$\mathrm{CIS} = \frac{1}{d(B+1)} \sum_{b=0}^{B} \frac{1}{n_b} \sum_{j=1}^{d} \left(\ell_{bj}^{\mathrm{mix}} - \ell_{bj}^{\mathrm{null}} \right).$$

Finally, the dataset-level integration score is still defined as

$$\text{Integration score} = \mathrm{BAS} + \mathrm{CIS}.$$

3.3.4 Validation of the statistical models underlying IntegrateRigor

To validate the modeling assumptions underlying IntegrateRigor, we examined the goodness-of-fit of the two probabilistic models: the NB mixture model (NBMM) for gene-level expression counts and the Gaussian mixture model (GMM) for each dimension of the integrated embedding. From the intestinal failure-associated liver disease (IFALD) dataset [90], we selected three donors comprising 32,397 genes and 15,895 cells. For embedding-level validation, we applied five integration methods (Seurat-RPCA, Harmony, scVI, FastMNN, and LIGER) with default settings to integrate the three batches and evaluated the GMM fit to the resulting integrated embedding (dimension-wise) for each method. For gene-level validation, we applied the NBMM to one batch before integration, as an illustrative example.

We quantified the goodness-of-fit using the Kolmogorov–Smirnov (KS) statistic [115], defined as the maximum absolute difference between the empirical distribution function and the fitted cumulative distribution function, to measure the magnitude of distributional discrepancy. We did not use the p-value to qualitatively assess goodness-of-fit, because it is well known that for large sample sizes—such as those in large-scale single-cell datasets—the KS p-value tends to become vanishingly small, so that even minor and practically negligible deviations from the fitted distribution can lead to rejection of the model [116].

For modeling the per-gene expression counts, the observed-versus-fitted histograms show that the NBMM closely matches the count distributions for representative genes in IFALD. This is consistent with extensive prior evidence that the NB distribution provides an effective model for overdispersed gene-expression count data in homogeneous cell populations (cells with the same underlying identity) [113]. Across the displayed examples, most genes have very small KS statistic values, often near zero and generally below 0.02, with only a few genes having values around 0.03 (Figure 3.6). This pattern extends beyond the examples shown. Across 2000 HVGs, the distribution of KS statistic values is concentrated near zero, with only a modest upper tail and a limited number of outliers (Figure 3.5a). Overall, these observations support the NBMM as a reasonable model for gene-level distribution modeling.

For modeling the per-dimension integration embedding values, Gaussian modeling is mo-

tivated both empirically and theoretically: classical results on linear dimension reduction imply that many low-dimensional embeddings of high-dimensional data are approximately Gaussian [117]. To evaluate this GMM assumption empirically, we fitted a GMM to each of the first five dimensions of the integrated embedding produced by each of the five integration methods and assessed the goodness-of-fit using the KS statistic (Figure 3.7). For scVI, Harmony, Seurat-RPCA, and FastMNN, the KS statistic values are consistently small across the first five dimensions, generally on the order of 0.004–0.016, indicating that their dimension-wise embedding values are well approximated by the GMM. The summary boxplot across dimensions shows the same trend, with the KS statistic values concentrated near zero (Figure 3.5b).

In contrast, the integrated embedding of LIGER exhibits substantially larger KS values, around 0.223–0.332 in the first five dimensions, indicating a poorer goodness-of-fit for GMM relative to the integrated embeddings of the other four methods. This result is consistent with the fact that LIGER is based on non-negative matrix factorization, which tends to produce embeddings with nonnegative, skewed, and less Gaussian-like distributions. As a result, we excluded LIGER from hyperparameter tuning and method selection analyses that rely on our integration score, which assumes a GMM.

Taken together, these results support the two core modeling assumptions in IntegrateRigor: the NBMM provides a good fit to per-gene expression counts, and the GMM provides a reasonable approximation for per-dimension values in the integrated embedding. In practice, these distributional assumptions should be checked before applying IntegrateRigor, to ensure that the NBMM and GMM remain appropriate and that IntegrateRigor results are valid for a given dataset.

3.3.5 Stability and convergence

The NBMM was fitted using a custom EM algorithm, whereas the GMM was fitted using the `Mclust` function in the R package `mclust`. The key parameters for IntegrateRigor are the number of mixture components, i.e., K_0, K_1, M_0, and M_1. However, for the mixture-

model fitting, as long as the specified number of mixture components is no smaller than the true number of underlying mixture components, our scores (BSS, BAS, and CIS) are expected to remain stable across different choices of K_0, K_1, M_0, and M_1 [**wang2025feature**]. By default, the number of mixture components was set to 5 for the NBMM and GMM ($K_0 = K_1 = M_0 = M_1 = 5$). Although the total number of cell identities (e.g., cell types) in a dataset is often larger than 5, the distribution for a single gene or a single embedding dimension typically does not exhibit many distinct modes, because each cell identity is defined jointly by multiple genes and multiple embedding dimensions. Empirically, we observed that the number of modes per gene or embedding dimension is modest, and setting the number of mixture components to 5 is sufficient in practice.

We further evaluated the stability of IntegrateRigor with respect to the specified number of mixture components. Specifically, we examined the stability of the BSS, the selected BSGs, and the integration score across varying numbers of mixture components: $K_0 = K_1 = M_0 = M_1 \in \{3, 4, 5, \ldots, 10\}$. Regarding the gene-wise NBMM fitting using our custom EM algorithm implementation, we vary the number of EM iterations from 10, 20, to 50, with 10 iterations used as the default. Across these settings, the resulting BSS values, whose calculation relies on NBMM fitting, were consistently highly correlated with those obtained under the default setting (with the Spearman correlation close to 1), and the selected BSGs showed substantial overlap with the default BSG set (with the Jaccard index for set similarity close to 1). Regarding the embedding-level GMM fitting under varying numbers of mixture components, the integration score consistently selected the same optimal hyperparameter. These results suggest that IntegrateRigor is a stable procedure. Further details are in Supplementary Figures 3.15–3.18.

3.3.6 Annotation-based evaluation metrics

3.3.6.1 Cell-type-aware integration local inverse Simpson's index (CiLISI)

The integration local inverse Simpson's index (iLISI) is a local neighborhood-based metric for batch mixing. For each cell, iLISI measures the diversity of cells from different batches

among its nearest neighbors in the integrated embedding. A higher iLISI value indicates that the cell's local neighborhood contains a more balanced mixture of batches, whereas a lower value indicates that the neighborhood is dominated by a subset of batches.

To evaluate batch mixing while accounting for cell identity preservation, we used a cell-type-aware version of iLISI [24, 55, 81], termed as CiLISI. For each cell type, we restricted the integrated embedding to cells annotated as that cell type and computed iLISI using batch labels within this cell subset. Thus, each cell's iLISI value reflects batch diversity among its nearest neighbors from the same cell type, avoiding confounding from differences of distinct cell types.

We then averaged the cell-level values within each cell type to obtain a cell-type-specific iLISI. These cell-type-specific iLISIs were further averaged across cell types, weighted by cell-type proportions, to obtain the final CiLISI score. Higher CiLISI indicates better mixing of batches within the same cell type.

3.3.6.2 Adjusted Rand index (ARI)

To evaluate preservation of cell identity structure, we used the ARI between unsupervised clusters obtained from the integrated embedding and known cell identity labels [24, 55]. For each integrated embedding, we performed Seurat Louvain clustering over a range of resolution parameters from 0.1 to 1.0, in increments of 0.05. At each resolution, we computed the ARI between the resulting cluster assignments and the known cell identity labels. The maximum ARI across all tested resolutions was taken as the final ARI score for that embedding. This procedure avoids dependence on a single clustering resolution and captures the extent to which the integrated embedding reflects the cell identity structure.

3.3.6.3 Cell-type local inverse Simpson's index (cLISI)

For each cell, cLISI measures the diversity of cell-type labels among its nearest neighbors. A lower cLISI value indicates that the local neighborhood is dominated by cells of the same cell type, reflecting better preservation of cell identity structure. A higher cLISI value indicates

stronger local mixing of different cell types, which may suggest loss of cell type separation or over-integration.

We first computed LISI using cell-type labels across all cells, without conditioning on batch, and refer to these scores as cell-level cLISI values. Each cell's cLISI value reflects the local diversity of cell-type labels among its neighboring cells, allowing cLISI to detect the mixing of distinct cell types across batches and thereby capture potential over-integration. We then averaged the cell-level cLISI values across cells to obtain the dataset-level cLISI value.

3.4 Discussion

IntegrateRigor reframes data integration from a heuristic preprocessing step into a dataset-driven, statistically guided optimization problem for cell identity recovery. Rather than introducing a new integration method, it provides a method-agnostic framework for improving existing methods through annotation-free batch-stable gene selection and hyperparameter tuning. Across diverse single-cell and spatial transcriptomics tasks, this strategy improved cell identity recovery by better balancing batch mixing and cell identity preservation, reducing both under-integration and over-integration, and revealing biologically meaningful structure that was not apparent under default settings.

Our results highlight two underappreciated points about integration practices. First, gene selection is itself a major determinant of cell identity recovery—a point that receives far less attention than the choice of integration method, even though the selected genes define the information the method operates on when aligning cells across batches. By identifying BSGs and excluding genes dominated by between-batch variation, IntegrateRigor improves the ability of diverse methods to recover comparable cell identities across batches. Second, integration performance depends not only on the method chosen but also on how that method is configured for a given dataset. Harmony showed only moderate performance with default settings but became one of the best-performing methods after IntegrateRigor-based optimization. This illustrates that method rankings can shift substantially when gene selection

and hyperparameter tuning are made more principled. More broadly, these findings argue for moving beyond static, one-size-fits-all integration pipelines and toward dataset-specific optimization, where the central question is how the method configuration best supports cell identity recovery for a particular dataset.

The quality of the integrated embedding has consequences that extend well beyond the integration step itself. Because cell identity recovery is a prerequisite for any cell-identity-based downstream analysis, the reliability of the integrated embedding determines the quality of all analyses that follow. For example, Crescendo [57] and CellANOVA [84] both take the integrated embedding directly as input to perform count-level batch correction and cell-type-specific differential expression analysis under particular experimental designs, respectively. Because their cell-state representations and decompositions are estimated from the integrated embedding, the reliability of their outputs depends directly on the quality of cell identity recovery in that embedding. An under-integrated embedding retains residual batch structure that can confound cell-state estimation; an over-integrated embedding blurs biologically distinct states, causing cell-type mixing that corrupts cell-type-specific analyses. Providing these downstream methods with a well-calibrated embedding—one that supports reliable cell identity recovery across batches—is therefore a prerequisite for accurate downstream inference.

It is worth clarifying the relationship between IntegrateRigor and CellANOVA [84]. CellANOVA recovers biological signals associated with within-cell-identity between-batch variation that is orthogonal to the experimental design—variation that may be inadvertently removed during integration. In that sense it addresses a form of over-integration, but its aim is distinct from ours. IntegrateRigor addresses the upstream task of cell identity recovery: reliably aligning cells with the same identity across batches before any downstream analysis is performed. For this cell identity recovery task, within-cell-identity between-batch variation is treated as residual under-integration to be corrected, not as signal to be recovered. Moreover, CellANOVA still depends on an existing integrated embedding to estimate its cell-state encoding, so its performance remains sensitive to the quality of that upstream embedding. IntegrateRigor addresses this dependency by providing an optimized embedding

as input.

The importance of integrated embedding for cell identity recovery was directly illustrated in the colorectal cancer application. Applying IntegrateRigor to obtain an optimized Harmony embedding enabled more reliable cell identity recovery, which in turn revealed previously unidentified cancer–immune interface niches in which distinct immune cell populations showed transcriptional and spatial signatures consistent with direct interaction with malignant epithelial cells. Without careful gene selection and parameter tuning, the default Harmony embedding left residual batch structure that prevented coherent cell identity recovery, whereas the embedding from the original study overcorrected and obscured these interface populations. This example illustrates how integration optimization for cell identity recovery directly shapes which cell populations can be distinguished.

Several extensions could further strengthen this framework. First, the current formulation focuses on cell identity preservation as the primary objective of integration optimization, treating between-batch variation in cell identity as the target of correction. A natural next step is to incorporate known sample-level covariates (e.g., treatment condition, disease status, or developmental stage) to distinguish generic batch effects from structured biological variation of interest. With this experimental design information, the same optimization principles could be extended beyond the cell identity annotation task to directly optimize the integrated embedding for downstream analyses such as those performed by Crescendo and CellANOVA: for example, optimizing integration that maximizes the power for condition-specific differential expression while controlling for batch. This extension reflects the natural order of analysis: first obtain reliable cell identities, then leverage them for cell-identity-based inference. Second, IntegrateRigor relies on two modeling assumptions: a negative binomial mixture model (NBMM) for per-gene expression counts and a Gaussian mixture model (GMM) for per-dimension values in the integrated embedding. In practice, these assumptions should be checked before applying IntegrateRigor to a new dataset, to ensure that both models provide adequate approximations and that the resulting scores are interpretable. This is particularly important for the GMM underlying the integration score, since embeddings produced by some methods, such as LIGER, may deviate from this assumption.

Future work could therefore explore more flexible latent models or nonparametric density estimators that better capture the geometry of different embeddings. Third, our findings also suggest that future benchmarks should evaluate methods not only under default settings but also after dataset-specific optimization, as default performance may substantially underestimate the quality achievable for cell identity recovery with principled configuration. Fourth, the same principles may extend to cross-modality integration, where preserving shared cell identity across different feature spaces presents additional challenges, and to semi-supervised settings in which partial reference annotations are available, analogous to reference-guided approaches such as scANVI [82, 118]. Fifth, we have so far assumed that a single reference batch contains all relevant cell identities, an assumption that is increasingly reasonable given the availability of large atlases that can serve as references. In settings where each batch contains only a subset of cell identities, a natural extension would be to construct a synthetic reference by merging and aligning mixture-model components across batches, rather than relying on a single reference batch; this will require methodological development in future work.

3.5 Code and data availability

3.5.1 Data availability

The IFNB dataset can be accessed through the `SeuratData` package using `SeuratData::LoadData("ifnb")`. The human immune dataset can be accessed at https://doi.org/10.6084/m9.figshare.12420968. The IFALD dataset can be accessed at https://cellxgene.cziscience.com/collections/ff69f0ee-fef6-4895-9f48-6c64a68c8289. The chicken heart dataset can be accessed at GSE149457. The colorectal cancer spatial transcriptomics and single-cell RNA-sequencing dataset can be accessed at https://zenodo.org/records/14602110. The mouse brain spatial transcriptomics dataset was obtained from the Allen Brain Cell Atlas using the `abc_atlas_access` package. Specifically, we used three Zhuang-ABCA slices, `Zhuang-ABCA-1.079`, `Zhuang-ABCA-2.037`, and `Zhuang-ABCA-3.010`, from the `Zhuang-ABCA-1`, `Zhuang-ABCA-2`, and `Zhuang-ABCA-3` datasets, respectively. The data can be accessed through

the Allen Brain Cell Atlas data-access portal and Python package at https://alleninstitute.github.io/abc_atlas_access/.

3.5.2 Code availability

The IntegrateRigor R package is publicly available at https://github.com/zhiqianZ/IntegrateRigor.

3.6 Acknowledgements

This chapter is based on my joint work with Dr. Chuanghu Wang, Chengfeng Jiang, Ziqi Rong, and my Ph.D. advisor, Dr. Jingyi Jessica Li. We appreciate the comments and feedback from members of the Junction of Statistics and Biology at UCLA (http://jsb.ucla.edu).

3.7 Figures

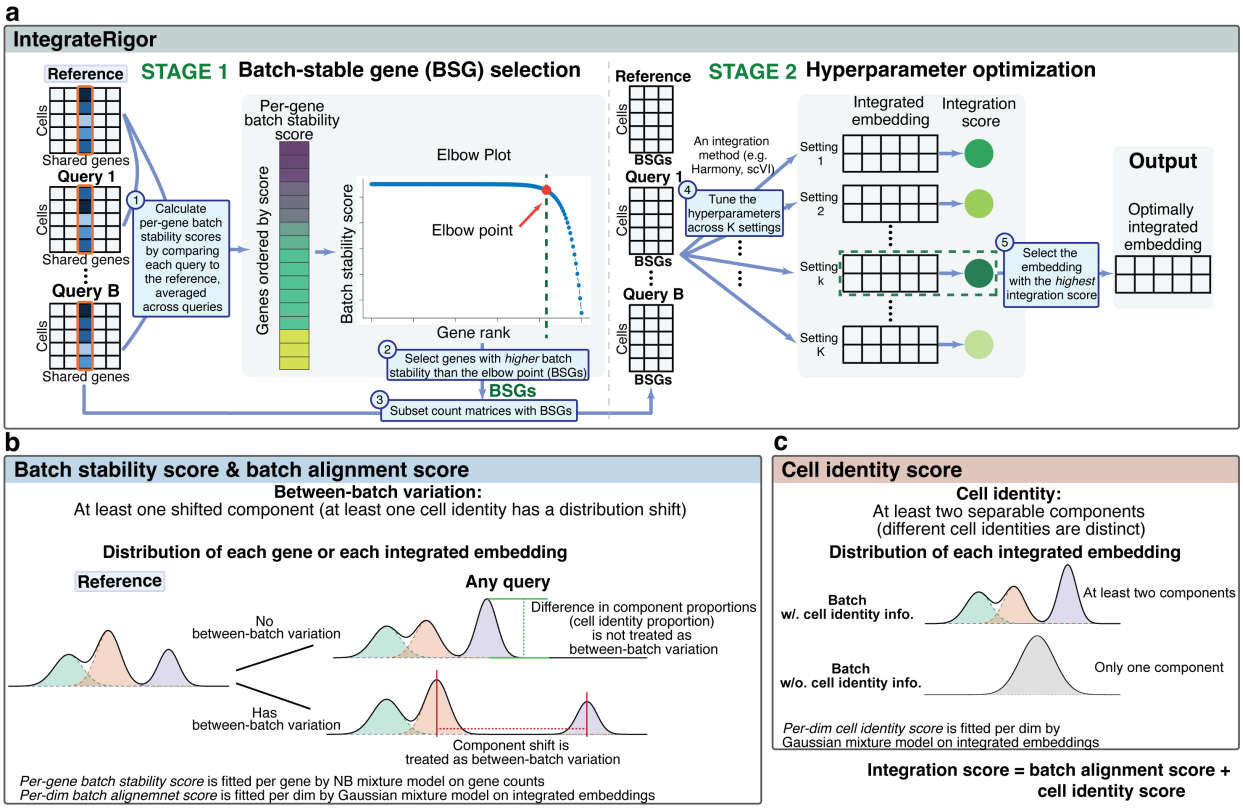


Figure 3.1: Overview of IntegrateRigor framework. a. Schematic of the IntegrateRigor workflow. In Stage 1, batch-stable genes (BSGs) are selected by comparing each query batch to a reference batch and computing a per-gene batch stability score (BSS), which quantifies the extent to which gene-level distributions are preserved across batches. Genes ranked above the elbow point are retained as BSGs. In Stage 2, count matrices are restricted to BSGs, and candidate hyperparameter settings for a chosen integration method are evaluated. The integrated embedding with the highest integration score is selected as optimal. **b.** Illustration of the BSS and batch alignment score (BAS). For each gene or each integrated embedding dimension, between-batch variation is modeled as a distributional shift in at least one mixture component corresponding to a shared cell identity across batches. Differences in mixture component proportions are not treated as between-batch variation. BSS is estimated on gene counts using a negative binomial mixture model, whereas per-dimension BASs are estimated on integrated embeddings using a Gaussian mixture model. **c.** Illustration of the cell identity score (CIS). For each integrated embedding dimension, preservation of cell identity is quantified by the presence of multiple Gaussian mixture components, indicating distinguishable cell identities. The per-dimension CIS is estimated using a Gaussian mixture model on the integrated embedding. The overall integration score is defined as the sum of BAS and CIS.

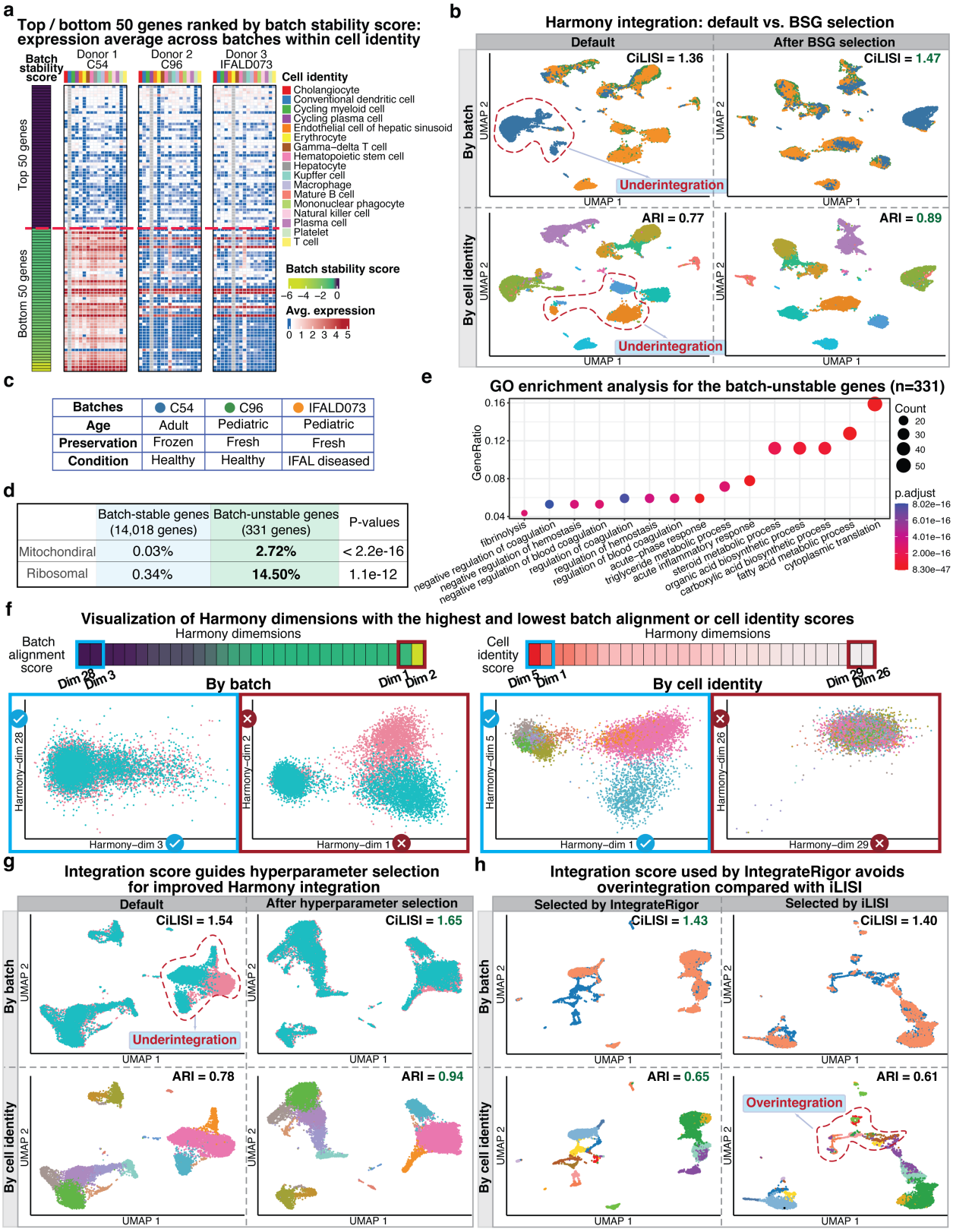
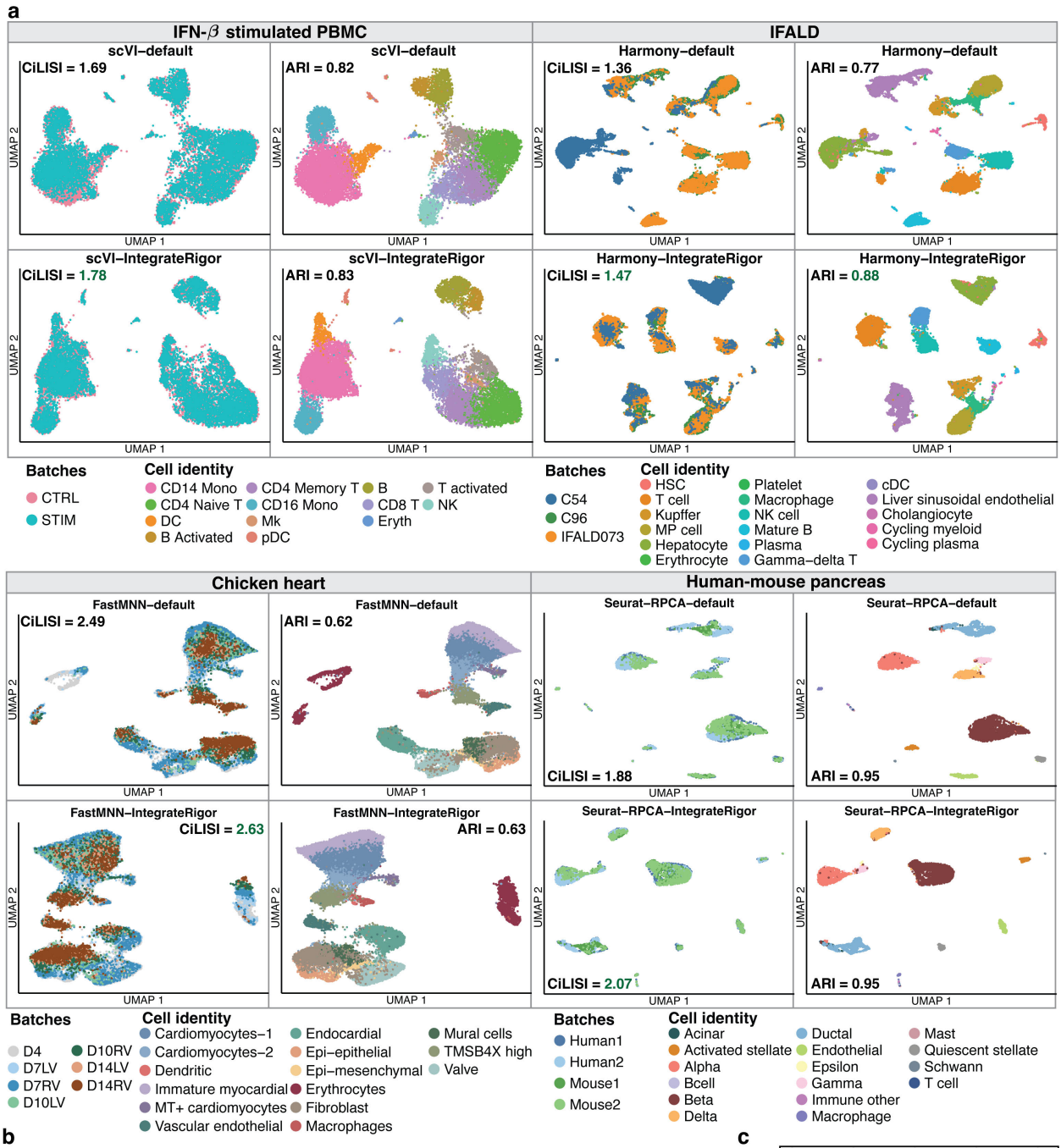

a
Top / bottom 50 genes ranked by batch stability score: expression average across batches within cell identity
Batch stability score
Donor 1 C54
Donor 2 C96
Donor 3 IFALD073
Top 50 genes
Bottom 50 genes
Cell identity
Cholangiocyte
Conventional dendritic cell
Cycling myeloid cell
Cycling plasma cell
Endothelial cell of hepatic sinusoid
Erythrocyte
Gamma−delta T cell
Hematopoietic stem cell
Hepatocyte
Kupffer cell
Macrophage
Mature B cell
Mononuclear phagocyte
Natural killer cell
Plasma cell
Platelet
T cell
Batch stability score
−6 −4 −2 0
Avg. expression
0 1 2 3 4 5
b
Harmony integration: default vs. BSG selection
Default
After BSG selection
By batch
By cell identity
CiLISI = 1.36
CiLISI = 1.47
ARI = 0.77
ARI = 0.89
Underintegration
Underintegration
UMAP 1
UMAP 2
c
Batches | C54 | C96 | IFALD073
Age | Adult | Pediatric | Pediatric
Preservation | Frozen | Fresh | Fresh
Condition | Healthy | Healthy | IFAL diseased
d
Batch-stable genes (14,018 genes) | Batch-unstable genes (331 genes) | P-values
Mitochondiral | 0.03% | 2.72% | < 2.2e-16
Ribosomal | 0.34% | 14.50% | 1.1e-12
e
GO enrichment analysis for the batch-unstable genes (n=331)
GeneRatio
0.16
0.12
0.08
0.04
Count
20
30
40
50
p.adjust
8.02e−16
6.01e−16
4.01e−16
2.00e−16
8.30e−47
fibrinolysis
negative regulation of coagulation
negative regulation of hemostasis
negative regulation of blood coagulation
regulation of coagulation
regulation of hemostasis
regulation of blood coagulation
acute−phase response
triglyceride metabolic process
acute inflammatory response
steroid metabolic process
organic acid biosynthetic process
carboxylic acid biosynthetic process
fatty acid metabolic process
cytoplasmic translation
f
Visualization of Harmony dimensions with the highest and lowest batch alignment or cell identity scores
Batch alignment score
Harmony dimemsions
Dim 28
Dim 3
Dim 1
Dim 2
By batch
Harmony−dim 28
Harmony−dim 3
Harmony−dim 2
Harmony−dim 1
Cell identity score
Harmony dimemsions
Dim 5
Dim 1
Dim 29
Dim 26
By cell identity
Harmony−dim 5
Harmony−dim 1
Harmony−dim 26
Harmony−dim 29
g
Integration score guides hyperparameter selection for improved Harmony integration
Default
After hyperparameter selection
By batch
By cell identity
CiLISI = 1.54
CiLISI = 1.65
ARI = 0.78
ARI = 0.94
Underintegration
UMAP 1
UMAP 2
h
Integration score used by IntegrateRigor avoids overintegration compared with iLISI
Selected by IntegrateRigor
Selected by iLISI
By batch
By cell identity
CiLISI = 1.43
CiLISI = 1.40
ARI = 0.65
ARI = 0.61
Overintegration
UMAP 1
UMAP 2

Figure 3.2: IntegrateRigor identifies batch-unstable genes and uses integration scores to improve gene selection and hyperparameter selection while avoiding overintegration. a. Heatmaps of the top and bottom 50 genes ranked by BSS, showing average expression across batches within each cell identity. Top-ranked genes are batch-stable, whereas bottom-ranked genes exhibit substantial batch-dependent variability. **b.** Harmony integration before and after selecting highly variable genes from the BSGs set. BSG selection improves batch mixing (higher CiLISI) while better preserving cell identity structure (higher ARI), reducing underintegration. **c.** Summary of the three IFALD batches, which differ in donors' age, conditions, and sample preservation. **d.** Comparison of mitochondrial and ribosomal gene proportions between batch-stable and batch-unstable genes, showing strong enrichment of ribosomal genes among the batch-unstable genes. Here, the batch-unstable genes are the genes with low BSS scores and are excluded from the integration. **e.** Gene Ontology enrichment analysis of the batch-unstable genes, highlighting biological processes related to coagulation, inflammatory response, lipid metabolism, and cytoplasmic translation. **f.** Visualization of Harmony dimensions with the highest and lowest batch alignment or CISs. Dimensions with high BASs show better batch mixing, whereas dimensions with high CISs better preserve biological structure. **g.** Integration scores guide hyperparameter selection for Harmony, improving both batch correction and cell identity preservation relative to the default setting. **h.** Compared with selection based on iLISI, IntegrateRigor selects a solution that avoids overintegration and better preserves cell identity, as reflected by a lower cLISI. Cell identity labels and color annotations for the UMAPs are provided in Supplementary Figures 3.11-Figure 3.13.



b

Relative performance gains from IntegrateRigor across diverse integration tasks

Datasets		IFNB-β stimulated PBMC		Human immune		IFALD		Chicken heart		Human-mouse pancreas		Mouse brain (spatial)	
Major sources of between-batch variation		Sample conditions		Sequencing platforms		Sample conditions		Developmental stages		Species		Samples & tissue sections	
Integration methods	Harmony	+12.0%	+17.5%	+5.1%	+2.2%	+8.4%	+15.0%	+1.6%	+3.9%	+20.2%	+41.0%	+0.7%	+0.4%
	Seurat-RPCA	+5.5%	-2.1%	+3.1%	-0.5%	+1.7%	-2.1%	-1.4%	+2.5%	+10.2%	-0.4%	+0.1%	+0.3%
	scVI	+5.2%	+1.0%	+9.5%	+1.7%	+6.8%	+5.8%	+0.3%	+0.2%	+2.8%	+0.8%	+2.4%	+3.2%
	FastMNN	+16.9%	+26.2%	+20.6%	+39.4%	+1.9%	+1.7%	+5.9%	+1.7%	+2.8%	-0.35%	-1.0%	-0.5%
	LIGER	+9.6%	-2.7%	-0.0%	+19.6%	+7.2%	+16.2%	+0.7%	+1.9%	+7.5%	+18.7%	+3.1%	+1.2%

CiLISI gain with IntegrateRigor over default integration

ARI gain with IntegrateRigor over default integration

c

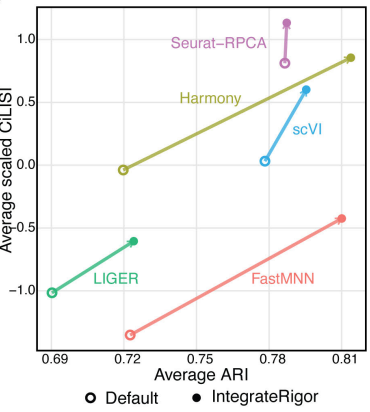


Figure 3.3: IntegrateRigor improves integration performance across diverse datasets and methods. a. Representative UMAPs for four datasets comparing default integration with IntegrateRigor-optimized integration. Cells are colored by batch (first column in each dataset's subfigure) or cell identity (second column in each dataset's subfigure). Across datasets and methods, IntegrateRigor improves batch mixing (CiLISI) while preserving or improving cell identity separation (ARI). **b.** Relative gains in CiLISI and ARI obtained by IntegrateRigor over default integration across six benchmark datasets and five integration methods. Results are grouped by the major source of between-batch variation. **c.** Method-level summary of average ARI and scaled CiLISI before and after IntegrateRigor optimization. Filled circles denote IntegrateRigor-selected solutions and open circles denote default settings.

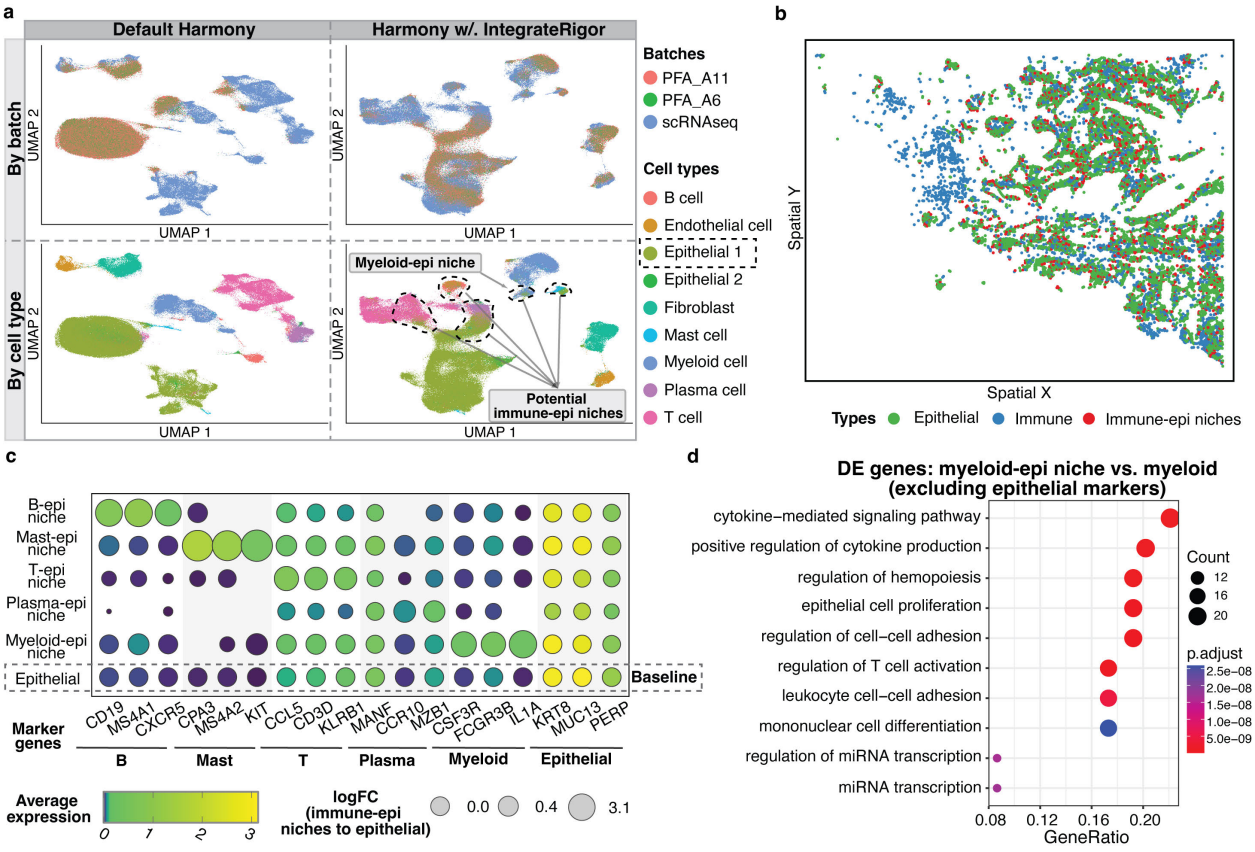


Figure 3.4: IntegrateRigor reveals unique cancer–immune interface niche in colorectal cancer data. a, The original Crescendo integration and annotations. The top panels show the original Crescendo embedding and spatial coordinates before integration, and the bottom panel shows the Harmony-integrated embedding colored by technology. **b**, UMAPs from Harmony integration with IntegrateRigor-selected genes, colored by batch or by Crescendo cell identity annotations. Regions where originally annotated epithelial and immune cells were intermixed are highlighted with dashed circles and were further clustered and annotated as distinct cancer–immune niches based on the IntegrateRigor embedding. **c**, Refined cell identity annotations for the spatial transcriptomics samples PFA_A11 and PFA_A6. Immune cells located near epithelial cancer cells were further annotated as cancer–immune niche populations, including cancer–B niche, cancer–mast niche, cancer–myeloid niche, cancer–plasma niche, and cancer–T niche. **d**, Spatial distribution of the refined cell identities in PFA_A11. The main spatial map shows epithelial cancer cells, fibroblast and endothelial cells, immune cells, and cancer–immune niches. The inset highlights a zoomed-in region where cancer–immune niche cells are spatially intermingled with epithelial cancer cells. **e**, Dot plot of canonical marker genes across refined cell identity groups. Dot color represents average expression, and dot size represents the log fold change of each cancer–immune niche population relative to epithelial cancer cells. Cancer–immune niche groups retained expression of their corresponding immune markers while showing epithelial marker expression at levels comparable to epithelial cancer cells. **f**, Gene Ontology enrichment analysis of differentially expressed genes comparing the cancer-myeloid niche with myeloid cells after excluding epithelial marker genes. Enriched pathways include cytokine-mediated signaling, regulation of cytokine production, regulation of cell–cell adhesion, and T cell activation, supporting the biological distinction of the cancer-myeloid niche from conventional myeloid cells.

3.8 Supplementary materials

S3.8.1 Dataset Summary

Table S3.1: Summary of datasets used in this study.

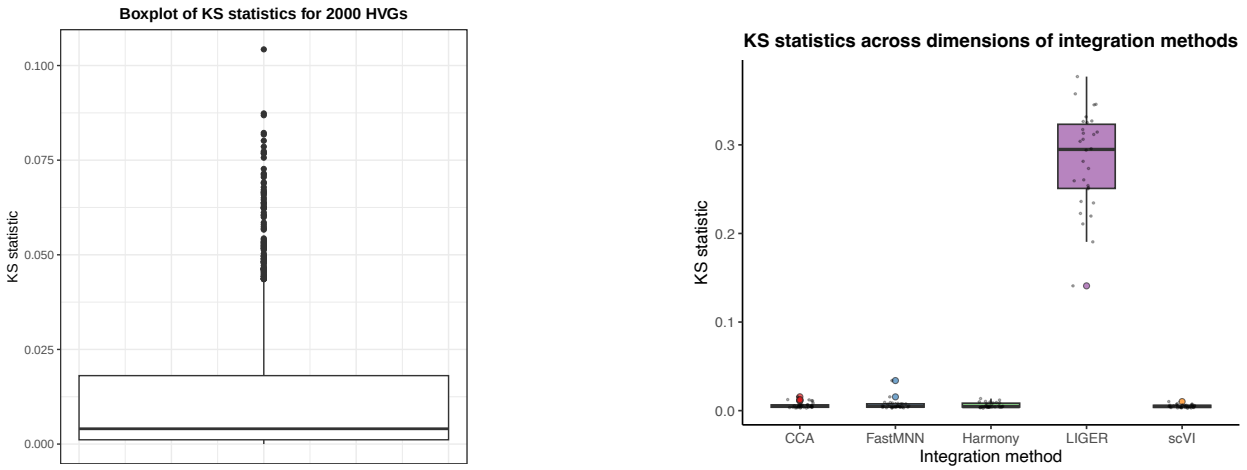
Dataset	#Cells	#Batches	#Cell identities	#Genes	Reference
IFN-β stimulated PBMC	13999	2	13	14053	[1]
IFALD	15895	3	17	32397	[2]
Immune	32484	9	16	12303	[3]
Chicken heart	22315	7	15	24356	[4]
Human–mouse pancreas	5547	4	17	11820	[5]
Mouse brain spatial	137442	3	34	1122	[6]
CRC (scRNA-seq & spatial)	192166	3	9	477	[7]

S3.8.2 Empirical Validation of IntegrateRigor

To empirically assess whether the distributional assumptions used in IntegrateRigor are reasonable for real single-cell data, we evaluated the goodness-of-fit of the two key modeling components on the IFALD dataset. First, we examined the negative binomial mixture model (NBMM) used to model gene expression counts. For the 2,000 highly variable genes, we compared the observed nonzero count distributions with the fitted distributions and quantified the discrepancy using the Kolmogorov–Smirnov (KS) statistic. Gene-level histograms show close agreement between the observed and fitted distributions after excluding the zero-count bin, and the overall distribution of KS statistics remains small for most genes, suggesting that the NBMM provides an adequate approximation to the empirical gene-expression distributions.

Second, we evaluated the Gaussian mixture model (GMM) assumption used for integrated embedding dimensions. For each integration method, we fitted a GMM with K = 5 components to each embedding dimension and compared the fitted density with the empirical distribution. Across Seurat-RPCA, FastMNN, Harmony, and scVI, the fitted GMMs closely matched the observed embedding distributions, with generally small KS statistics across dimensions. In contrast, LIGER showed substantially larger KS statistics, indicating poorer agreement with the GMM assumption for its embedding space. These results suggest

that the modeling assumptions underlying IntegrateRigor are broadly compatible with commonly used integration outputs, while also allowing the framework to detect cases where an embedding distribution deviates from the assumed mixture structure.



(a) NBMM KS statistics

(b) GMM KS statistics

Figure 3.5: Goodness-of-fit assessment using KS statistics on the IFALD dataset. Left panel: Distribution of KS statistics for 2,000 highly variable genes fitted by the negative binomial mixture model (NBMM), showing generally small distributional discrepancies between the observed and fitted gene expression distributions. Right panel: Distribution of KS statistics across embedding dimensions for Gaussian mixture model (GMM) fits to integrated embeddings from different integration methods. Most methods show small KS statistics, whereas LIGER exhibits substantially larger KS values, indicating poorer compatibility with the GMM assumption.

S3.8.3 Genes contributing to batch effect in the unintegrated PCA overlaps with the batch-unstable genes

We examined whether genes identified by IntegrateRigor as batch-unstable correspond to genes that visually drive batch separation before integration. To do this, we generated a PCA biplot using the unintegrated gene expression matrix. In this plot, each point represents a cell projected onto the first two principal components, with colors indicating batch identity. The arrows represent the top 100 genes with the largest contributions to the PCA axes; the direction and length of each arrow indicate how strongly and in which direction each gene contributes to variation in the PCA space.

The biplot shows clear batch-associated structure in the unintegrated embedding. Importantly, many genes whose loading directions align with batch-separated regions are classified by IntegrateRigor as batch-unstable. In contrast, batch-stable genes are less concentrated along the major batch-separating directions. This overlap supports the interpretation that

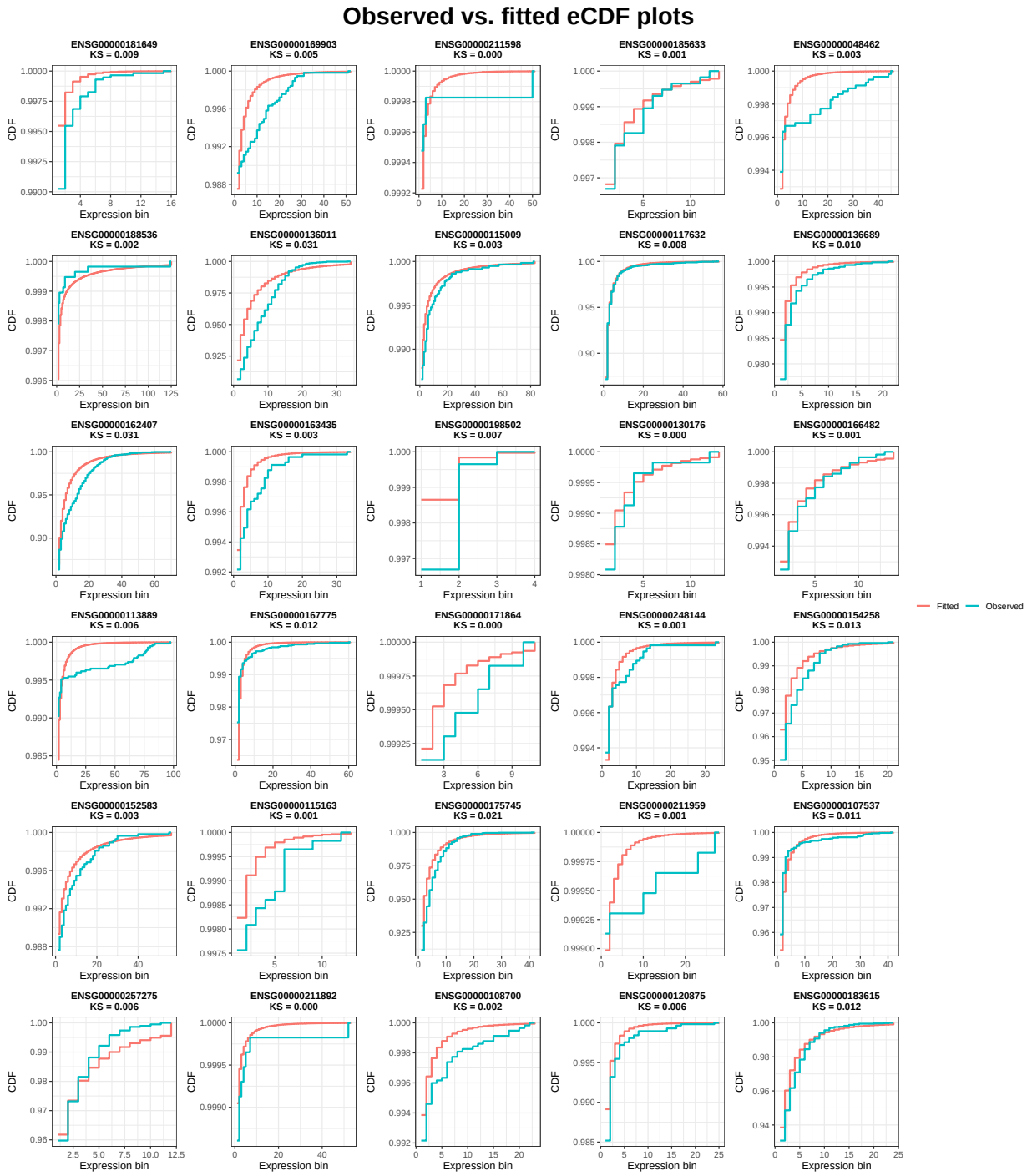


Figure 3.6: Observed and fitted gene-expression count distributions for randomly selected genes in the IFALD datasets. For each selected gene, the histogram compares the empirical nonzero count distribution (Observed) with the model-implied fitted distribution (Fitted), excluding the zero-count bin to better visualize the fit among expressed cells. Each panel reports the gene Ensembl ID and the Kolmogorov–Smirnov (KS) statistic between the observed and fitted nonzero distributions. Smaller KS values indicate closer agreement between the fitted model and the observed distribution.

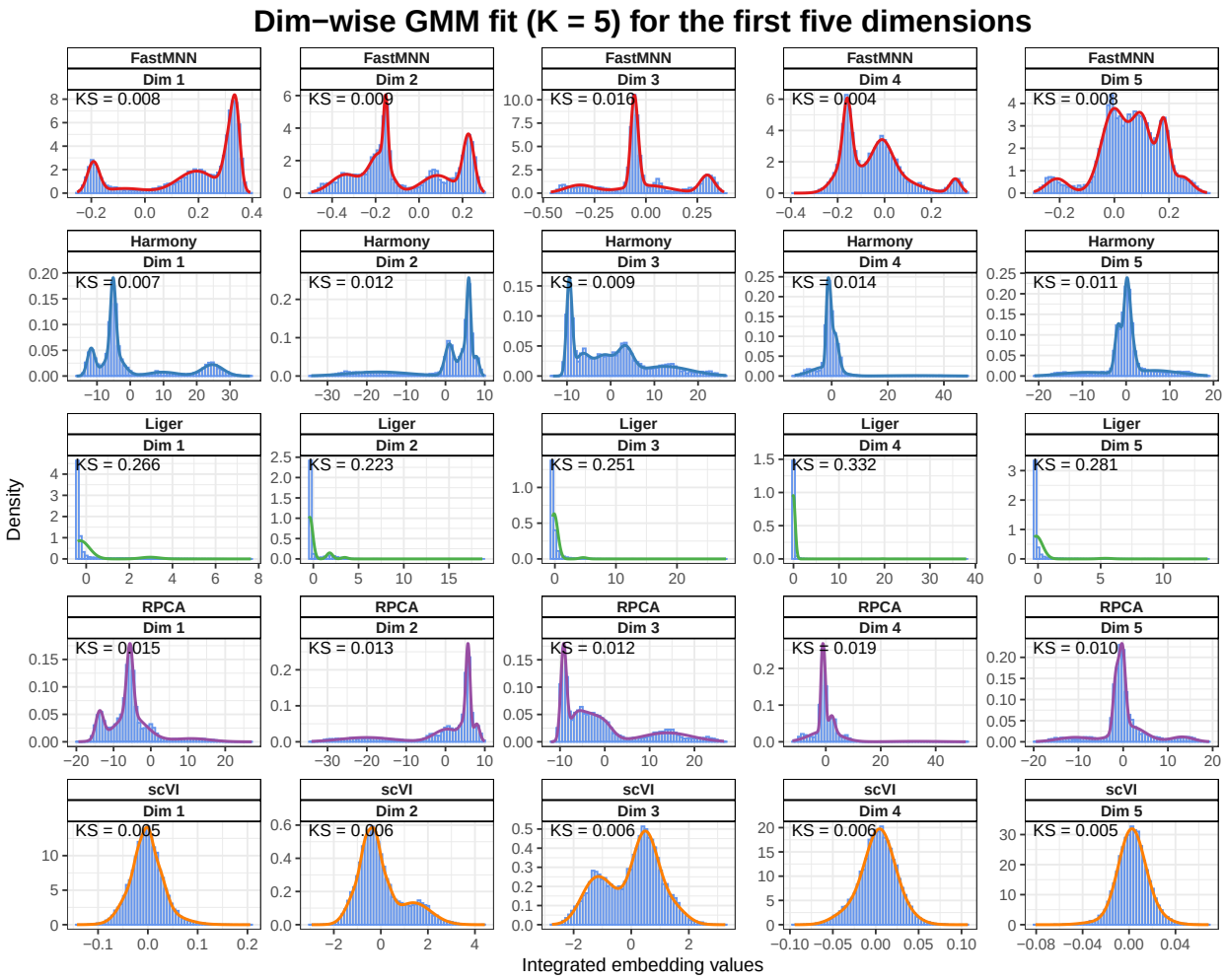


Figure 3.7: Dimension-wise Gaussian mixture model fits for integrated embeddings on the IFALD datasets. For each integration method, the distribution of embedding values is shown for the first five dimensions. Blue histograms represent the empirical distribution of integrated embedding values, and the overlaid colored density curves represent the fitted Gaussian mixture model with K = 5 components. Each panel reports the corresponding integration method, embedding dimension, and Kolmogorov–Smirnov (KS) statistic between the observed and fitted distributions. Smaller KS values indicate better agreement between the fitted mixture model and the empirical embedding distribution.

IntegrateRigor identifies genes whose variation is driven by batch effects and should therefore be down-weighted or excluded during integration-specific feature selection.

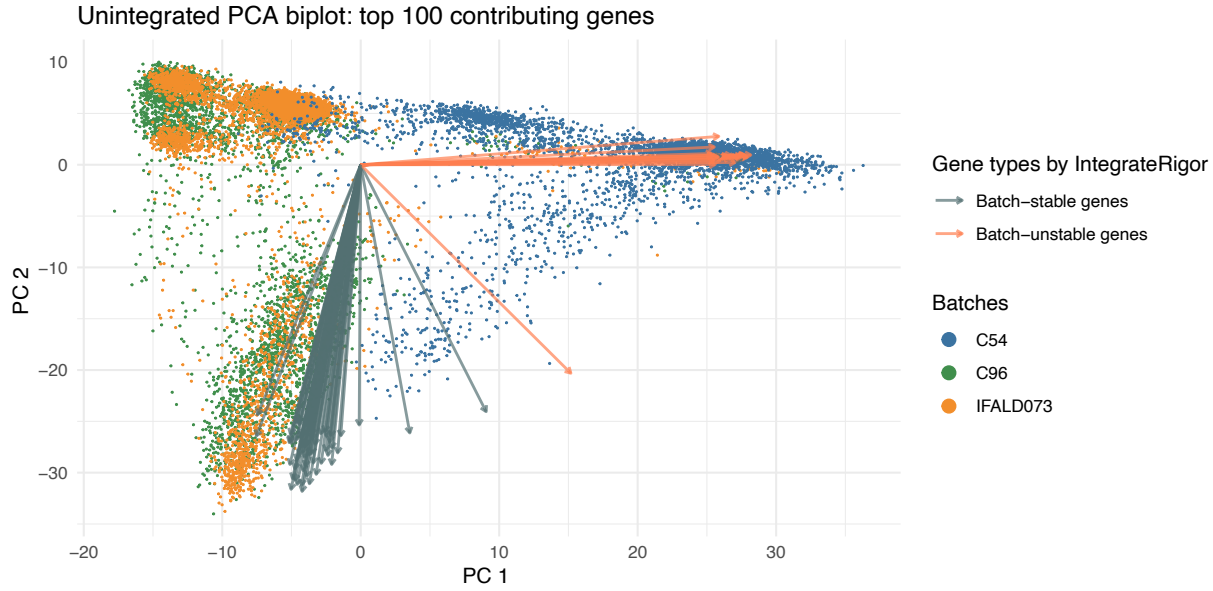


Figure 3.8: PCA biplot for batch-stability assessment of top contributing genes in the unintegrated PCA space. The unintegrated PCA embedding is shown for cells from three batches, with cells colored by batch. Arrows represent the top 100 genes contributing to the PCA structure, and arrow colors indicate whether each gene is classified by IntegrateRigor as batch-stable or batch-unstable. Batch-unstable genes tend to align with directions separating batches, whereas batch-stable genes are less associated with batch-driven variation, supporting the use of IntegrateRigor for identifying genes that are less affected by batch effects before integration.

S3.8.4 Batch stability score correlates with annotation-dependent GTE scores

To validate the gene-level batch stability score used in IntegrateRigor, we also compared it with an existing annotation-dependent measure of gene-level technical effects (GTEs). GTEs quantify the extent to which each gene is affected by technical variation across batches or technologies, but their estimation relies on prior cell identity annotations. In contrast, IntegrateRigor estimates gene-wise batch stability directly from the observed expression distribution without requiring cell type or state labels. Therefore, a strong positive association between IntegrateRigor's batch-instability score and GTEs would support the validity of our annotation-free scoring strategy.

Across six datasets, we observed consistently positive correlations between the negative batch stability score from IntegrateRigor and the overall technical effects estimated by GTEs. Each point represents one gene, and both quantities are shown on the log scale. Genes with

larger GTE values tend to receive larger negative batch stability scores, indicating that IntegrateRigor assigns higher batch-instability scores to genes with stronger annotation-dependent technical effects. The Pearson correlations ranged from moderate to strong across datasets and were statistically significant in all cases.

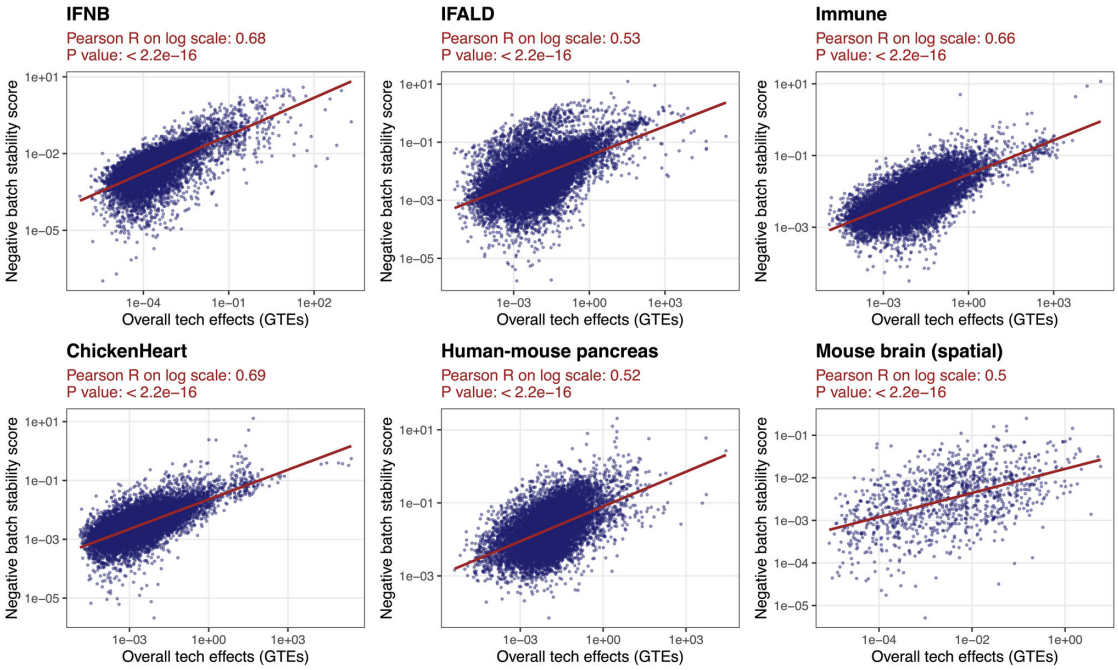


Figure 3.9: Correlation between gene-wise technical effects and IntegrateRigor batch-instability scores across datasets. Each panel shows the relationship between the overall gene-level technical effect estimated by GTEs and the negative batch stability score computed by IntegrateRigor on the log scale. Each point represents one gene, and both axes are shown on the log scale. The red line indicates the fitted linear trend. Across all six datasets, genes with larger estimated technical effects tend to have higher negative batch stability scores, with significant positive Pearson correlations. This agreement suggests that IntegrateRigor effectively identifies genes whose expression variation is strongly associated with batch- or technology-driven effects.

S3.8.5 IntegrateRigor identifies ribosomal genes more frequently as batch-unstable genes

We examined the biological and technical characteristics of genes classified by IntegrateRigor as batch-stable genes (BSGs) or batch-unstable genes (BUSGs). Because ribosomal and mitochondrial genes are commonly associated with technical variation, sequencing depth, cell quality, or sample-processing differences, we evaluated whether these gene categories were enriched among BUSGs across datasets.

We notice that ribosomal genes were consistently enriched among batch-unstable genes across datasets(Table S3.2), but individual ribosomal genes differed in their stability. For the

four datasets, i.e., datasets excluding the cross-species and spatial datasets, there were still 15 ribosomal genes repeatedly classified as batch-stable (Fig. 3.10). To better understand this difference in stability among ribosomal genes, we compared ribosomal genes identified as unstable in at least one dataset with ribosomal genes that remained stable across all datasets. The unstable ribosomal genes (e.g. *RPL21*, *RPL3*, *RPL5*, *RPL26*, *RPS3*, *RPS6*) were genes encoding the core structural components of the ribosome, whereas the consistently stable ribosomal genes (e.g. *RPS27L*, *RPL22L1*, *RPS6KB1*, *RPS6KB2*) were enriched for ribosome-related genes with additional or specialized functions, such as ribosome-associated regulation [8–11]. This suggests that ribosomal genes are not uniformly sensitive to between-batch variations, consistent with previous gene-wise analysis [12].

	Counts		Ribosomal genes (%)		Mitochondrial genes (%)	
Dataset	BSG	BUSG	BSG	BUSG	BSG	BUSG
IFNB	8831	14018	0.85	6.82	0.00	0.00
IFALD	10522	10136	0.34	14.50	0.03	2.72
Immune	264	331	0.59	29.79	0.00	0.00
ChickenHeart	94	78	0.56	29.49	0.08	5.13

Table S3.2: Ribosomal and mitochondrial gene enrichment among IntegrateRigor-defined batch-stable and batch-unstable genes across datasets. Here, BSG denotes batch-stable genes. BUSG denotes batch-unstable genes.

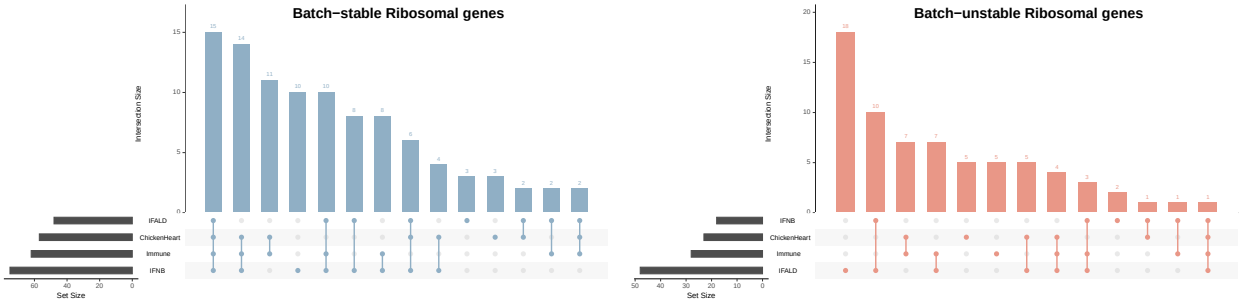


Figure 3.10: Cross-dataset overlap of ribosomal genes classified as batch-stable or batch-unstable by IntegrateRigor. Upset plots showing the intersection patterns of ribosomal genes identified as batch-stable genes (BSGs; top or batch-unstable genes (BUSGs; bottom) across datasets. The horizontal bars indicate the total number of ribosomal genes in each dataset-specific BSG or BUSG set, while the vertical bars indicate the size of each cross-dataset intersection. Connected dots below each vertical bar denote the datasets included in the corresponding intersection. The results show that ribosomal genes classified as BUSGs are repeatedly shared across datasets, supporting their association with batch-driven technical variation.

S3.8.6 Figures with complete legends for Main Figure

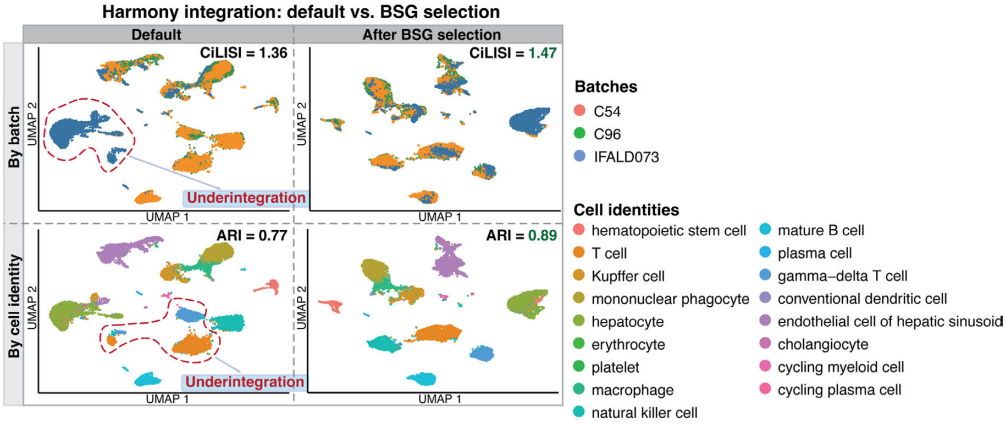


Figure 3.11: Main Figure 2b with complete batch and cell identity labels

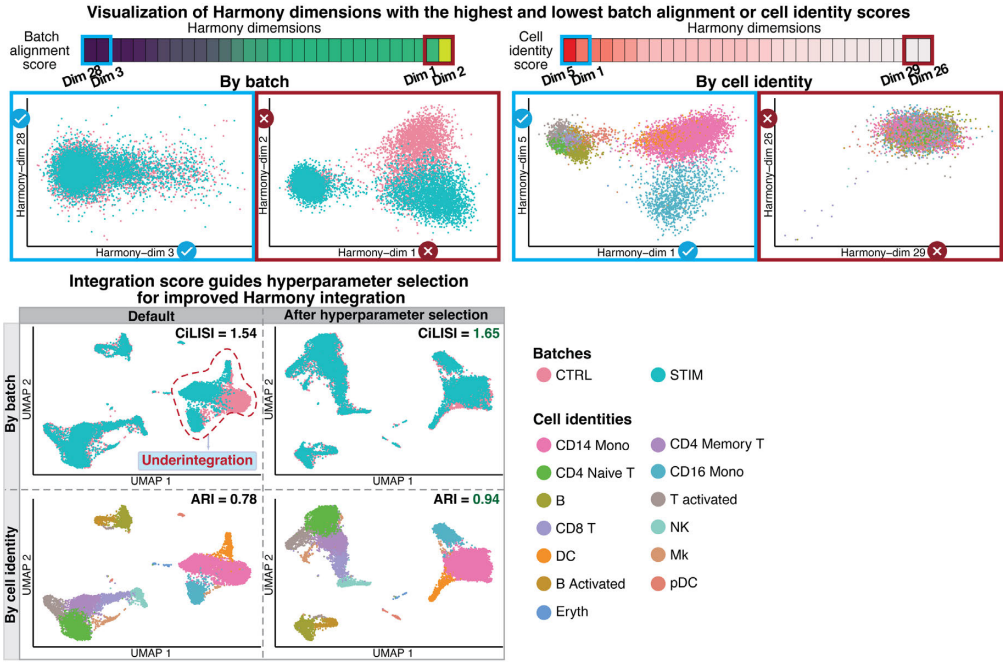


Figure 3.12: Main Figure 2 f-g with complete batch and cell identity labels

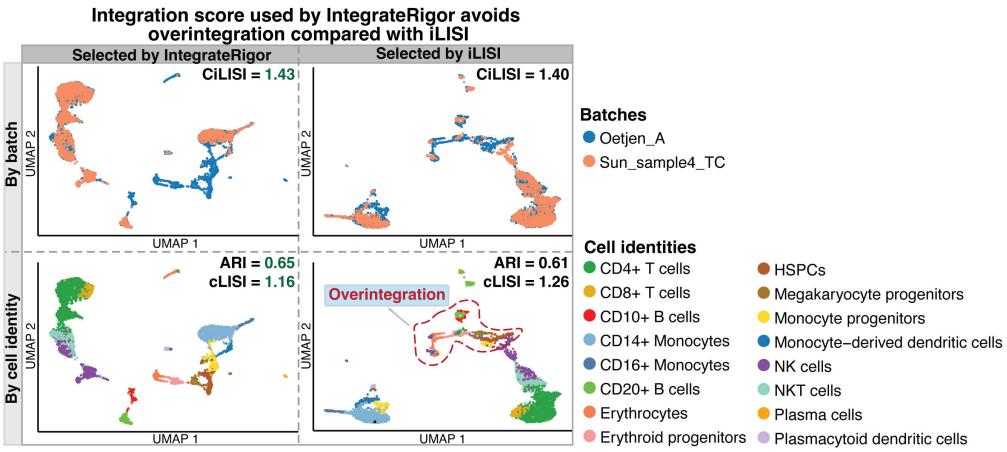


Figure 3.13: Main Figure 2h with complete batch and cell identity labels

S3.8.7 Harmony parameters selected by existing metrics

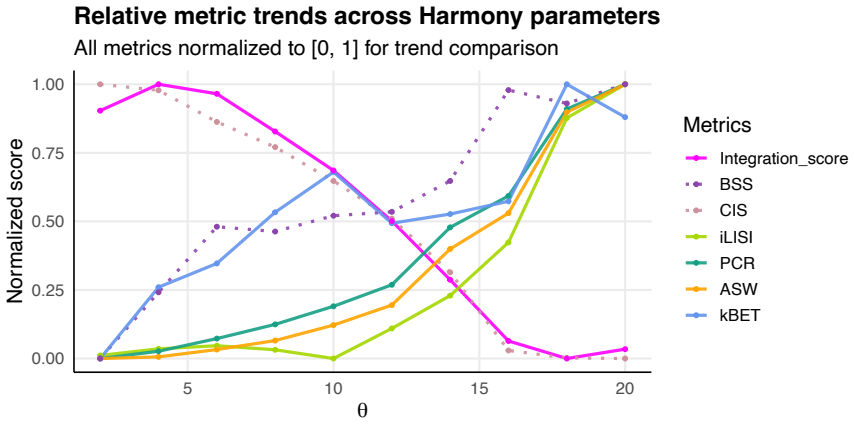


Figure 3.14: Relative metric trends across Harmony integration parameters. Line plot showing how integration performance metrics change across different values of the Harmony parameter θ. In Harmony, it controls the strength of the diversity clustering penalty, with larger values encouraging stronger batch mixing during integration. All metrics are min–max normalized to the range $[0, 1]$ to facilitate comparison of relative trends across metrics. The IntegrateRigor integration score peaks at moderate θ values and decreases as θ becomes larger, suggesting that overly strong batch correction may reduce biological structure preservation. In contrast, several conventional batch-mixing metrics, including LISI, PCR, ASW, and kBET, continue to improve with larger θ, highlighting their tendency to favor stronger integration even when it may lead to over-integration.

S3.8.8 IntegrateRigor sensitivity and convergence to the IntegrateRigor hyperparameters

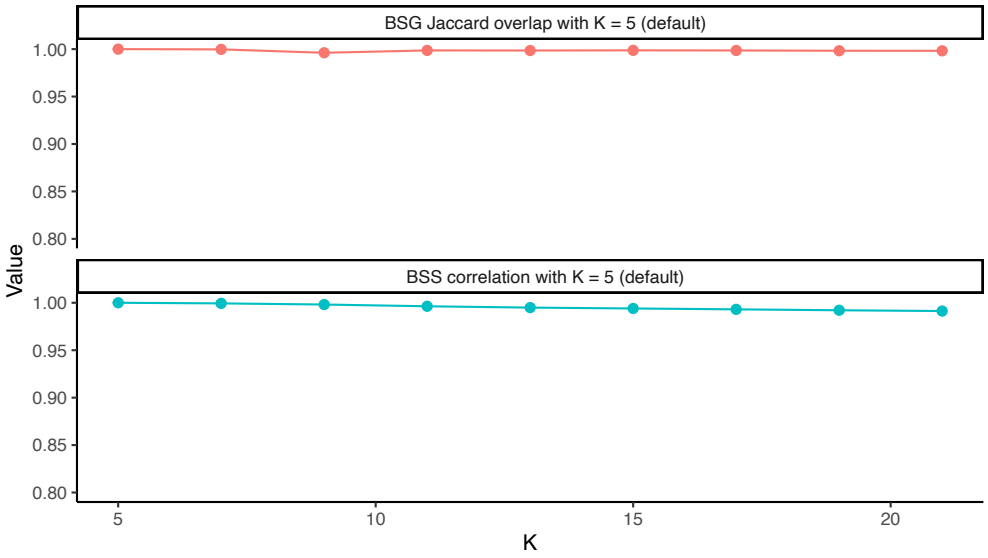


Figure 3.15: Stability of BSG selection and batch stability scores across different numbers of mixture components. Line plots showing the robustness of IntegrateRigor outputs across different choices of K, where K = 5 is the default value. The top panel shows the Jaccard overlap between the BSGs selected at each K and those selected at K = 5. The bottom panel shows the correlation between gene-wise batch stability scores (BSSs) computed at each K and those computed at K = 5.

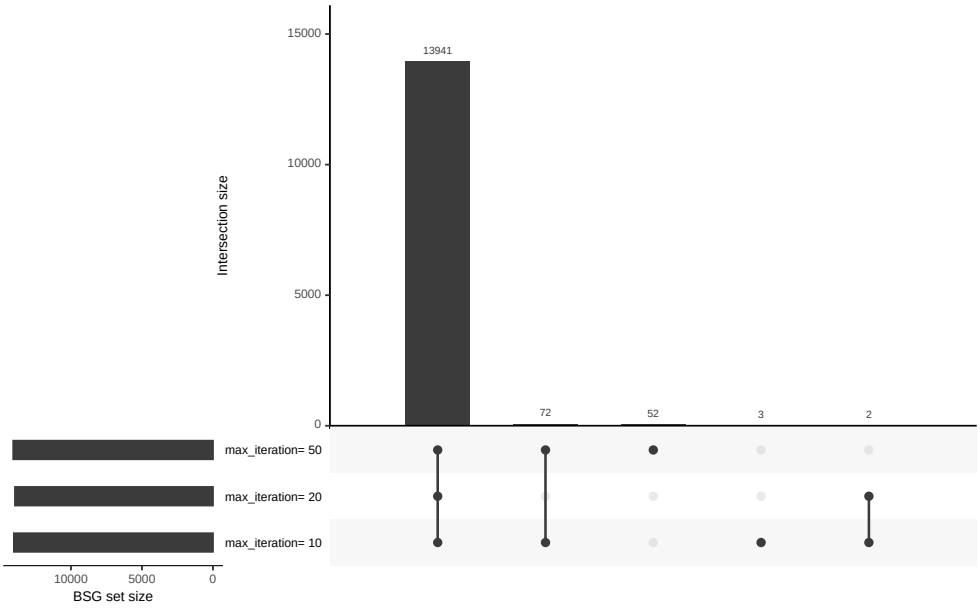


Figure 3.16: Overlap of selected batch-stable genes across different numbers of EM algorithms' iterations per gene. UpSet plot showing the overlap of batch-stable genes (BSGs) selected by IntegrateRigor under different maximum numbers of EM algorithms' iterations per gene. Horizontal bars indicate the number of BSGs selected under each iteration setting, vertical bars indicate intersection sizes, and connected dots denote the iteration settings included in each intersection.

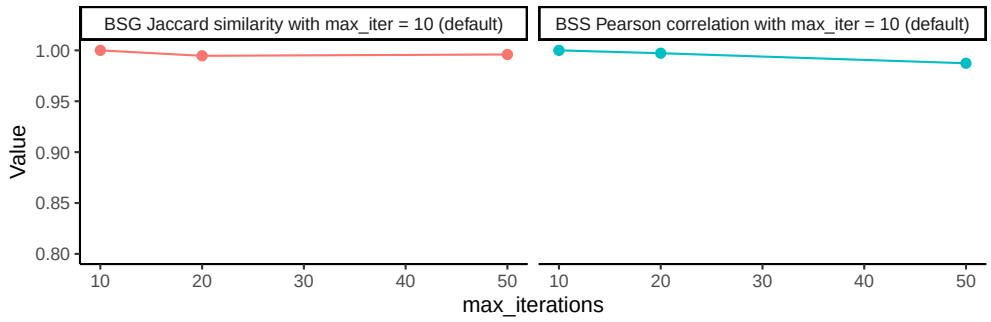


Figure 3.17: Stability of BSG selection and batch stability scores across different numbers of EM algorithms' iterations per gene. Line plots showing the convergence of IntegrateRigor outputs across different maximum numbers of EM algorithms' iterations per gene, where iterations=10 is the default value. The left panel shows the Jaccard similarity between BSGs selected at each iteration setting and those selected with the default setting. The right panel shows the Pearson correlation between gene-wise batch stability scores (BSSs) computed at each iteration setting and those computed with the default setting.

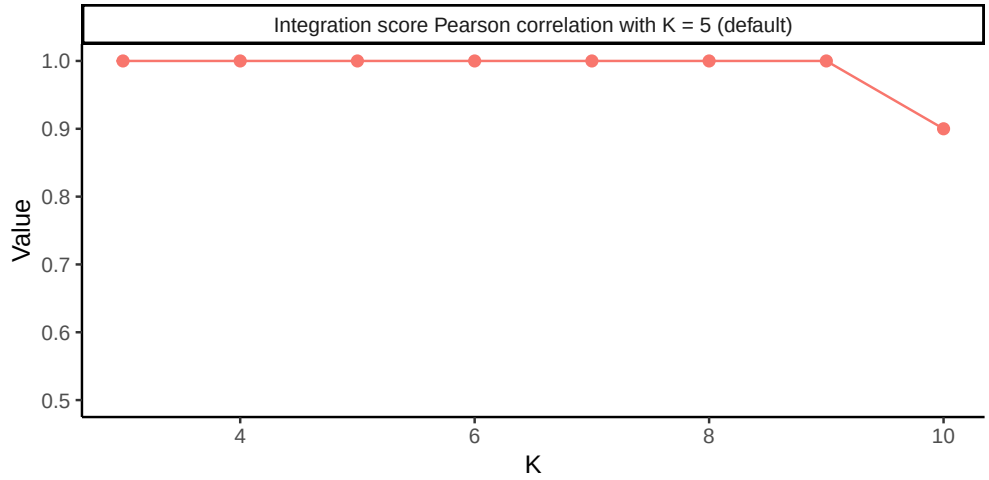


Figure 3.18: Stability of integration score across different numbers of Gaussian mixture components. Line plots showing the correlation of the integration scores across different choices of K among the same set of hyperparameter values in Harmony, where K = 5 is the default value.

S3.9 Complete IntegrateRigor results

S3.9.0.1 Metric evaluation

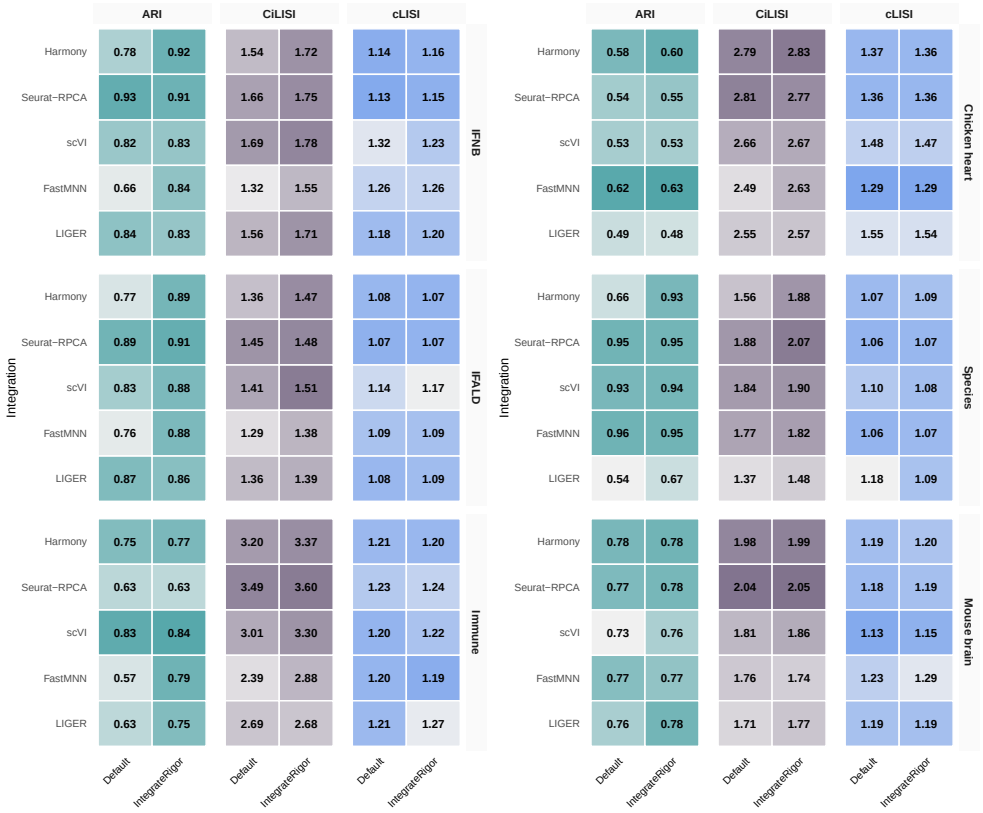


Figure 3.19: Metrics evaluation for IntegrateRigor vs. default integration over ARI, CiLISI, and cLISI.

S3.9.0.2 UMAP visualization of default integration vs. IntegrateRigor

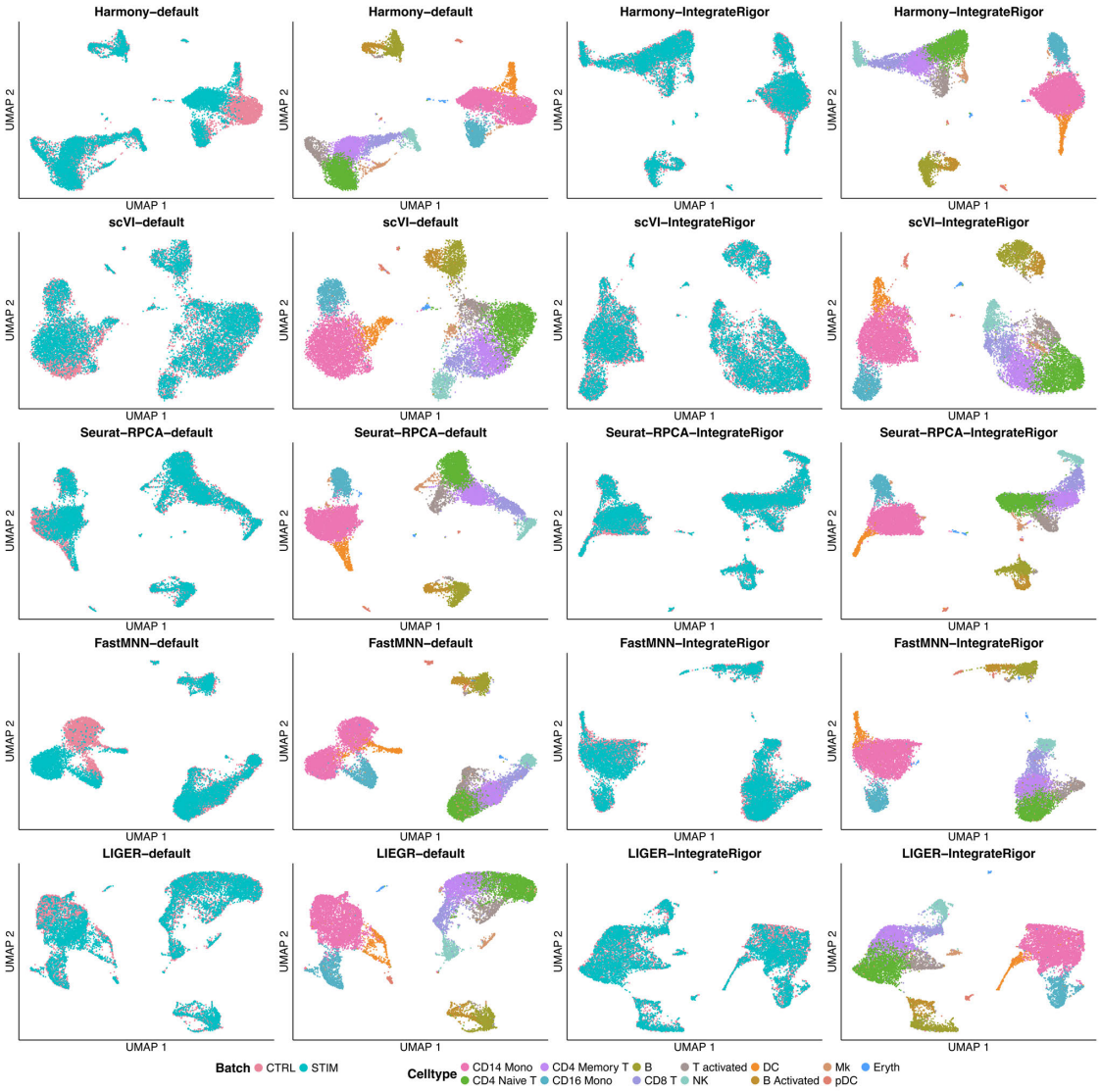


Figure 3.20: Comparison of default and IntegrateRigor-selected integration results on the IFN-β stimulated dataset. UMAP visualizations of the dataset after integration using Harmony, scVI, Seurat-RPCA, FastMNN, and LIGER under either default feature/parameter settings or IntegrateRigor-selected settings. For each method and setting, cells/spots are shown twice: colored by batch on the left and by annotated cell type on the right.

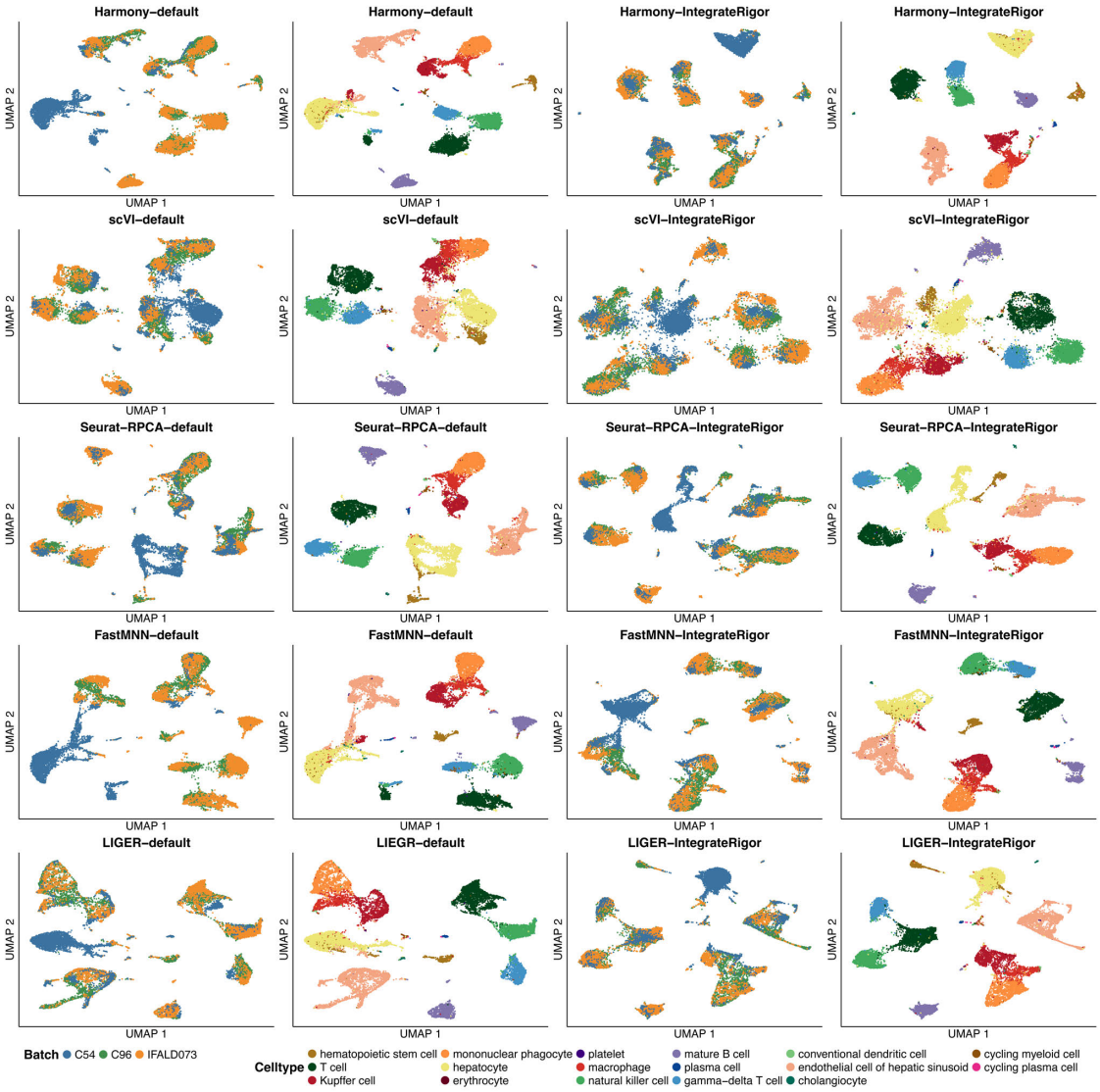


Figure 3.21: Comparison of default and IntegrateRigor-selected integration results on the IFALD dataset. UMAP visualizations of the dataset after integration using Harmony, scVI, Seurat-RPCA, FastMNN, and LIGER under either default feature/parameter settings or IntegrateRigor-selected settings. For each method and setting, cells/spots are shown twice: colored by batch on the left and by annotated cell type on the right.

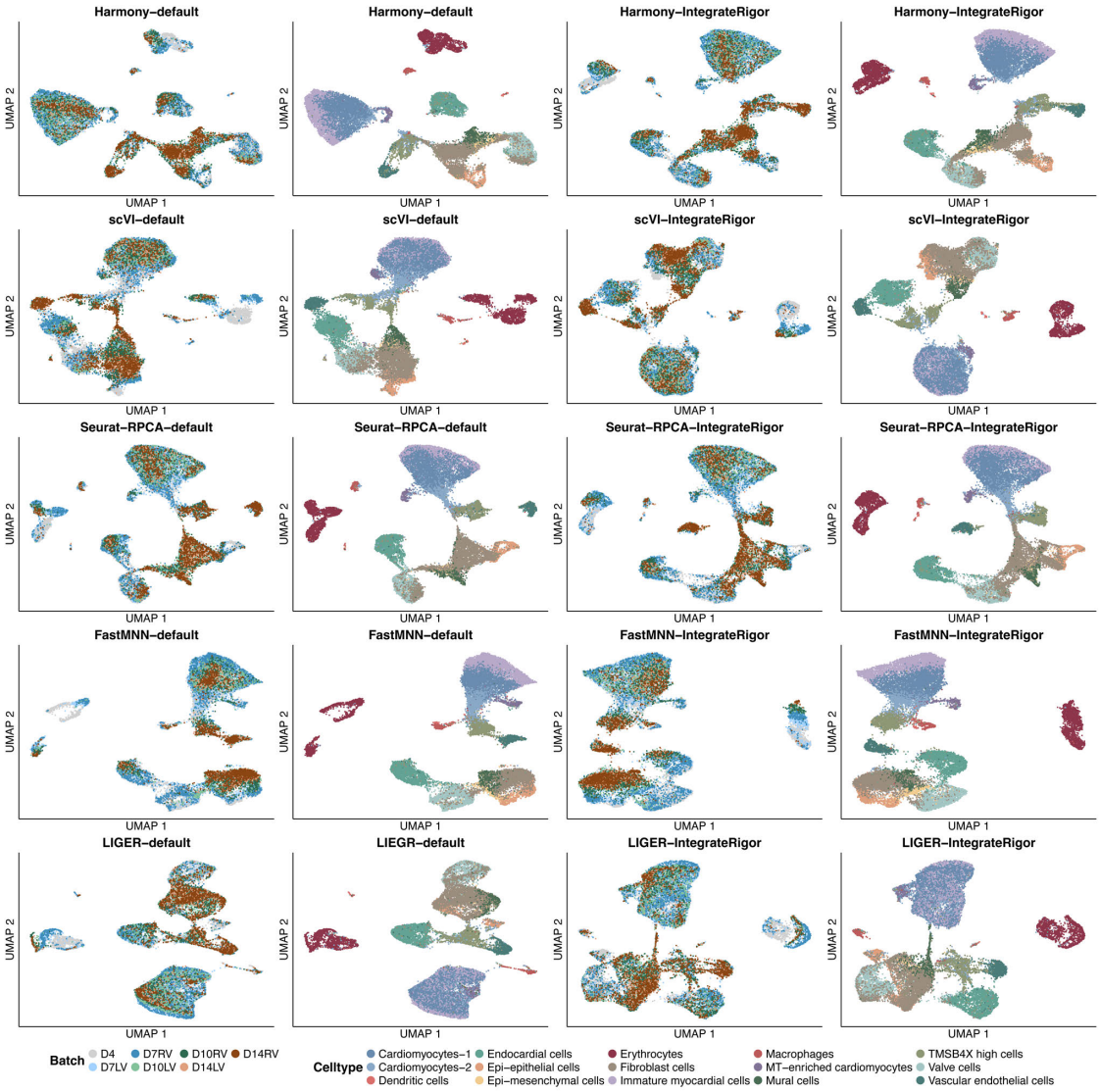


Figure 3.22: Comparison of default and IntegrateRigor-selected integration results on the chicken heart dataset. UMAP visualizations of the dataset after integration using Harmony, scVI, Seurat-RPCA, FastMNN, and LIGER under either default feature/parameter settings or IntegrateRigor-selected settings. For each method and setting, cells/spots are shown twice: colored by batch on the left and by annotated cell type on the right.

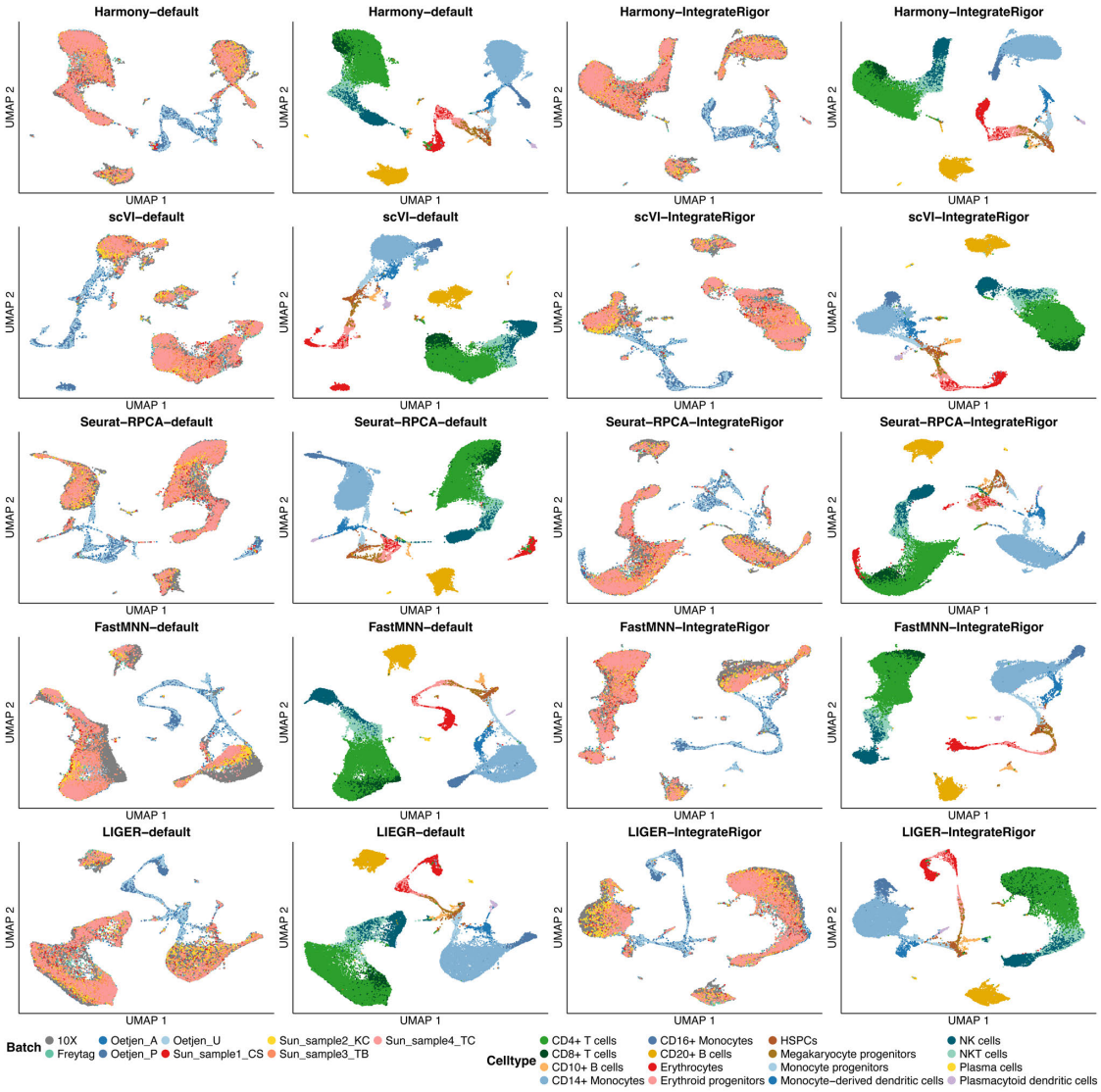


Figure 3.23: Comparison of default and IntegrateRigor-selected integration results on the immune dataset. UMAP visualizations of the dataset after integration using Harmony, scVI, Seurat-RPCA, FastMNN, and LIGER under either default feature/parameter settings or IntegrateRigor-selected settings. For each method and setting, cells/spots are shown twice: colored by batch on the left and by annotated cell type on the right.

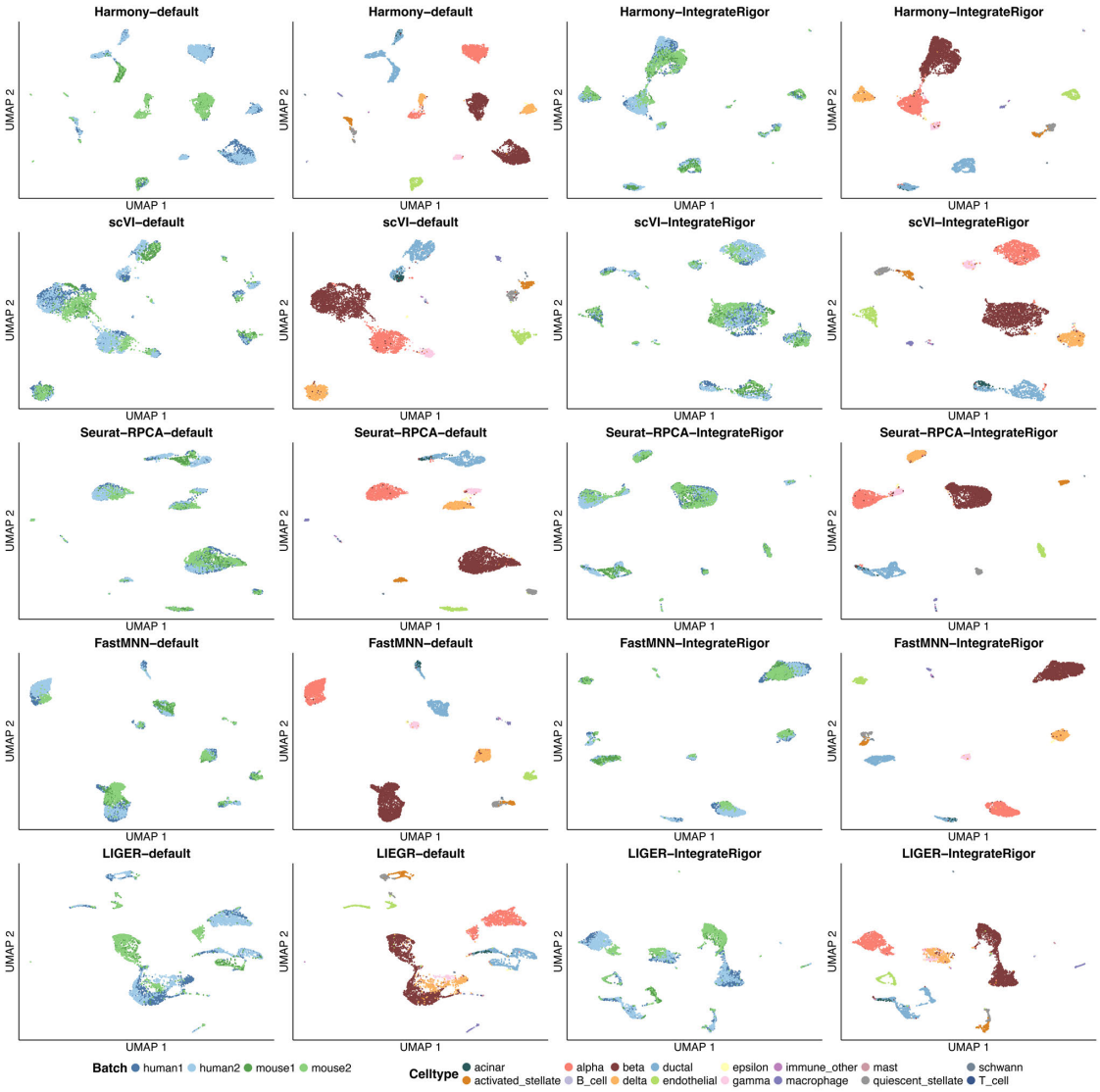


Figure 3.24: Comparison of default and IntegrateRigor-selected integration results on the human-mouse pancreas dataset. UMAP visualizations of the dataset after integration using Harmony, scVI, Seurat-RPCA, FastMNN, and LIGER under either default feature/parameter settings or IntegrateRigor-selected settings. For each method and setting, cells/spots are shown twice: colored by batch on the left and by annotated cell type on the right.



Figure 3.25: Comparison of default and IntegrateRigor-selected integration results on the mouse brain spatial dataset. UMAP visualizations of the dataset after integration using Harmony, scVI, Seurat-RPCA, FastMNN, and LIGER under either default feature/parameter settings or IntegrateRigor-selected settings. For each method and setting, cells/spots are shown twice: colored by batch on the left and by annotated cell type on the right.

S3.9.1 Integration score across different methods

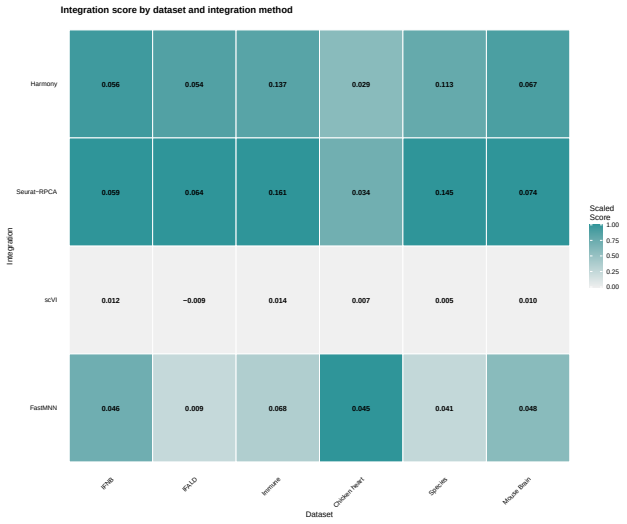


Figure 3.26: Integration score values across different methods and datasets

S3.9.2 Cancer-immune niches in the colorectal cancer dataset

S3.9.2.1 Default Harmony integration results

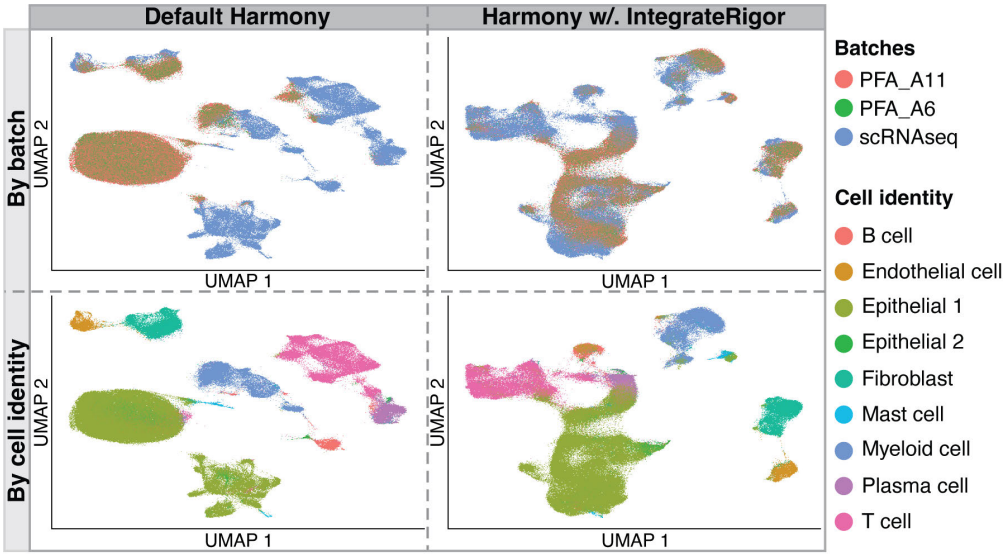


Figure 3.27: Comparison of default Harmony integration and Harmony integration with IntegrateRigor for colorectal cancer dataset. UMAP embeddings are shown for each integration strategy, with columns corresponding to default Harmony and Harmony with IntegrateRigor, and rows colored by batch or cell identity. For default Harmony, cells from different batches remain partially separated, indicating underintegration.

S3.9.2.2 GO terms for different immune-epithelial niches

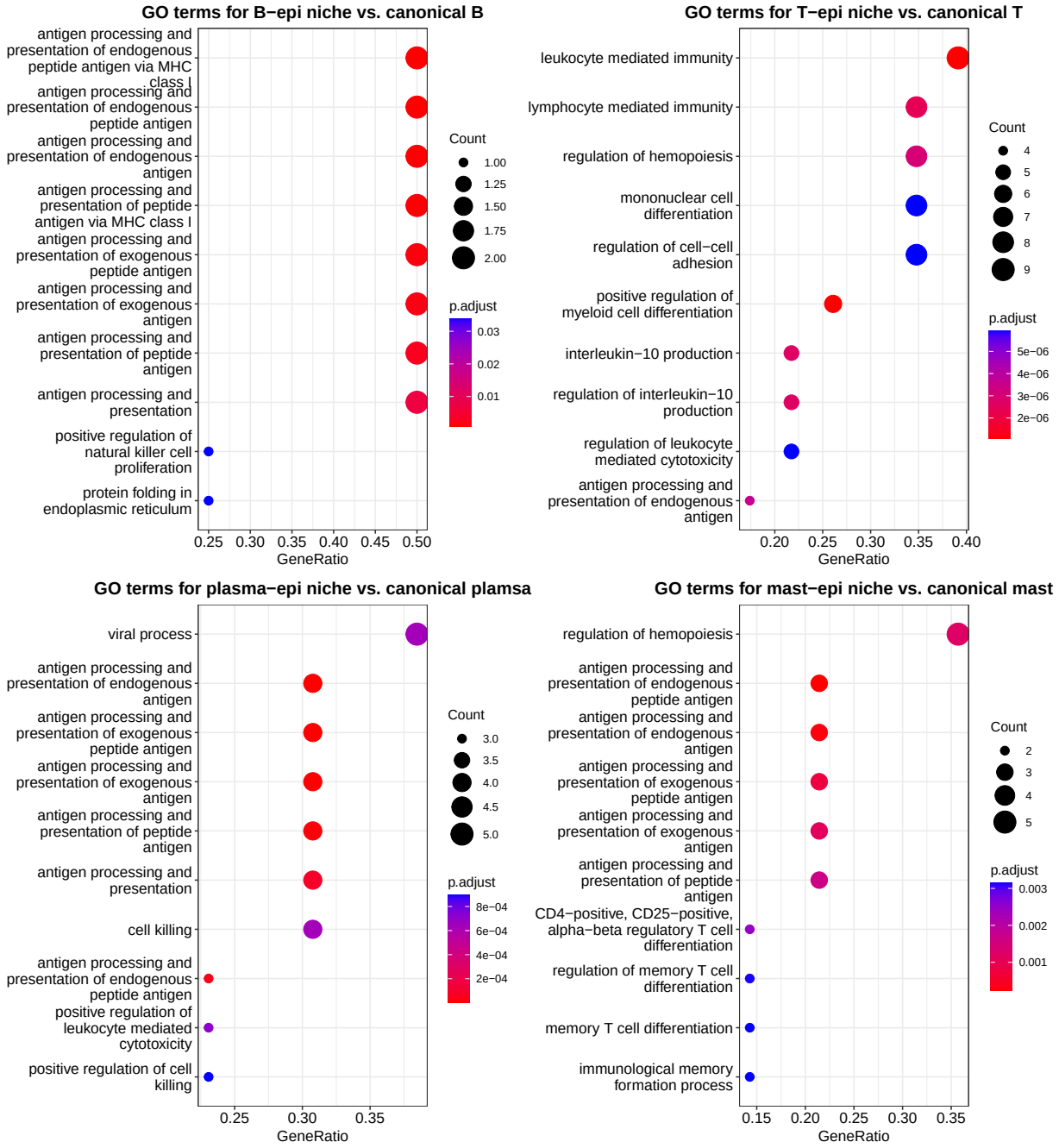


Figure 3.28: GO enrichment analysis of genes distinguishing epithelial-niche cell states from canonical immune cell types. Dot plots showing enriched Gene Ontology biological process terms for genes differentially enriched in epithelial-niche-associated B, T, plasma, and mast cell states compared with their corresponding canonical immune cell types. Each panel reports the top enriched GO terms for one comparison. The x-axis shows the gene ratio, dot size indicates the number of genes contributing to each GO term, and color represents the adjusted p-value. Across comparisons, epithelial-niche-associated cell states show enrichment for immune-related processes, including antigen processing and presentation, leukocyte-mediated immunity, lymphocyte-mediated immunity, and regulation of hematopoiesis, suggesting that these niche-associated populations retain immune functional programs while exhibiting distinct tissue-context-associated transcriptional features.

CHAPTER 4

Benchmarking of cell-type visualization pipelines with semi-synthetic atlas-scale scRNA-seq datasets

4.1 Introduction

Single-cell RNA sequencing (scRNA-seq) enables high-throughput transcriptome profiling at the single-cell level, significantly advancing biological research. The widespread adoption of this technology has led to the generation of atlas-scale scRNA-seq datasets, each comprising over 100,000 cells [65, 66]. In scRNA-seq analysis pipelines, two-dimensional (2D) visualization plays a crucial role in exploring relationships among cells—represented as high-dimensional gene expression vectors—and in generating scientific hypotheses, such as identifying gene expression patterns within specific cell types or subpopulations. Hence, many visualization methods have been developed, each offering specific advantages depending on data characteristics [10, 33–35]. However, debate continues regarding how accurately visualization captures the true underlying data structure [36], highlighting the need for rigorous and comprehensive evaluation of visualization methods.

Although several benchmark studies have evaluated visualization methods [21, 38, 64], three major limitations remain. First, most benchmarks overlook crucial preprocessing steps such as normalization and integration, which can substantially impact 2D visualization results. Normalization typically adjusts for differences in cell library sizes and stabilizes gene variance to become irrelevant to gene mean; Table 4.3 summarizes six commonly used normalization methods. Integration aligns cells from different batches to ensure comparability. Many integration methods—such as *Seurat-CCA* and *Seurat-RPCA* in Seurat v5 [50], *FastMNN* [119], and *Harmony* [120]—require input as the principal component scores of normalized data. Others, like *scVI* [121], operate on raw count data, while *LIGER* [122]

requires non-negative normalized data input. Despite this methodological diversity, previous benchmarks have not thoroughly examined the combined effects of normalization and integration on visualization. While Huang et al. [64] and Sun et al. [21] considered preprocessing steps, their studies did not directly evaluate 2D visualizations. Huang et al. relied on qualitative assessments without quantitative metrics, while Sun et al. focused on metrics related to cell clustering and trajectory inference based on outputs with dimensions greater than or equal to 2D. In contrast, our benchmark significantly expands preprocessing method coverage and introduces a comprehensive set of quantitative metrics specifically designed to assess 2D visualization quality (Figure 4.7).

Second, existing benchmark studies often rely on real datasets whose "ground truths" may be biased. For instance, cell-type labels used as ground truths are often computationally inferred by specific combinations of normalization, integration, and visualization methods, inherently favoring those methods in subsequent evaluations. For another instance, topology-related ground truths—such as inter-cell-type distances before 2D visualization—can be biased if defined after preprocessing. While obtaining unbiased ground truths for real data remains challenging, semi-synthetic data offer a practical alternative. Semi-synthetic data are generated by sampling from probability distributions fitted to real scRNA-seq data, preserving biological realism while providing known ground truths that are independent of preprocessing choices. In our benchmark study, we propose to use the state-of-the-art simulator scDesign3 [67] to generate comprehensive semi-synthetic datasets that closely mimic real scRNA-seq data while containing unbiased ground truths, enabling informative and fair benchmarking of methods.

Third, most existing benchmark studies have not emphasized data scale, despite the fact that atlas-scale scRNA-seq datasets pose distinct visualization challenges. These datasets often contain millions of cells and exhibit far greater diversity in cell types than those typically used in previous benchmarks. Visualization is essential for exploring atlas-scale data; however, the crowding issue—where cells are more likely to overlap in 2D visualization as the number of cells increases—complicates the visualization task [33, 123]. A common workaround, down-sampling, is suboptimal for atlas-scale data visualization, as it introduces

randomness and risks omitting rare cell types. Therefore, it is preferable to visualize atlas-scale data in full, without down-sampling. Our benchmark study will specifically address the challenges associated with visualizing atlas-scale scRNA-seq data. Leveraging the scDesign3 simulator, we can efficiently generate millions of semi-synthetic cells that reflect the complexity of atlas-scale scRNA-seq datasets. To address the three limitations outlined above, our benchmark study will evaluate 180 pipelines combining normalization, integration, and visualization methods. Here, a "pipeline" refers to a specific combination of normalization, integration, and visualization methods; we use the term "method combination" interchangeably throughout this manuscript. Although individual steps have been benchmarked in prior studies [15, 38, 61–63], simply combining the top-performing methods from each step does not guarantee optimal overall performance, as interaction effects between methods often exist. Furthermore, evaluation metrics designed for individual steps may not capture their combined impact on visualization quality. For instance, benchmarks of normalization methods often assess library size correction and variance stabilization, yet the downstream influence of these factors on visualization remains unclear. Therefore, a comprehensive evaluation of the full pipeline is essential to improve the reliability of scRNA-seq data visualization.

In addition to computational efficiency, we will evaluate each pipeline using eight additional metrics to assess its ability to: preserve between-cell-type distances and cell-type-specific variances; remove confounding factors (e.g., 2D visualization positions influenced by library size or zero-expression gene proportions); separate distinct cell types; avoid trajectory-like artifacts within cell types; and eliminate batch effects. In summary, our study aims to inform the construction of 2D visualization pipelines, deepen understanding of the combined effects of preprocessing steps, and promote the use of semi-synthetic data for more rigorous benchmarking in the single-cell field.

4.2 Reference datasets for benchmarking

This section describes the five reference scRNA-seq datasets that will be used to generate semi-synthetic datasets with known ground truths for benchmarking. All reference datasets were measured by unique molecular identifier (UMI)-based technologies, which eliminate

PCR amplification bias, and each includes multiple batches. A detailed summary of these datasets is provided in Table 4.1.

Table 4.1: Five real datasets used as references for semi-synthetic data generation. The semi-synthetic datasets were generated to match the cell-type and batch proportions of these real datasets, with the total number of cells randomly sampled between 0.5 and 1 million.

Dataset	Platform	# of cells	# of genes	# of batches	# of cell types	reference
Mouse cortex	10x, DroNc-seq, sci-rna-seq	8,780	28,692	3	8	Ding et al. [124]
Human prefrontal cortex	10x	11,464	33,178	8	7	Otero-Garcia et al. [125]
Human liver	CITE-seq	16,099	31,849	3	12	Jardine et al. [126]
Macaque retina fovea	10x	21,017	36,162	4	12	Peng et al. [127], Xu et al. [128]
IFALD liver	10x	32,397	80,288	19	18	Edgar et al. [90]

4.2.1 Mouse cortex dataset

This dataset consists of single cells sequenced from mouse cortex tissue using four different technologies—10x, DroNc-seq, Smart-seq2, and sci-RNA-seq—across two separate experiments conducted on different days [124]. To isolate one level of batch effects, we used data from the second experiment, where batch variation is limited to sequencing technology. Since our analysis focuses on UMI-based platforms, we excluded cells sequenced with Smart-seq2, resulting in three technologies treated as three batches. The final dataset includes 28,692 genes and 8,780 cells across 8 cell types. The raw count data is available from the Single Cell Portal under accession code SCP425.

4.2.2 Human prefrontal cortex dataset

This dataset consists of excitatory neurons sequenced from the prefrontal cortex of donors with and without Alzheimer's disease, using 10x 3' v2 and 10x 3' v3 technologies [125]. To eliminate condition effects, we included only cells from healthy donors, treating the eight healthy individuals as eight separate batches. The final dataset contains 33,178 genes and 11,464 cells across 7 cell types. The raw count data is available from the GEO under accession code GSE129308, and can also be accessed at https://cellxgene.cziscience.com/e/85c60876-7f35-40c5-a256-7808d84c6ba5.cxg/.

4.2.3 Human liver dataset

This dataset was generated by sequencing single cells from human liver tissue using CITE-seq [126]. It includes 31,849 genes and 16,099 cells across 12 cell types. Liver samples from three different donors are treated as three separate batches. The raw count data is available from the GEO under accession code GSE166895, and can also be accessed at https://cellxgene.cziscience.com/e/471647b3-04fe-4c76-8372-3264feb950e8.cxg/.

4.2.4 Macaque retina fovea dataset

This dataset was generated by sequencing single cells from the retinas of four macaque types [128]. Cells were collected from both the fovea and peripheral regions, but to focus on a single level of batch effects, we retained only the data from the fovea. The four macaque types are treated as four separate batches. The final dataset includes 36,162 genes and 21,017 cells across 12 cell types. The raw count data and cell type annotations are available from the GEO under accession code GSE118546 and can also be accessed at https://figshare.com/articles/dataset/scPSM/19306661.

4.2.5 IFALD (intestinal failure–associated liver disease) liver dataset

This dataset was generated by sequencing single cells from the human liver diseases [90]. Cells were collected from both the normal liver tissue and the liver tissue with intestinal failure–associated liver disease. The 19 healthy and diseased donors are treated as four separate batches. The final dataset includes 32,397 genes and 80,288 cells across 18 cell types. The raw count data and cell type annotations are available at https://cellxgene.cziscience.com/collections/ff69f0ee-fef6-4895-9f48-6c64a68c8289.

Using these five real datasets as references, we will employ the state-of-the-art simulator scDesign3 [67] to generate atlas-scale semi-synthetic data with specified ground truths,

including cell-type and batch labels, as well as cell-type-specific gene-wise means and variances.

4.3 Methods

4.3.1 Selection of methods to be benchmarked

For normalization, we evaluated a total of seven methods. Four of these were selected from Seurat (including SeuratWrappers) and Scanpy: *sctransform* [5], *log1pCP10k*, *log1pCPM*, and *log1pCPMedian* [49]. In addition, we included two methods—*log1pPF* and *PFlop1pPF*—which have demonstrated strong performance in previous studies [15, 129]. We also considered *Sanity* [130], a theory-backed probabilistic normalization approach.

For integration, we evaluated six methods implemented in Seurat or Scanpy: *Seurat-CCA* [48], *Seurat-RPCA*, *Harmony* [56], *FastMNN* [59], *LIGER* [60], and *scVI* [121].

For visualization, we applied five methods from Seurat and Scanpy: *Seurat-UMAP*, *scanpy-UMAP*, *BH-tSNE* [131], *FIt-SNE* [132], and *graphFA* [133]. Seurat and Scanpy implementations of UMAP were treated as distinct due to their differences in input construction—Seurat uses cell-cell cosine similarities, while Scanpy uses the nearest-neighbor graph also used for clustering [134]. We also included three additional widely used methods: *densMAP* [34], *denSNE* [34], and *PHATE* [35].

Figure 4.7 summarizes all methods considered in this study, and Table 4.2 provides the specific packages and versions used.

In theory, combining all normalization, integration, and visualization methods would result in 336 possible pipelines. However, two visualization methods (*PHATE* and *denSNE*) were excluded due to extended runtimes. Additionally, two integration methods—*scVI* and *LIGER*—have specific input constraints: *scVI* requires raw count data, while *LIGER* requires non-negative normalized values. For *LIGER*, we applied library-size normalization without log-transformation, $\frac{X_{gc}}{\sum_g X_{gc}}$, as recommended.

After accounting for runtime limitations and input constraints, we selected seven normalization methods (*log1pCP10k*, *log1pCPM*, *log1pCPMedian*, *log1pPF*, *PFlog1pPF*, *Sanity*

and *sctransform*), six integration methods (*Seurat-CCA*, *Seurat-RPCA*, *Harmony*, *FastMNN*, *LIGER*, and *scVI*), and six visualization methods (*Seurat-UMAP*, *scanpy-UMAP*, *densMAP*, *FIt-SNE*, *BH-tSNE*, and *graphFA*) for benchmarking. Due to the specific input requirements of *scVI* and *LIGER*, not all method combinations are compatible. As a result, these methods across the three pipeline steps yield a total of 180 valid pipelines.

4.3.2 Overview of methods benchmarked

4.3.2.1 Normalization methods

Residual-based normalization method

- *sctransform* [5] is a residual-based normalization method. It fits a regularized negative binomial regression for each gene, with the log-transformed cell library size as the only covariate, to calculate the Pearson residuals as the normalized data.

Bayesian normalization method

- *Sanity* [130] is a Bayesian normalization and denoising method. It models the observed UMI counts as noisy measurements of latent true gene expression levels, accounting for sampling noise and technical variation, and estimates posterior expression values that are used as the normalized data.

Size-factor-based normalization methods

- *log1pCP10k* scales raw counts to a fixed total library size of 10,000 counts per cell, and performs log 1 plus (log1p) transformation.
- *log1pCPM*, similar to *log1pCP10k*, scales raw counts to a fixed total library size of 1 million counts per cell, and performs log1p transformation.
- *log1pCPMedian* [49] scales raw counts to the median library size across all cells, followed by a log1p transformation.
- *log1pPF* [15] scales raw counts to the mean library size across all cells, followed by a log1p transformation.

Table 4.2: Summary of the software versions for each method

Method	Step	Software	Language
log1pCP10k	Normalization	Seurat (v5.1.0)	R
sctransform	Normalization	Seurat (v5.1.0)	R
log1pCPMedian	Normalization	scanpy (v1.10)	Python
log1pCPM	Normalization	Seurat (v5.1.0)	R
log1pPF	Normalization	https://github.com/pachterlab/BHGP_2022	Python
PFlog1pPF	Normalization	https://github.com/pachterlab/BHGP_2022	Python
Sanity	Normalization	https://github.com/jmbreda/Sanity	C++ command line application
Seurat-CCA	Integration	Seurat (v5.1.0)	R
Seurat-RPCA	Integration	Seurat (v5.1.0)	R
Harmony	Integration	SeuratWrappers (v0.3.5)	R
FastMNN	Integration	SeuratWrappers (v0.3.5)	R
LIGER	Integration	rliger (v2.1.0)	R
scVI	Integration	SeuratWrappers (v0.3.5)	R
Seurat-UMAP	Visualization	Seurat (v5.1.0)	R
scanpy-UMAP	Visualization	scanpy (v1.10)	Python
densMAP	Visualization	Seurat (v5.1.0)	R
denSNE	Visualization	densvis (v1.14.0)	R
FIT-SNE	Visualization	Seurat (v5.1.0)	R
BH-tSNE	Visualization	Seurat (v5.1.0)	R
graphFA	Visualization	scanpy (v1.10)	Python
PHATE	Visualization	phateR (1.0.7)	R

- *PFlog1pPF* [15, 129] refines the *log1pPF* method by adding an additional step after *log1pPF* that scales each cell's sum of the *log1pPF* value to the mean *log1pPF* value across all cells.

Table 4.3: Normalization methods evaluated in the visualization pipeline benchmark.

Method	Transformation
log1pCP10k ***[Seurat v5 default]*** log 1 plus count per 10k	$\log\left(\frac{X_{gc}}{\sum_g X_{gc}} \times 10^4 + 1\right)$
log1pCPM log 1 plus count per million	$\log\left(\frac{X_{gc}}{\sum_g X_{gc}} \times 10^6 + 1\right)$
log1pCPMedian **[Scanpy v1.10 default]** log 1 plus count per median cell library size	$\log\left(\frac{X_{gc}}{\sum_g X_{gc}} \times \text{median}_c\left\{\sum_g X_{gc}\right\} + 1\right)$
log1pPF log 1 plus proportional fitting, i.e., count per mean cell library size	$\log\left(\frac{X_{gc}}{\sum_g X_{gc}} \times \frac{\sum_c \sum_g X_{gc}}{\#\text{ of cells}} + 1\right)$
PFlog1pPF proportional fitting log 1 plus proportional fitting	$\frac{Y_{gc}}{\sum_g Y_{gc}} \times \frac{\sum_c \sum_g Y_{gc}}{\#\text{ of cells}}, \quad Y_{gc} = \log\left(\frac{X_{gc}}{\sum_g X_{gc}} \times \frac{\sum_c \sum_g X_{gc}}{\#\text{ of cells}} + 1\right).$
sctransform regularized negative binomial regression	$Z_{gc} = \frac{X_{gc} - \hat{\mu}_{gc}}{\sqrt{\hat{\mu}_{gc} + \hat{\mu}_{gc}^2/\hat{\theta}_g}},$ $X_{gc} \sim NB(\mu_{gc}, \theta_g),$ $\log(\mu_{gc}) = \beta_{0g} + \beta_{1g}\left(\sum_{g'} X_{g'c}\right).$ where $\hat{\mu}_{gc}$ and $\hat{\theta}_g$ are fitted from negative binomial regression per gene.

4.3.2.2 Integration methods

- *Harmony* [56] integrates batches in the principal component analysis (PCA) space through an iterative process. It first clusters cells pooled from different batches in

the principal component space and calculates cluster-specific correction terms based on batch-specific centroids. Using the correction terms, *Harmony* next updates cell embeddings by shifting each cell toward the centroid of its corresponding cluster. This clustering and embedding adjustment repeats iteratively until convergence. *Harmony* outputs the integrated principal component (PC) embeddings, which can be directly used for visualization.

- *Seurat-CCA* [48] first projects cells into a canonical correlation analysis (CCA) space, maximizing the correlation between two batches. Based on the L2-normalized canonical correlation vectors, it then identifies cross-batch mutual nearest neighbors (MNN) as "anchor" pairs. Next, it adjusts each cell's gene expression values using a weighted average of the k nearest anchors within the same batch. The weights are determined by the cell's distances to the k nearest anchors within the same batch and the distances between these anchors and the paired anchors in another batch. The closer within-batch anchors contribute more to the cell's gene expression value adjustment. *Seurat-CCA* outputs an integrated gene expression matrix, which is subsequently projected into low-dimensional space using PCA for visualization.

- *Seurat-RPCA* [48], compared with *Seurat-CCA*, integrates batches by first projecting cells into a low-dimensional space using reciprocal principal component analysis (RPCA) instead of CCA to improve computational efficiency. *Seurat-RPCA* outputs an integrated gene expression matrix, which is subsequently projected into low-dimensional space using PCA for visualization.

- *LIGER* [60] integrates batches by projecting cells into a joint low-dimensional space using integrative nonnegative matrix factorization and quantile normalizes the results as output. *LIGER* outputs the integrated low-dimensional embeddings, which can be directly used for visualization.

- *FastMNN* [59] integrates batches by first projecting cells using PCA. It then identifies MNN pairs—cells from different batches sharing similar gene expression values and calculates batch-specific local correction vectors to adjust the cell's PC embedding

based on these MNN pairs. *FastMNN* outputs the integrated PC embeddings, which can be directly used for visualization.

- *scVI* [121] integrates batches using a generative model based on variational autoencoders. It first projects cells into a shared low-dimensional latent space, simultaneously modeling biological variation, technical batch effects, and sequencing noise. The method iteratively learns batch-corrected latent embeddings through the optimization of a variational inference procedure. *scVI* outputs the integrated low-dimensional embeddings, which can be directly used for visualization.

4.3.2.3 Visualization methods

- *Seurat-UMAP* [48] implements the original UMAP algorithm, which constructs a weighted k-nearest neighbor graph to capture the local manifold structure and optimizes a low-dimensional embedding by minimizing the divergence between high- and low-dimensional representations [10]. In Seurat, UMAP is applied to low-dimensional embeddings using cosine distance as the metric.

- *scanpy-UMAP* [49] differs from Seurat's implementation by taking the cell-cell nearest-neighbor graph as input rather than relying directly on PCA's cosine distances as in *Seurat-UMAP*.

- *densMAP* [34] modifies the original UMAP algorithm to explicitly preserve local density structure. It estimates local densities in both high- and low-dimensional spaces and introduces an additional density-preserving loss term into UMAP's optimization. This loss term encourages embeddings to maintain relative density relationships from the original space.

- *BH-tSNE* [131] accelerates the original t-SNE algorithm, which computes cell-cell similarities between all pairs of points in both high- and low-dimensional spaces and minimizes their divergence via gradient descent [33]. To reduce the computational cost, *BH-tSNE* avoids explicitly calculating all pairwise similarities by grouping distant cells

into clusters using a spatial tree structure and approximating their contributions to the gradient.

- *FIt-SNE* [132] accelerates t-SNE by using interpolation-based techniques to approximate gradient calculations. It places data points onto an interpolation grid and efficiently computes repulsive forces using the fast Fourier transform, significantly reducing runtime while preserving embedding quality.

- *graphFA* [133] is a visualization algorithm that employs force-directed placement, treating cells as nodes and cell-cell similarities as edge weights. In this framework, nodes are modeled to be subject to attractive forces proportional to edge weights and repulsive forces inversely related to their weights from other nodes. Through iterative optimization, nodes' 2D coordinates converge to a state where attractive and repulsive forces are balanced. In Scanpy, the default visualization utilizes the nearest-neighbor graph, with edge weights derived from the UMAP algorithm.

4.3.3 Benchmarking procedure

1. **Quality control and semi-synthetic data generation**

   For each of the five reference datasets, we performed quality control based on three criteria: (1) within each batch, remove any cell type containing fewer than five cells; (2) exclude cell types with ambiguous annotations (e.g., NA or mixed types); and (3) remove genes expressed in fewer than ten cells.

   After quality control, we used Seurat to identify the top 3,000 highly variable genes (HVGs), following its default workflow for multi-batch scRNA-seq data in version 5. HVGs are selected for two reasons: (1) most scRNA-seq analysis methods operate on HVGs; and (2) generating semi-synthetic data is more scalable when limited to HVGs.

   Using the HVGs from the filtered reference dataset, we trained scDesign3 with a model that accounts for both cell types and batch effects. We then generated a semi-synthetic dataset from the trained model, following the original cell type and batch proportions, subject to sampling variability. The total number of cells in each semi-synthetic dataset

was randomly set between 0.5 and 1 million. Further details on the data generation process are provided in **Section 4.3.4 Semi-synthetic Data Generation**.

2. **Normalization, integration, and visualization**

   For each semi-synthetic dataset from Step 1, we applied all 180 pipelines, i.e., combinations of normalization, integration, and visualization methods. Normalization was performed separately within each batch. Integration was conducted using all HVGs. Depending on the integration method, scaling and PCA may be performed as part of the process.

   Two integration methods have specific requirements: *scVI* directly models count data and does not require normalization, scaling, or PCA. *LIGER* requires non-negative input and thus does not support scaling or PCA.

   All integration methods generated 50-dimensional cell embeddings, which were then used for two-dimensional visualization.

3. **Metric evaluation**

   After generating the 2D visualizations from all 180 pipelines, we evaluated them using the nine metrics described in **Section 4.3.5, Benchmarking Metrics**. Our evaluation focused on the *Pipeline performance comparison.* We linearly rescaled each of the eight evaluation metrics (excluding runtime and peak memory) to the $[0, 1]$ range within each dataset across all pipelines. For each pipeline, we computed the mean and variance of each rescaled metric across the five datasets. An overall performance score was then calculated as a weighted sum of the eight metrics, excluding runtime and peak memory. The eight metrics fall into six categories, and each category was assigned equal weight. Pipelines were ranked based on both their overall scores and individual metric performance.

4.3.4 Semi-synthetic data generation

4.3.4.1 Model fitting on reference data

Consider a reference scRNA-seq count matrix $\mathbf{X} = [X_{gc}] \in \mathbb{N}^{p \times n}$, where $g = 1, \ldots, p$ indexes genes and $c = 1, \ldots, n$ indexes cells. Each cell c has an associated cell-type label $t_c \in \{1, \ldots, K\}$ and a batch label $b_c \in \{1, \ldots, B\}$, where K is the total number of cell types and B is the total number of batches. Since all five reference datasets were generated using UMI-based platforms, we employ the negative binomial (NB) distribution [135] to model the count of gene g in cell c, conditional on its cell type and batch:

$$X_{gc} \mid t_c = k, b_c = b \sim \mathrm{NB}(\mu_{gkb}, \phi_{gk}), \quad c = 1, \ldots, n. \tag{4.1}$$

Here, the mean parameter μ_{gkb} depends on the cell type and batch, while the dispersion parameter ϕ_{gk} depends on the cell type only. To account for cell-type and batch effects, we assume the following log-linear relationships:

$$\log \mu_{gkb} = \zeta_g^{\mu} + \eta_{gk}^{\mu} + \xi_{gb}^{\mu},$$
$$\log \phi_{gk} = \zeta_g^{\phi} + \eta_{gk}^{\phi},$$

where ζ_g^{μ} and ζ_g^{ϕ} are intercepts for gene g; η_{gk}^{μ} and η_{gk}^{ϕ} capture the cell-type effects; and ξ_{gb}^{μ} represents the batch effect on gene expression means.

Equivalent to (4.1), the NB model can be expressed as a Gamma–Poisson mixture:

$$X_{gc} \mid t_c = k, b_c = b \sim \mathrm{Poisson}(\theta_{gkb}), \quad c = 1, \ldots, n,$$
$$\theta_{gkb} \sim \mathrm{Gamma}\left(\frac{1}{\phi_{gk}}, \frac{1}{\phi_{gk} \cdot \mu_{gkb}}\right).$$

This hierarchical model separates technical noise (modeled by the Poisson component) from biological variability (captured by the Gamma component) [136–139]. We use scDesign3, which flexibly estimates gene-specific parameters and accounts for gene–gene correlations,

to fit model (4.1) on each of the reference datasets. The estimated parameters are:

$$\hat{\mu}_{gkb} = \exp\left(\hat{\zeta}_g^{\mu} + \hat{\eta}_{gk}^{\mu} + \hat{\xi}_{gb}^{\mu}\right),$$
$$\hat{\phi}_{gk} = \exp\left(\hat{\zeta}_g^{\phi} + \hat{\eta}_{gk}^{\phi}\right),$$

for $g = 1, \ldots, p$, $k = 1, \ldots, K$, and $b = 1, \ldots, B$. scDesign3 also estimates gene–gene correlations via a Gaussian copula.

4.3.4.2 Semi-synthetic data generation from the fitted model

We generated semi-synthetic cells by sampling from the fitted model, preserving the original cell-type and batch proportions of the reference dataset. Each semi-synthetic dataset contained a randomly selected number of cells between 0.5 and 1 million. These datasets serve as inputs for benchmarking. Figure 4.3 illustrates the workflow of fitting model (4.1) using scDesign3 and generating a semi-synthetic dataset.

While one might consider using the parameter estimates $\hat{\mu}_{gkb}$ and $\hat{\phi}_{gk}$ as ground truths to benchmark directly on the reference datasets, this approach has major limitations. These estimates are derived from data that rely on prior cell-type annotations, which may bias evaluations toward the original annotation pipeline. Using such parameters for both modeling and evaluation introduces circularity and potential bias. In contrast, treating $\hat{\mu}_{gkb}$ and $\hat{\phi}_{gk}$ as true parameters within model (4.1) and using them to generate semi-synthetic data avoids this issue.

We believe that the semi-synthetic datasets generated from our model can reasonably represent clean scRNA-seq data, where each cell type is realistically homogeneous and unaffected by annotation noise or outliers. This is supported by results in Figure 4.5, where we show that model (4.1) fits the reference data well. Semi-synthetic data also offer two major advantages: (1) ground truths are known and independent of any existing pipeline, enabling unbiased evaluations; and (2) we can simulate atlas-scale datasets with millions of cells, allowing us to assess scalability. For these reasons, we used semi-synthetic data for benchmarking instead of the original reference datasets.

Specifically, for cell type k in batch b, we define the mean and variance of gene g (after removing technical noise) as:

$$\begin{aligned} m_g^{(k,b)} &:= \widehat{\mathbb{E}}[\theta_{gkb}] = \hat{\mu}_{gkb}, \\ v_g^{(k,b)} &:= \widehat{\mathrm{Var}}[\theta_{gkb}] = \hat{\phi}_{gk}\hat{\mu}_{gkb}^2. \end{aligned} \tag{4.2}$$

These values, $\left\{(m_g^{(k,b)}, v_g^{(k,b)}) : g = 1, \ldots, p; k = 1, \ldots, K; b = 1, \ldots, B\right\}$, are treated as the ground truths for metric definitions and performance evaluation in our benchmarking.

4.3.5 Benchmarking metrics

An effective 2D visualization should preserve the relative ordering of between-cell-type distances and within-cell-type variances, while ensuring that within-cell-type variance is not influenced by the number of cells per type. Since our semi-synthetic datasets contain homogeneous cells within each cell type, a high-quality visualization should clearly separate different cell types, avoid distortion caused by zero-expressed genes, and prevent artificial elongation of cell clusters. Additionally, the visualization should be free of residual technical artifacts, such as effects from cell library sizes or batch differences, which are expected to be mitigated by normalization and integration.

To assess these aspects of 2D visualization quality, we define nine benchmarking metrics, including runtime. For the eight non-runtime/memory metrics, higher values indicate better performance. Table 4.1 summarizes the notations used throughout this section.

Figure 4.1: Summary of notations used in metric definitions

Notation*	Explanation
n	total number of cells across cell types and batches
p	number of genes
K	number of cell types
B	number of batches
t_c	cell type label of cell c
b_c	batch label of cell c
$n^{(k,b)}$	number of cells in cell type k and batch b
$m_g^{(k,b)}$	ground-truth expression mean of gene g in cell type k and batch b
$v_g^{(k,b)}$	ground-truth expression variance of gene g in cell type k and batch b
$\mathbf{Y} \in \mathbb{R}^{n\times 2}$	2D visualization matrix for all cells

* Non-boldface notations denote scalar values.

4.3.5.1 Distance Preservation (dist. presv.)

This metric evaluates how well a 2D visualization preserves the relative similarities among cell types by comparing pairwise distances in the expression space with those in the 2D visualization.

The ground-truth between-cell-type distances are defined based on genes' ground-truth mean expression levels in each cell type and batch, defined in (4.2). For cell type $k = 1, \ldots, K$ in batch $b = 1, \ldots, B$, its ground-truth gene expression vector is

$$\mathbf{m}^{(k,b)} := \left(m_1^{(k,b)}, m_2^{(k,b)}, \ldots, m_p^{(k,b)}\right)^\mathsf{T}.$$

While one could compute Euclidean distances directly between these high-dimensional vectors, such comparisons are often unreliable due to the curse of dimensionality [140]. Small variations across many genes can compound, leading to unstable distances between cell types.

To mitigate this, we apply PCA to the set $\{\mathbf{m}^{(k,b)}\}_{k=1}^{K}$ within each batch b. Each vector is projected onto the top K PCs, yielding a PC score vector $\mathbf{pc}_{\mathbf{m}}^{(k,b)} \in \mathbb{R}^K$. The pairwise Euclidean distances between these PC score vectors define the ground-truth between-cell-type distances in batch b:

$$\mathbf{D}_{\text{true}}^{(b)} := \left[\text{dist}\left(\mathbf{pc}_{\mathbf{m}}^{(k_1,b)},\ \mathbf{pc}_{\mathbf{m}}^{(k_2,b)}\right)\right]_{k_1,k_2=1}^{K} \in \mathbb{R}^{K\times K}.$$

Let $\mathbf{Y} \in \mathbb{R}^{n\times 2}$ denote the 2D visualization matrix, where the i-th row gives the coordinates of cell i. For each cell type k in batch b, we compute its 2D centroid:

$$\tilde{\mathbf{m}}^{(k,b)} := \left(\frac{1}{n^{(k,b)}} \sum_{c:\ t_c=k,\ b_c=b} \mathbf{Y}_{c1},\ \frac{1}{n^{(k,b)}} \sum_{c:\ t_c=k,\ b_c=b} \mathbf{Y}_{c2}\right)^{\mathsf{T}},$$

where $n^{(k,b)} := \sum_{c=1}^{n} \mathbb{1}_{\{t_c=k,\ b_c=b\}}$ is the number of cells in cell type k and batch b. The pairwise distances between these centroids define the 2D between-cell-type distances in batch b:

$$\mathbf{D}_{\text{2D}}^{(b)} := \left[\text{dist}\left(\tilde{\mathbf{m}}^{(k_1,b)},\ \tilde{\mathbf{m}}^{(k_2,b)}\right)\right]_{k_1,k_2=1}^{K} \in \mathbb{R}^{K\times K}.$$

The *distance preservation* metric is then computed as the average Spearman correlation between the ground-truth and 2D between-cell-type distances in each batch across all B batches:

$$\text{Distance preservation} := \frac{1}{B}\sum_{b=1}^{B} \text{Spearman Corr}\left(\mathbf{D}_{\text{true}}^{(b)},\ \mathbf{D}_{\text{2D}}^{(b)}\right).$$

A higher value indicates stronger agreement between the 2D visualization and the original expression-space distances, reflecting better preservation of inter-cell-type structure.

4.3.5.2 Variance preservation (var. presv.)

This metric evaluates how well a 2D visualization preserves the biological variability within each cell type—that is, whether the spread of cells in the 2D space reflects the underlying gene expression variation for that cell type.

Based on the Poisson-Gamma model in Equation (4.2), we define the ground-truth

within-cell-type variance for cell type k in batch b as the sum of the gene-wise variances:

$$S_{\text{true}}^{(k,b)} := \sum_{g=1}^{p} v_g^{(k,b)}.$$

Let $\mathbf{S}_{\text{true}}^{(b)} := \left(S_{\text{true}}^{(1,b)}, \ldots, S_{\text{true}}^{(K,b)}\right)^{\mathsf{T}}$ denote the vector of ground-truth within-cell-type variances for all K cell types in batch b.

The corresponding 2D within-cell-type variance—interpreted as the visualized "size" of a cell type—is computed as the trace of the 2×2 covariance matrix of the 2D coordinates of the cell type. Specifically, let $\mathbf{Y}^{(k,b)} \in \mathbb{R}^{n^{(k,b)} \times 2}$ be the submatrix of $\mathbf{Y}$ containing the 2D coordinates of the $n^{(k,b)}$ cells in cell type k and batch b. Then,

$$S_{\text{2D}}^{(k,b)} := \text{Trace}\left(\text{Cov}(\mathbf{Y}^{(k,b)})\right). \tag{4.3}$$

Let $\mathbf{S}_{\text{2D}}^{(b)} = \left(S_{\text{2D}}^{(1,b)}, \ldots, S_{\text{2D}}^{(K,b)}\right)^{\mathsf{T}}$ denote the vector of visualized variances for all K cell types in batch b.

The *variance preservation* metric is defined as the average Spearman correlation between the ground-truth and 2D within-cell-type variances across all B batches:

$$\text{Variance preservation} := \frac{1}{B}\sum_{b=1}^{B} \text{Spearman Corr}\left(\mathbf{S}_{\text{true}}^{(b)},\ \mathbf{S}_{\text{2D}}^{(b)}\right).$$

A higher value indicates better preservation of the relative ordering of within-cell-type variances in the 2D visualization.

4.3.5.3 Variance-Cell-Number Irrelevance (var-cell-number irrel.)

This metric assesses whether the 2D within-cell-type variance (defined in Equation (4.3)) is independent of the number of cells in each cell type. Ideally, the spread of cells in a 2D visualization should reflect biological variation in gene expression, not the number of cells in the cell type. However, prior studies have shown that some 2D visualization methods—such as t-SNE and UMAP—can produce within-cell-type spreads that are influenced by the

number of cells [32, 141]. This artifact may lead to misinterpretations.

Let $\mathbf{N}^{(b)} = \left(n^{(1,b)}, \ldots, n^{(K,b)}\right)^{\mathsf{T}}$ denote the vector of cell counts for the K cell types in batch b. We compute the Spearman correlation between $\mathbf{S}_{\text{2D}}^{(b)}$ and $\mathbf{N}^{(b)}$ in each batch b and define the *variance-cell-number irrelevance* metric as:

$$\text{Variance-cell-number irrelevance} := 1 - \frac{1}{B}\sum_{b=1}^{B}\left|\text{Spearman Corr}\left(\mathbf{S}_{\text{2D}}^{(b)},\ \mathbf{N}^{(b)}\right)\right|, \tag{4.4}$$

where a higher value indicates that the visualized variance ("size") of a cell type is less influenced by the number of cells, which is desirable for unbiased interpretation.

4.3.5.4 Visualization-zero-proportion irrelevance (zero-prop. irrel.)

Previous work by Townes et al. [19] demonstrated that normalization can exaggerate differences between zero and nonzero expression values (Figure 4.4a), potentially skewing PCA and subsequently affecting downstream integration and visualization methods that rely on PCA output. To evaluate whether the proportion of zero-expressed genes in a cell influences its position in the 2D visualization, we use distance correlation [142]. Distance correlation quantifies the statistical dependence between two random vectors of possibly different dimensions. It ranges from 0 (independence) to 1 (strong dependence), providing a measure for assessing how strongly the number of zero-expressed genes in a cell impacts its 2D coordinates.

Let $\mathbf{z}^{(k,b)}$ denote the vector of zero-gene proportions for cells in cell type k and batch b, defined as:

$$\mathbf{z}^{(k,b)} := \left(\frac{\sum_{g=1}^{p}\mathbb{1}\{X_{gc}=0\}}{p} : t_c = k,\ b_c = b\right)^{\mathsf{T}} \in [0,1]^{n^{(k,b)}}.$$

Let $\mathbf{Y}^{(k,b)}$ be the submatrix of $\mathbf{Y}$ containing the 2D coordinates of the $n^{(k,b)}$ cells in cell type k and batch b.

Because the zero-expression rate can vary meaningfully across cell types, but is random and biologically irrelevant within each homogeneous semi-synthetic cell type, we assess this metric at the cell-type level. We compute the distance correlation between the zero-proportion vector and the 2D coordinates within each cell type and batch. The *visualization-*

zero-proportion irrelevance metric is then defined as:

$$\text{Visualization-zero-proportion irrelevance} := 1 - \frac{1}{KB}\sum_{b=1}^{B}\sum_{k=1}^{K}\text{Distance Corr}(\mathbf{z}^{(k,b)},\ \mathbf{Y}^{(k,b)}),$$

where a higher value indicates that a cell's 2D location is less influenced by the proportion of genes with zero expression, suggesting more reliable visualization unaffected by normalization artifacts.

4.3.5.5 Visualization-library-size irrelevance (lib-size irrel.)

Normalization methods aim to reduce the influence of varying library sizes across cells. However, Booeshaghi et al. [15] (Figure 4.4b) observed that some normalization approaches fail to fully remove this effect, leading to residual dependence between a cell's 2D coordinates and its library size after visualization. To evaluate whether a cell's total read count influences its 2D location, we use distance correlation [142], which quantifies statistical dependence between two variables, ranging from 0 (independence) to 1 (strong dependence).

Let $\mathbf{l}^{(k,b)}$ denote the vector of log-transformed library sizes for cells in cell type k and batch b, defined as:

$$\mathbf{l}^{(k,b)} := \left(\log\left(\sum_{g=1}^{p} X_{gc}\right) : t_c = k,\ b_c = b\right)^{\mathsf{T}} \in \mathbb{R}^{n^{(k,b)}}.$$

Let $\mathbf{Y}^{(k,b)}$ be the submatrix of $\mathbf{Y}$ containing the 2D coordinates of the $n^{(k,b)}$ cells in cell type k and batch b.

Because total read count (i.e., library size) can vary meaningfully across cell types but is not expected to carry biological relevance within a homogeneous semi-synthetic cell type, we assess this metric at the cell-type level. We compute the distance correlation between the log-scale library size vector and the 2D coordinates within each cell type and batch. The *visualization-library-size irrelevance* metric is then defined as:

$$\text{Visualization-library-size irrelevance} := 1 - \frac{1}{KB}\sum_{b=1}^{B}\sum_{k=1}^{K}\text{Distance Corr}(\mathbf{l}^{(k,b)},\ \mathbf{Y}^{(k,b)}),$$

where a higher value indicates that a cell's 2D location is less influenced by its library size, suggesting a more reliable visualization less affected by normalization artifacts.

4.3.5.6 Cell-type shape

We assume that semi-synthetic cells within the same cell type originate from a homogeneous distribution and should form a roughly cloud-like cluster. Therefore, in a reliable 2D visualization, these cell types should not appear elongated or trajectory-like. To assess the extent of elongation, let $\ell_1^{(k,b)}$ and $\ell_2^{(k,b)}$ denote the standard deviations along the first and second principal components, respectively, of the 2D coordinates for cells of type k in batch b. Intuitively, $\ell_1^{(k,b)}$ corresponds to the length of the 2D cell-type shape, and $\ell_2^{(k,b)}$ to its width.

Since the true 2D shape of each cell type is unknown and may vary naturally, we adopt a threshold-based approach to detect excessive elongation. Specifically, if the ratio $\ell_2^{(k,b)}/\ell_1^{(k,b)}$ falls below 1/5 (i.e., the length exceeds five times the width), the cell type is considered artificially elongated and assigned a score of 0. Otherwise, it receives a score of 1, indicating an acceptable shape.

We define the *cell-type shape* metric as the proportion of cell types across all batches that meet this threshold criterion:

$$\text{Cell-type shape} := \frac{1}{KB} \sum_{b=1}^{B} \sum_{k=1}^{K} \mathbb{1} \left\{ \frac{\ell_2^{(k,b)}}{\ell_1^{(k,b)}} \geqslant \frac{1}{5} \right\},$$

where a higher value indicates that more cell types display compact, non-elongated shapes, as expected in faithful 2D visualizations.

4.3.5.7 Cell-type separation (cell-type sep.)

This metric evaluates how well different cell types are separated in a 2D visualization using the silhouette score, which quantifies how similar a cell is to others of the same type compared to those of different types. For a given cell c of cell type k in batch b, we define:

$$a(c) := \frac{1}{n^{(k,b)} - 1} \sum_{c' \neq c,\ t_{c'}=k,\ b_{c'}=b} \text{dist}(\mathbf{Y}_{c\cdot},\ \mathbf{Y}_{c'\cdot}),$$

$$b(c) := \min_{k' \neq k} \frac{1}{n^{(k',b)}} \sum_{t_{c'}=k',\ b_{c'}=b} \mathrm{dist}(\mathbf{Y}_{c\cdot},\ \mathbf{Y}_{c'\cdot}),$$

where $\mathbf{Y}_{c\cdot}$ is the 2D coordinate of cell c in the visualization. Here, $a(c)$ measures the average distance between cell c and other cells of the same type and batch, while $b(c)$ measures the average distance from cell c to the closest other cell type (within the same batch).

The silhouette score for cell c is defined as:

$$s(c) := \frac{b(c) - a(c)}{\max\{a(c), b(c)\}}.$$

We then define the *cell-type separation* metric as the average silhouette score across all cell types and batches:

$$\text{Cell-type separation} := \frac{1}{KB} \sum_{b=1}^{B} \sum_{k=1}^{K} \frac{1}{n^{(k,b)}} \sum_{c:\ t_c=k,\ b_c=b} s(c),$$

where a higher value indicates that cells of different types are more distinctly separated in the 2D visualization, aiding interpretability and usability.

4.3.5.8 Batch mixture (batch mix.)

Batch effects are systematic variations caused by technical or experimental factors—such as differences between individuals or sequencing platforms—that can obscure meaningful biological patterns in 2D visualizations. A reliable integration method should mitigate these effects, ensuring that cells of the same type appear well mixed across batches.

To assess batch mixing quality, we use the Local Inverse Simpson's Index (LISI) [120], a metric designed to quantify local diversity. Specifically, we compute the LISI score $\mathrm{lisi}(c)$ for each cell c, which reflects how well its local neighborhood includes cells from multiple batches.

We then define the *batch mixture* metric as the average LISI score across all cell types and batches:

$$\text{Batch mixture} := \frac{1}{KB} \sum_{b=1}^{B} \sum_{k=1}^{K} \frac{1}{n^{(k,b)}} \sum_{c:\ t_c=k,\ b_c=b} \mathrm{lisi}(c),$$

where a higher value indicates stronger mixing of cells from different batches within the same cell type, suggesting that batch effects have been effectively removed.

4.3.5.9 Runtime and peak memory

We record the runtime and memory used of each pipeline on each dataset as a reference for computational efficiency. While runtime and memory may vary due to implementation details and programming language (e.g., R vs. Python), it is reported for completeness and is not included in the overall performance scoring.

Remark: Calculating pairwise distances across millions of cells is computationally demanding. For the four metrics that rely on such computations—*zero-prop. irrelevance*, *lib-size irrelevance*, *cell-type separation*, and *batch mixture*—we randomly sample 10,000 cells from each semi-synthetic dataset, compute the metric 100 times, and report the average as an approximation.

4.3.5.10 Metric Aggregation

To aggregate the metrics into an overall pipeline evaluation, we first rescale each of the eight non-runtime/memory metrics to the $[0, 1]$ range within each dataset across pipelines. Let x_{ij}^m denote the value of metric m for pipeline j on dataset i. The rescaled metric is defined as:

$$\tilde{x}_{ij}^m = \frac{x_{ij}^m - \min_j(x_{ij}^m)}{\max_j(x_{ij}^m) - \min_j(x_{ij}^m)}.$$

We compute the average and variance of each rescaled metric across the 11 datasets:

$$\text{mean}_j^m = \frac{1}{11}\sum_{i=1}^{11} \tilde{x}_{ij}^m,$$
$$\text{var}_j^m = \frac{1}{10}\sum_{i=1}^{11} (\tilde{x}_{ij}^m - \text{mean}_j^m)^2.$$

The overall performance score and its variance are computed as:

$$\text{Overall score}_j = \sum_{m=1}^{8} w_m \cdot \text{mean}_j^m,$$
$$\text{Overall variance}_j = \sum_{m=1}^{8} w_m^2 \cdot \text{var}_j^m.$$

Because some metrics are conceptually related—e.g., *variance preservation* and *variance-cell-number irrelevance* both assess variance fidelity, and *zero-prop. irrel.* and *lib-size irrel.* both assess robustness to normalization artifacts—we group them into two joint categories. The remaining four metrics form individual categories. We assign equal weight to each of the six categories. Thus, metrics within a two-metric category receive a weight of $w = 1/12$, while those in a single-metric category receive a weight of $1/6$.

4.4 Benchmark results

The benchmark (Figures 4.6 and 4.8) revealed large performance differences across the evaluated visualization pipelines, indicating that visualization quality depends strongly on the combined choices of normalization, integration, and visualization method rather than on any single step alone. The highest-ranked pipelines achieved consistently strong overall scores, whereas the lowest-ranked pipelines showed substantially reduced overall performance and greater instability across metrics.

Among the top-performing pipelines, most used *sctransform* normalization, suggesting that this normalization strategy provided a strong foundation for downstream integration and visualization in this benchmark. The top-ranked pipeline combined *sctransform*, *fastMNN*, and *graphFA*, achieving the highest overall score. Other highly ranked pipelines paired *sctransform* with *Harmony* or *SeuratCCA*, followed by visualization methods including *scanpyUMAP*, *SeuratUMAP*, *densMAP*, and *graphFA*. This pattern suggests that strong performance was not driven by a single integration or visualization method, but by specific compatible combinations of preprocessing, integration, and embedding construction.

The benchmark also showed that the best-performing pipelines balanced multiple aspects

of visualization quality. These pipelines generally achieved favorable metric-score averages across structure preservation, cell-type separation, library-size independence, zero-proportion independence, and batch-mixing-related metrics. In contrast, pipelines near the bottom of the ranking showed weaker overall scores and more uneven metric profiles, indicating that poor overall visualization performance often reflected simultaneous degradation across several evaluation criteria rather than failure in only one metric.

Notably, lower-ranked pipelines were not uniformly poor across all evaluation dimensions. Several *scVI*-based pipelines appeared near the bottom according to the overall score, but some of them achieved top-performing results for specific individual metrics, for example, cell-type distance preservation and variance preservation. This indicates that *scVI*-based embeddings can preserve or optimize selected aspects of the data structure, even when their aggregate performance across all benchmark metrics is limited. Therefore, these results further emphasize the importance of multi-metric evaluation: ranking pipelines by a single metric may favor methods that perform well for one criterion while overlooking broader weaknesses in the final visualization.

Runtime and memory usage also varied substantially across pipelines and did not perfectly align with visualization quality. Some high-performing pipelines required relatively high computational time or memory (e.g., pipelines with graphFA took a long time to run and Sanity used memory intensively), whereas some lower-ranked pipelines were computationally cheaper but produced weaker overall embeddings. Considering overall score together with runtime and peak memory, and metric standard deviation, the benchmark supports *sctransform + Seurat-CCA + scanpyUMAP* as the final recommended pipeline. This pipeline ranked among the top overall performers while requiring less runtime and memory than the highest-scoring *sctransform + fastMNN + graphFA* pipeline, and is more stable across different datasets compared with *sctransform + Harmony + graphFA*. This provides a more favorable balance between visualization quality, computational efficiency, and stability.

Overall, these results demonstrate that visualization benchmarking should be conducted at the pipeline level. The strong performance of several *sctransform*-based pipelines, together with the metric-specific strengths but weaker aggregate performance of several *scVI*-based

pipelines, suggests that visualization quality is highly sensitive to interactions among preprocessing, integration, and embedding methods. This supports the need for systematic, multi-metric benchmarking frameworks that evaluate complete workflows and identify practical recommended pipelines by jointly considering overall quality, runtime, and memory usage.

4.5 Discussion

This benchmark provides an end-to-end evaluation of scRNA-seq visualization pipelines by jointly considering normalization, integration, and visualization methods. Rather than evaluating each analytical step in isolation, our results show that the final visualization quality depends on interactions among all components of the pipeline. A method that performs well as an individual preprocessing or integration step does not necessarily lead to the best two-dimensional embedding when combined with downstream methods. This finding highlights the importance of benchmarking complete workflows, especially for visualization tasks where the final output is shaped by accumulated decisions across the full analysis pipeline.

Across the evaluated pipelines, *sctransform*-based workflows were consistently among the strongest performers, suggesting that residual-based normalization can provide a stable input for downstream integration and visualization. However, the best-performing pipelines were not determined by normalization alone. Specific combinations of normalization, integration, and visualization methods produced substantially different results, emphasizing that pipeline-level compatibility is essential. In particular, the final recommended pipeline, *sctransform* + *SeuratCCA* + *scanpyUMAP*, achieved a favorable balance among overall visualization quality, runtime, memory usage, and stability across datasets. Although *sctransform* + *fastMNN* + *graphFA* achieved the highest overall score, its computational cost was higher, especially because *graphFA*-based pipelines required longer runtime. Similarly, pipelines involving *Sanity* tended to be memory intensive. These results suggest that practical pipeline recommendations should not rely only on accuracy-related metrics, but should also account for computational feasibility and cross-dataset stability.

Our results also demonstrate the value of multi-metric benchmarking. Several pipelines

with lower overall rankings still performed strongly for specific metrics. For example, some *scVI*-based pipelines achieved top performance for selected evaluation criteria despite having lower aggregate scores. This indicates that different pipelines may preserve different aspects of the data, such as local structure, cell-type separation, batch mixing, or independence from library-size-related artifacts. Therefore, a single metric is unlikely to fully capture visualization quality. A comprehensive benchmark should evaluate multiple complementary properties and summarize them in a way that reflects the intended use of the visualization.

One limitation of the current benchmark is that it uses a fixed set of parameter values for each method across datasets. Although this design improves comparability and reproducibility, it may underestimate the performance of methods whose results are sensitive to dataset-specific parameter choices. In real applications, optimal parameters may depend on the strength of batch effects, cell-type composition, sample size, sparsity, and the biological structure of interest. Therefore, future benchmarking could be extended from a fixed-parameter comparison to a data-adaptive framework, where parameters are selected separately for each dataset.

A natural direction is to incorporate data-adaptive optimization tools into the benchmarking framework. For the integration step, IntegrateRigor could be used to select integration-specific genes and tune integration hyperparameters in an annotation-free manner, allowing each dataset to be integrated under settings that better balance batch mixing and cell-identity preservation. For the visualization step, scDEED [143] could be incorporated to select visualization hyperparameters by evaluating whether low-dimensional embeddings preserve meaningful high-dimensional structure while avoiding misleading visual distortions. Combining IntegrateRigor for integration optimization and scDEED for visualization hyperparameter selection would allow the benchmark to evaluate not only fixed pipelines, but also adaptive pipelines that are optimized for each dataset.

Such a data-adaptive benchmark would better reflect practical single-cell analysis, where analysts rarely rely on a single universal parameter setting for all datasets. It would also help distinguish between methods that perform well only under carefully chosen fixed settings and methods that remain reliable after principled, dataset-specific optimization. More broadly,

extending the benchmark in this direction could provide a framework for recommending complete visualization workflows that are tailored to the data while remaining objective, reproducible, and scalable.

4.6 Data availability

The mouse cortex dataset is available from the Single Cell Portal under accession code SCP425. The human prefrontal cortex dataset is available from GEO under accession code GSE129308. The human liver dataset is accessible from GEO under accession code GSE166895. The IFALD dataset can be accessed at https://cellxgene.cziscience.com/collections/ff69f0ee-fef6-4895-9f48-6c64a68c8289. The macaque retina fovea dataset can be accessed from GEO under accession code GSE118546.

Code availability

All scripts used to reproduce this study are publicly available on GitHub: `https://github.com/zhiqianZ/vizbench-r.git`. The repository contains code for dataset download using the Omnibenchmark [144], quality control, semi-synthetic data simulation, pipeline execution, and metric calculation. The specific software versions used for each method are listed in Table 4.2. The repository will be updated to reflect the finalized benchmark results.

4.7 Acknowledgements

This chapter is based on my joint work with Qingyang Wang, and my Ph.D. advisor, Dr. Jingyi Jessica Li. We appreciate the comments and feedback from the members of the Junction of Statistics and Biology at UCLA (`http://jsb.ucla.edu`).

4.8 Figures

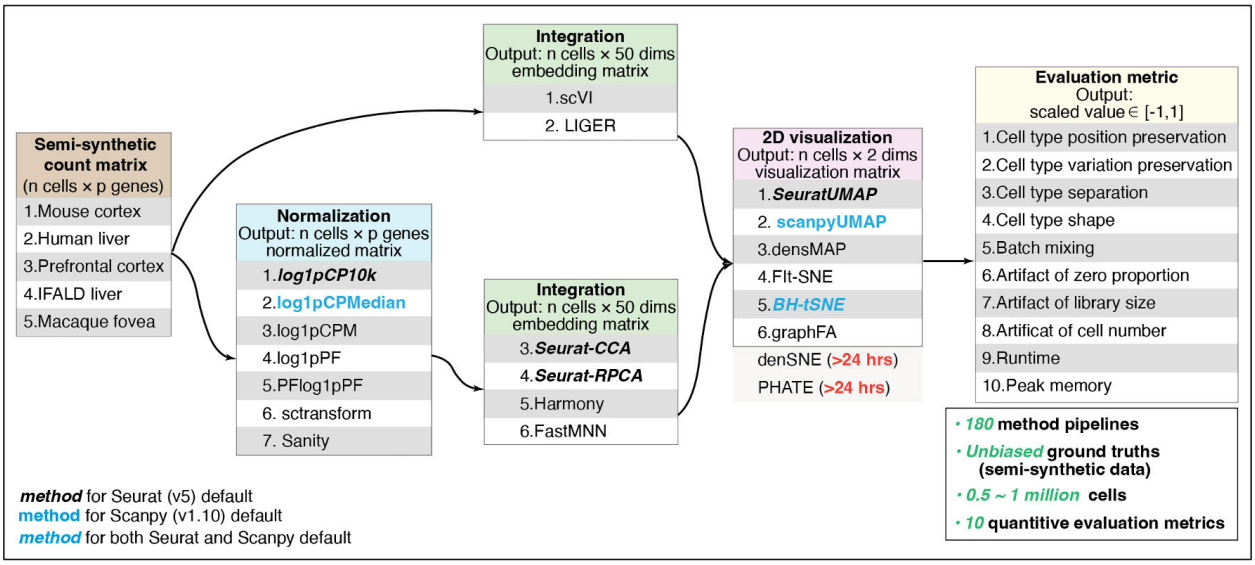


Figure 4.2: Benchmarking framework for evaluating single-cell visualization pipelines. The benchmarking workflow evaluates end-to-end visualization pipelines using semi-synthetic atlas-scale scRNA-seq count matrices with known ground truths. Each pipeline consists of a normalization method, an optional integration method that produces a 50-dimensional embedding, and a 2D visualization method. In total, 180 method combinations were evaluated across five semi-synthetic datasets, ranging from 0.5 to 1 million cells. Pipeline performance was assessed using ten quantitative metrics

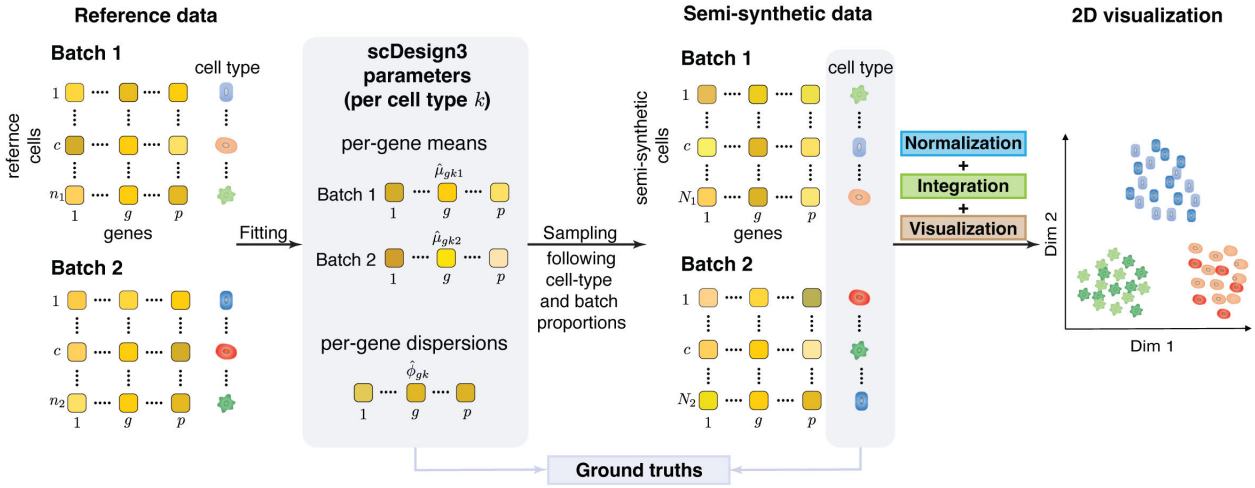


Figure 4.3: Workflow of semi-synthetic data generation. This diagram illustrates how we generate a semi-synthetic dataset from a reference dataset for benchmarking. scDesign3 first estimates gene-specific mean and dispersion parameters, conditional on cell type and batch. Then, in combination with randomly sampled cell-type and batch labels, these parameters are used by scDesign3 to simulate atlas-scale semi-synthetic data. The randomly sampled cell-type and batch labels follow the original proportions from the reference data, subject to sampling variability. The estimated parameters and labels serve as the ground truths for evaluating pipeline performance.

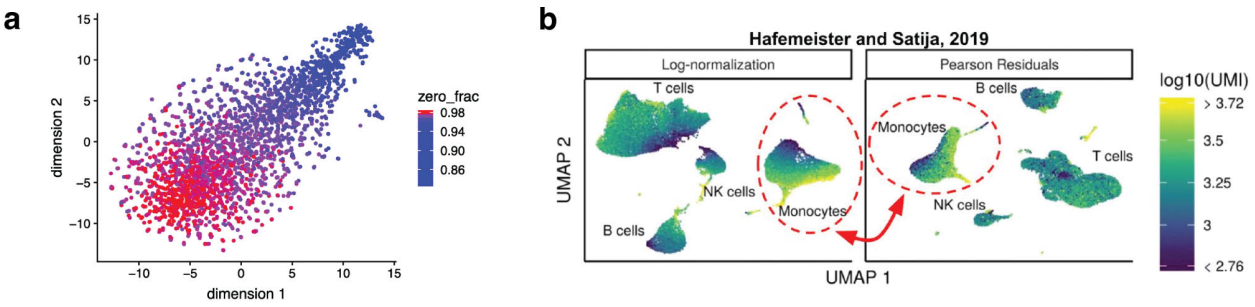


Figure 4.4: Examples of misleading patterns in 2D visualizations. a. Adapted from Figure 3d in Townes et al. [19], this panel shows human PBMC monocytes, where 2D locations are visibly influenced by the fraction of zero counts after applying *log1pCPM* normalization followed by t-SNE. The blue-to-red color scale indicates the fraction of zeros per cell, illustrating how preprocessing steps can distort biological structure in visualizations. **b.** Adapted from Figure 1a in Booeshaghi et al. [15], this panel shows cells whose 2D locations are affected by library size after normalization with either *log1pCP10k* (left) or *sctransform* (right), followed by UMAP. The viridis color scale denotes log-transformed library size, highlighting how normalization artifacts can create misleading visualizations.

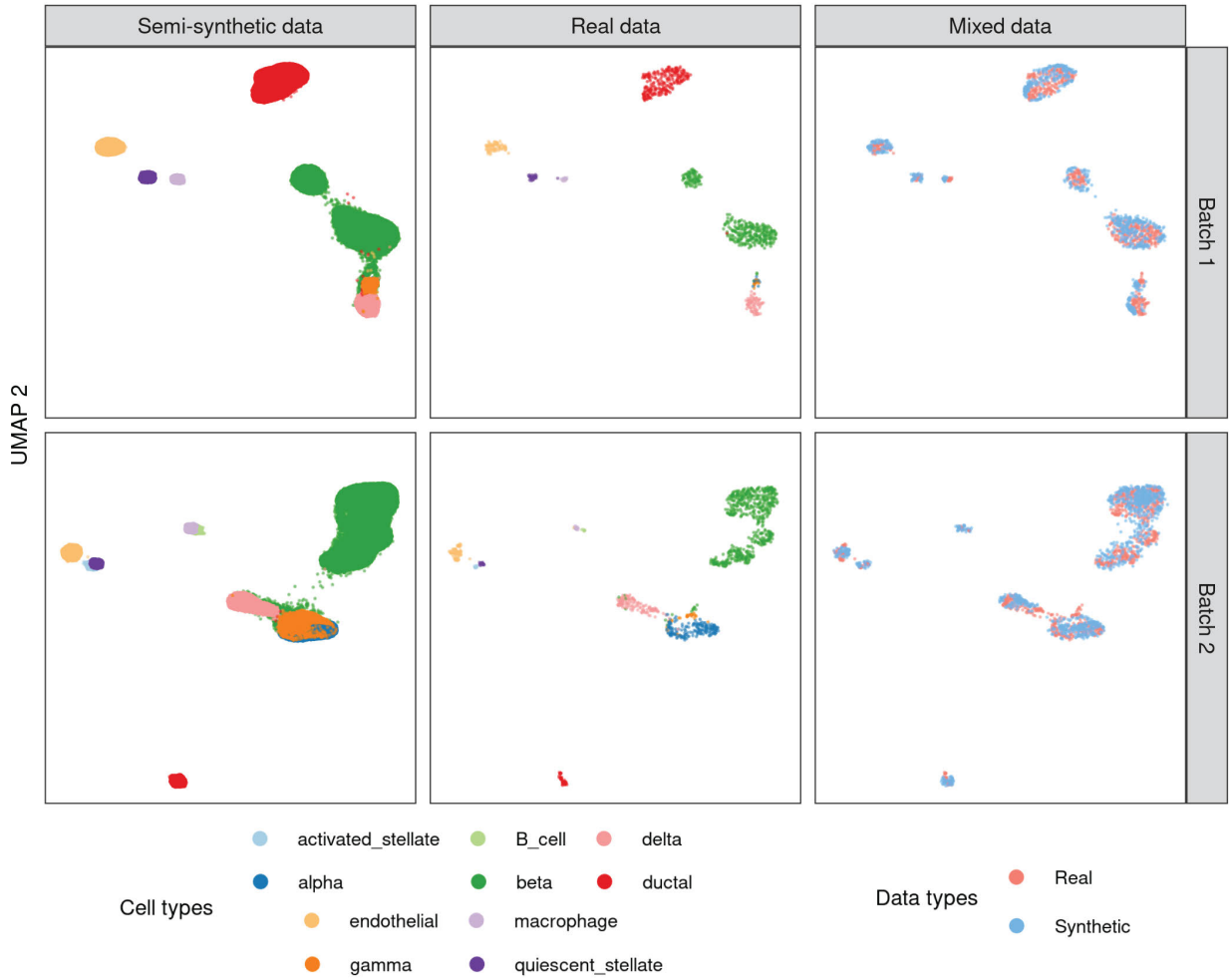


Figure 4.5: *Seurat-UMAP* visualization of semi-synthetic cells from the pilot mouse pancreas dataset, compared to their real counterparts. The first two columns show batch-wise UMAP visualizations of semi-synthetic and real cells, each projected into the same embedding space. The third column overlays both datasets, with semi-synthetic cells downsampled to match the number of real cells, enabling a direct visual comparison of their distributions.

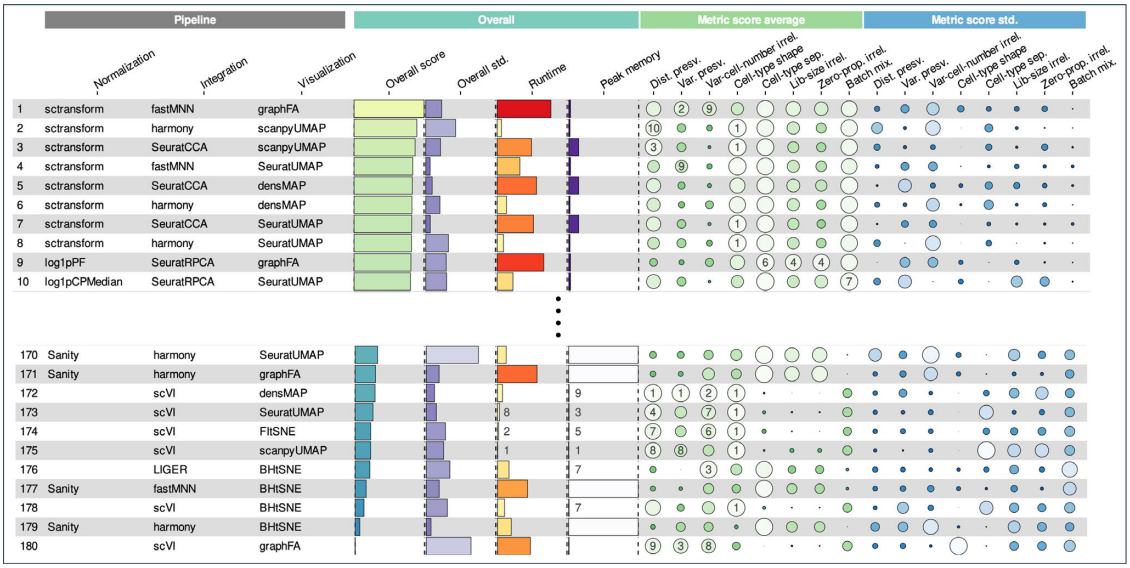


Figure 4.6: Ranking of visualization pipelines across normalization, integration, and visualization choices. Pipelines were ranked by the overall score, which summarizes performance across the evaluated visualization metrics. Each row represents one complete pipeline, defined by the combination of normalization, integration, and visualization methods. The left panels show the overall score, overall score standard deviation, runtime, and peak memory usage. The right panels summarize the average metric scores and their standard deviations across datasets, with larger circles indicating higher ranks or stronger relative performance for each metric. Only the top and bottom 10 pipelines are shown here

4.9 Supplementary figures

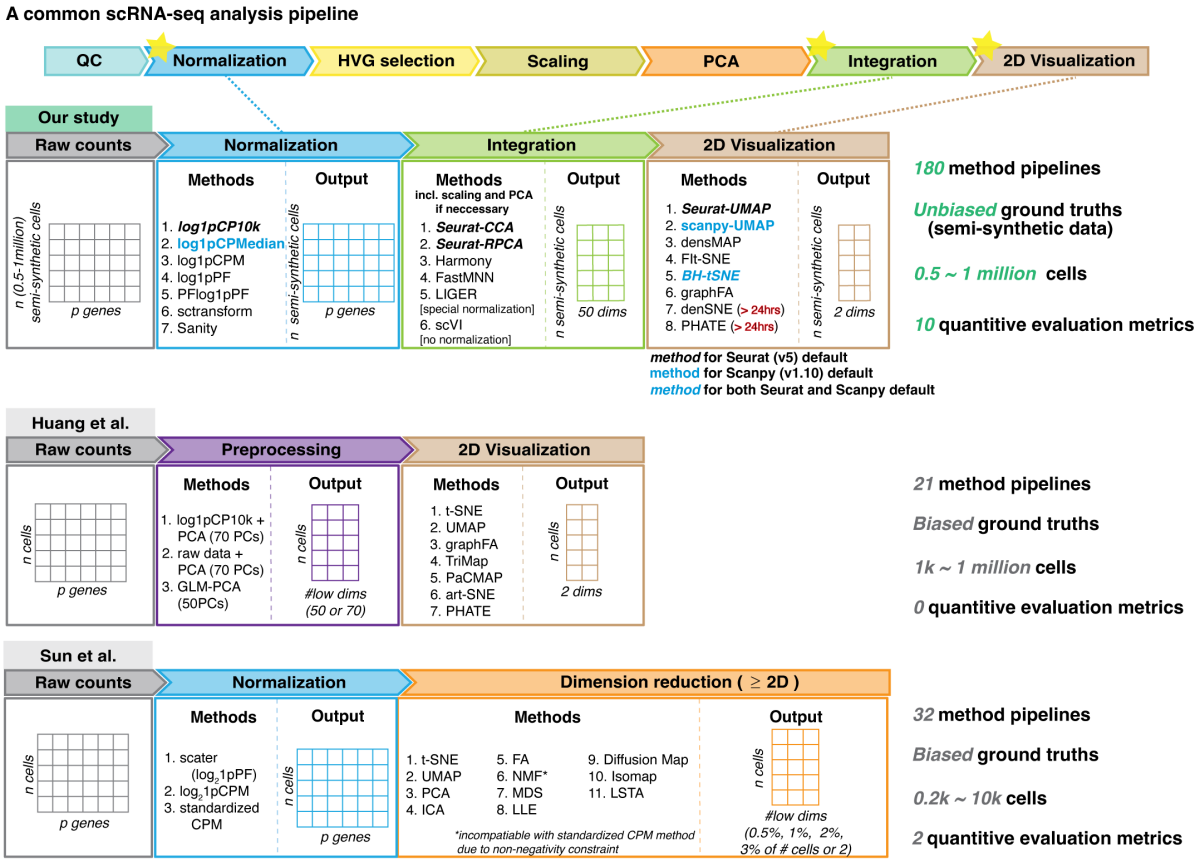


Figure 4.7: Comparison of our benchmark study with existing studies. This figure compares the preprocessing steps and methods used in our benchmark study with those in two prior studies—Huang et al. [1] and Sun et al. [2]. Our analysis focuses on the joint effects of normalization, integration, and visualization methods on 2D visualization, evaluating a total of 180 pipelines. We highlight default methods used by popular scRNA-seq analysis tools, Seurat and Scanpy, which are distinguished using different font styles. Our study leverages semi-synthetic scRNA-seq datasets with unbiased ground truths, allowing for evaluation using nine quantitative metrics. In contrast, Huang et al. evaluated seven visualization methods and examined the effects of three combinations of normalization and dimensionality reduction (shown under the Preprocessing step in the figure). Notably, the output dimensions vary across dimensionality reduction methods (e.g., PCA vs. GLM-PCA), potentially limiting result comparability. Moreover, their assessment of preprocessing impacts was based solely on visual inspection, without quantitative evaluation. Sun et al. explored 11 dimensionality reduction methods combined with three normalization approaches, focusing on low-dimensional outputs beyond 2D. As a result, their study does not provide a direct assessment of 2D visualization quality. For clustering, they assessed the effects of normalization by computing normalized mutual information between known cell types and clustering results across various algorithms applied to the low-dimensional outputs. For trajectory inference, they evaluated normalization effects using Kendall correlation between known pseudotime and pseudotime inferred from the low-dimensional outputs. Both existing studies relied on cell-type labels in real datasets as ground truths, without addressing potential biases. Moreover, they examined fewer method combinations and did not account for the integration step.

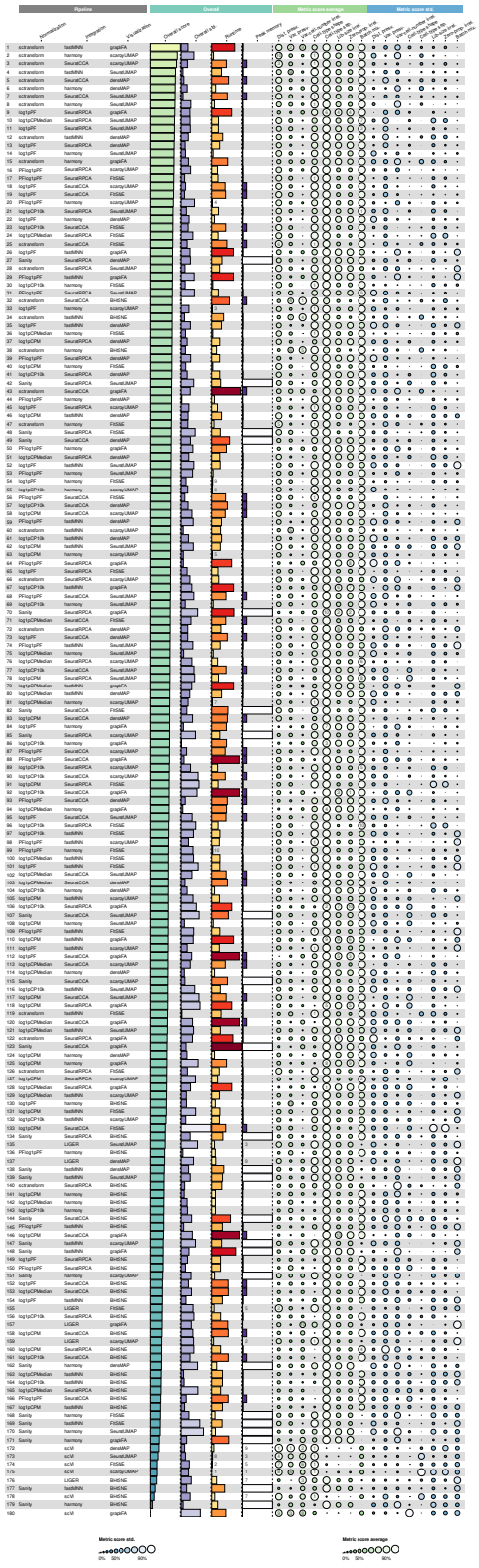


Figure 4.8: Complete ranking of visualization pipelines across normalization, integration, and visualization choices. Pipelines were ranked by the overall score, which summarizes performance across the evaluated visualization metrics. Each row represents one complete pipeline, defined by the combination of normalization, integration, and visualization methods. The left panels show the overall score, overall score standard deviation, runtime, and peak memory usage. The right panels summarize the average metric scores and their standard deviations across datasets, with larger circles indicating higher ranks or stronger relative performance for each metric.

CHAPTER 5

Summary and future directions

This dissertation improves the reliability and reproducibility of single-cell and spatial omics analysis through three studies. First, it examines how human subjectivity affects scRNA-seq cell-type annotation, showing that pipeline choices, marker-gene selection, prior experience, and confirmation bias can influence annotation outcomes. Second, it introduces IntegrateRigor, a method-agnostic framework that uses annotation-free gene selection and integration metrics to objectively optimize integration methods and hyperparameters in a dataset-specific manner. Third, it benchmarks atlas-scale scRNA-seq visualization pipelines by evaluating combinations of normalization, integration, and visualization methods using semi-synthetic data with known ground truths.

Together, these studies focus on three sources of analytical variability: human interpretation, integration quality, and pipeline-level visualization performance. At the same time, they represent only part of the broader challenge of achieving rigor in single-cell analysis. Important open problems remain in experimental design and the control of confounding between biological and technical variation [84, 145, 146], uncertainty quantification across multi-step pipelines, reference-atlas construction and label transfer [12, 147], and multimodal integration [23]. For example, while this dissertation shows that human subjectivity can affect cell-type annotation, it does not quantify annotation uncertainty in a label-free manner. Such uncertainty may accumulate throughout the analysis pipeline through choices in quality control, normalization, feature selection, clustering, integration, visualization, and label transfer. Quantifying this accumulated uncertainty remains an important direction for rigorous single-cell analysis, because biological discoveries from these pipelines depend on a sequence of methodological choices that are often heuristic, dataset-dependent, and incompletely reported. Another important direction is to develop principled strategies for

propagating annotation uncertainty into downstream cell-type-dependent analyses, so that biological discoveries can account for label ambiguity rather than relying on fixed labels as if they were ground truth.

The need for objective and rigorous analysis becomes increasingly urgent as single-cell and spatial datasets continue to grow in scale, complexity, and modality, and as the number of available computational methods rapidly expands. Rigorous analysis requires frameworks that evaluate not only whether a method performs well on a benchmark, but also whether its assumptions, decisions, and outputs remain stable across datasets, technologies, analysts, and biological contexts. Overall, this dissertation shows that improving single-cell analysis requires not only new algorithms, but also explicit evaluation of human decisions, data-adaptive optimization of analytical choices, and benchmark frameworks that assess complete pipelines rather than isolated steps. Building on these directions, future work should further develop uncertainty-aware, data-adaptive, and pipeline-level frameworks that make single-cell and spatial analyses more reproducible, interpretable, and reliable across biological settings.

5.1 How does human subjectivity affect the reliability of single-cell RNA-seq cell-type annotation?

Chapter 2 presents a human-subject study that systematically evaluated how analysts' backgrounds and subjective analytical choices affect scRNA-seq cell-type annotation. Even when participants followed the same Seurat V5-based pipeline and were restricted to the same available functions, their annotations still showed substantial variability. Participants were generally able to identify major cell types, but they often struggled to distinguish closely related subtypes. The study also showed that consensus across multiple analysts can improve annotation accuracy, suggesting that individual-level subjectivity is a major source of variability in manual annotation.

The results further showed that relevant analytical experience matters. Participants with prior single-cell analysis experience achieved higher annotation accuracy, especially in more

challenging datasets where the true cell-type composition did not fully match the provided reference list. Parameter tuning improved clustering accuracy, but this improvement was largely reduced after cell-type labels were assigned to clusters, indicating that the label-assignment step introduced additional subjectivity. The number of reference marker genes used was negatively associated with annotation accuracy, suggesting that treating all marker genes as equally informative may make the annotation process more difficult rather than more reliable.

This chapter also identified confirmation bias as an important factor in annotation errors. Participants performed worse when datasets did not contain all expected cell types, and their focused search for rare cell types increased false positive annotations, especially when the target rare types were absent. Although Biomni reduced some forms of manual effort, it underperformed most human participants and showed inconsistent behavior, especially for rare cell types. Future work should validate these findings in larger and more diverse analyst populations, including professional single-cell researchers, and should develop more standardized and evidence-weighted marker-gene guidelines. More broadly, this chapter suggests that journals and single-cell studies should require transparent reporting of parameter choices, marker genes, prior expectations, and the reasoning used to assign cell-type labels.

Future work could validate these findings in larger and more diverse analyst populations, including professional single-cell researchers. In addition, more standardized and evidence-weighted marker-gene guidelines are needed, together with transparent reporting standards for parameter choices, prior expectations, and annotation reasoning.

5.2 IntegrateRigor: annotation-free integration optimization for cell identity recovery reveals cancer–immune interface niches

Chapter 3 introduces IntegrateRigor, a data-driven and method-agnostic framework that reframes single-cell and spatial transcriptomics integration as a dataset-adaptive optimization problem. The results showed that batch-stable gene selection improves integration by excluding genes dominated by sample-specific or technical variation while retaining genes that

preserve shared cell identity structure. In the IFALD dataset, batch-unstable genes were enriched for mitochondrial, ribosomal, inflammatory, coagulation, and metabolic signals, consistent with the idea that poor integration can arise when input genes reflect nuisance variation rather than transferable cell identity structure.

IntegrateRigor also provides an annotation-free integration score that balances batch alignment and cell identity preservation. The results showed that optimizing only for label-free batch-mixing metrics can lead to over-integration, whereas IntegrateRigor selects intermediate hyperparameter settings that better preserve biological structure while improving batch correction. Across diverse integration tasks involving different sample conditions, sequencing platforms, developmental stages, species, and spatial tissue sections, IntegrateRigor consistently improved batch alignment while preserving or improving cell identity structure. Importantly, the relative performance of integration methods changed after dataset-specific optimization; for example, Harmony showed only moderate performance under default settings but became one of the strongest methods when optimized by IntegrateRigor.

In the colorectal cancer single-cell and spatial transcriptomics application, IntegrateRigor revealed cancer-immune niche structures that were obscured by both under-integration under default settings and over-integration in previous analyses. These candidate niches were supported by embedding structure, spatial localization near cancer-immune boundaries, mixed immune and epithelial marker expression, and functional enrichment patterns.

Future work can extend IntegrateRigor by incorporating known sample-level covariates, such as age, disease status, or preservation protocol, to better distinguish unwanted between-batch variation from biologically meaningful structured differences. Another direction is to extend the framework to cross-modality, multi-reference, or semi-supervised settings, especially when partial cell identity annotations are available.

In summary, this chapter suggests that progress in integration should come not only from developing new algorithms, but also from principled frameworks that allow existing methods to adapt to each dataset.

5.3 Benchmarking of cell-type visualization pipelines with semi-synthetic atlas-scale scRNA-seq datasets

Chapter 4 presents a benchmark framework for evaluating full cell-type visualization pipelines on atlas-scale scRNA-seq datasets. Rather than treating visualization as an isolated step, this chapter evaluates complete pipelines that combine normalization, integration, and visualization methods. This design is motivated by the observation that preprocessing steps can strongly shape 2D visualization, and that the best method at one step does not necessarily produce the best overall pipeline when combined with methods from other steps. The benchmark therefore focuses on pipeline-level performance and on interactions among normalization, integration, and visualization.

The analysis showed that semi-synthetic data generated by scDesign3 can closely resemble real scRNA-seq data while providing known ground truths for evaluation. This supports the use of semi-synthetic atlas-scale datasets as a practical strategy for benchmarking visualization quality without relying entirely on potentially biased real-data labels. The runtime and peak memory also identified methods that were computationally impractical at atlas scale, including PHATE and denSNE, which were excluded from the full benchmark because of excessive runtimes. Using the remaining methods, the study constructed 180 valid pipelines and demonstrated how multiple metrics can be aggregated to compare pipeline performance.

Future work could complement semi-synthetic benchmarking with evaluations on the original real datasets, allowing pipeline performance to be assessed under both controlled ground-truth settings and realistic biological complexity. The benchmark could also be expanded to larger datasets and more method combinations, and data-adaptive frameworks such as IntegrateRigor may be incorporated to compare not only fixed default pipelines but also optimized pipelines with dataset-specific parameter selection.

Bibliography

[1] Aviv Regev et al. “The human cell atlas.” In: *elife* 6 (2017), e27041.

[2] Dario Bressan, Giorgia Battistoni, and Gregory J Hannon. “The dawn of spatial omics.” In: *Science* 381.6657 (2023), eabq4964.

[3] Getnet Molla Desta and Alemayehu Godana Birhanu. “Advancements in single-cell RNA sequencing and spatial transcriptomics: transforming biomedical research.” In: *Acta Biochimica Polonica* 72 (2025), p. 13922.

[4] Guohao Liang, Hong Yin, and Fangyuan Ding. “Technical advances and applications of spatial transcriptomics.” In: *GEN biotechnology* 2.5 (2023), pp. 384–398.

[5] Christoph Hafemeister and Rahul Satija. “Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression.” In: *Genome biology* 20.1 (2019), pp. 1–15.

[6] Luke Zappia et al. “Feature selection methods affect the performance of scRNA-seq data integration and querying.” In: *Nature methods* (2025), pp. 1–11.

[7] Hoa Thi Nhu Tran et al. “A benchmark of batch-effect correction methods for single-cell RNA sequencing data.” In: *Genome biology* 21 (2020), pp. 1–32.

[8] Vincent D Blondel et al. “Fast unfolding of communities in large networks.” In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008.

[9] Dmitry Kobak and Philipp Berens. “The art of using t-SNE for single-cell transcriptomics.” In: *Nature communications* 10.1 (2019), p. 5416.

[10] Leland McInnes, John Healy, and James Melville. “Umap: Uniform manifold approximation and projection for dimension reduction.” In: *arXiv preprint arXiv:1802.03426* (2018).

[11] Malte D. Luecken and Fabian J. Theis. “Current best practices in single-cell RNA-seq analysis: a tutorial.” In: *Molecular Systems Biology* 15.6 (2019), e8746. DOI: 10.15252/msb.20188746.

[12] Lukas Heumos et al. “Best practices for single-cell analysis across modalities.” In: *Nature Reviews Genetics* 24 (2023), pp. 550–572.

[13] Changde Cheng et al. “A review of single-cell rna-seq annotation, integration, and cell–cell communication.” In: *Cells* 12.15 (2023), p. 1970.

[14] S. Zhang et al. “Review of single-cell RNA-seq data clustering for cell-type identification and characterization.” In: *RNA* 29.5 (2023), pp. 517–530. DOI: 10.1261/rna.078965.121.

[15] A Sina Booeshaghi et al. “Depth normalization for single-cell genomics count data.” In: *bioRxiv* (2022).

[16] Michael B Cole et al. “Performance assessment and selection of normalization procedures for single-cell RNA-seq.” In: *Cell systems* 8.4 (2019), pp. 315–328.

[17] Lyla Atta et al. “Gene count normalization in single-cell imaging-based spatially resolved transcriptomics.” In: *Genome Biology* 25.1 (2024), p. 153.

[18] Pengyi Yang, Hao Huang, and Chunlei Liu. “Feature selection revisited in the single-cell era.” In: *Genome Biology* 22.1 (2021), p. 321.

[19] F William Townes et al. “Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model.” In: *Genome biology* 20.1 (2019), pp. 1–16.

[20] Lauren L Hsu and Aedín C Culhane. “Correspondence analysis for dimension reduction, batch integration, and visualization of single-cell RNA-seq data.” In: *Scientific Reports* 13.1 (2023), p. 1197.

[21] Shiquan Sun et al. “Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis.” In: *Genome biology* 20.1 (2019), pp. 1–21.

[22] Geng Chen, Baitang Ning, and Tieliu Shi. “Single-cell RNA-seq technologies and related computational data analysis.” In: *Frontiers in genetics* 10 (2019), p. 317.

[23] Ricard Argelaguet et al. “Computational principles and challenges in single-cell data integration.” In: *Nature biotechnology* 39.10 (2021), pp. 1202–1215.

[24] Malte D Luecken et al. “Benchmarking atlas-level data integration in single-cell genomics.” In: *Nature methods* 19.1 (2022), pp. 41–50.

[25] Hoa Thi Nhu Tran et al. “A benchmark of batch-effect correction methods for single-cell RNA sequencing data.” In: *Genome biology* 21.1 (2020), p. 12.

[26] Yang Liu et al. “Robust integration of multiple single-cell RNA sequencing datasets using a single reference space.” In: *Nature biotechnology* 39.7 (2021), pp. 877–884.

[27] Karin Hrovatin et al. “Integrating single-cell RNA-seq datasets with substantial batch effects.” In: *BioRxiv* (2024), pp. 2023–11.

[28] Maren Büttner et al. “A test metric for assessing single-cell RNA-seq batch correction.” In: *Nature methods* 16.1 (2019), pp. 43–49.

[29] Shuang Li et al. “Toward informed batch correction for single-cell transcriptome integration.” In: *Nature Computational Science* (2026), pp. 1–11.

[30] Tallulah S Andrews and Martin Hemberg. “Identifying cell populations with scRNASeq.” In: *Molecular aspects of medicine* 59 (2018), pp. 114–122.

[31] Fu Xiang Quah and Martin Hemberg. “SC3s: efficient scaling of single cell consensus clustering to millions of cells.” In: *BMC bioinformatics* 23.1 (2022), p. 536.

[32] Zhiqian Zhai et al. “Supervised capacity preserving mapping: a clustering guided visualization method for scRNA-seq data.” In: *Bioinformatics* 38.9 (2022), pp. 2496–2503.

[33] Laurens Van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* 9.11 (2008).

[34] Ashwin Narayan, Bonnie Berger, and Hyunghoon Cho. “Assessing single-cell transcriptomic variability through density-preserving data visualization.” In: *Nature biotechnology* 39.6 (2021), pp. 765–774.

[35] Kevin R Moon et al. “PHATE: a dimensionality reduction method for visualizing trajectory structures in high-dimensional biological data.” In: *BioRxiv* 120378 (2017).

[36] Vivien Marx. "Seeing data as t-SNE and UMAP do." In: *Nature Methods* 21.6 (2024), pp. 930–933.

[37] Zoe A Clarke et al. "Tutorial: guidelines for annotating single-cell transcriptomic maps using automated and manual methods." In: *Nature protocols* 16.6 (2021), pp. 2749–2764.

[38] Etienne Becht et al. "Dimensionality reduction for visualizing single-cell data using UMAP." In: *Nature biotechnology* 37.1 (2019), pp. 38–44.

[39] Beate Vieth et al. "A systematic evaluation of single cell RNA-seq analysis pipelines." In: *Nature Communications* 10.1 (2019), p. 4667. DOI: 10.1038/s41467-019-12266-7.

[40] Luyi Tian et al. "Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments." In: *Nature methods* 16.6 (2019), pp. 479–487.

[41] Wanqiu Chen et al. "A multicenter study benchmarking single-cell RNA sequencing technologies using reference samples." In: *Nature Biotechnology 2020 39:9* 39 (9 Dec. 2020), pp. 1103–1114. ISSN: 1546-1696. DOI: 10.1038/s41587-020-00748-9. URL: https://www.nature.com/articles/s41587-020-00748-9.

[42] Dvir Aran et al. "Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage." In: *Nature immunology* 20.2 (2019), pp. 163–172.

[43] Yuqi Tan and Patrick Cahan. "SingleCellNet: a computational tool to classify single cell RNA-Seq data across platforms and across species." In: *Cell systems* 9.2 (2019), pp. 207–213.

[44] Fan Yang et al. "scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data." In: *Nature Machine Intelligence* 4.10 (2022), pp. 852–866.

[45] Jacob H Levine et al. "Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis." In: *Cell* 162.1 (2015), pp. 184–197.

[46] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. "From Louvain to Leiden: guaranteeing well-connected communities." In: *Scientific reports* 9.1 (2019), pp. 1–12.

[47] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[48] Yuhan Hao et al. "Dictionary learning for integrative, multimodal and scalable single-cell analysis." In: *Nature biotechnology* 42.2 (2024), pp. 293–304.

[49] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. "SCANPY: large-scale single-cell gene expression data analysis." In: *Genome biology* 19.1 (2018), p. 15.

[50] Yuhan Hao et al. "Dictionary learning for integrative, multimodal and scalable single-cell analysis." In: *Nature biotechnology* 42.2 (2024), pp. 293–304.

[51] Dylan Kotliar et al. "Identifying gene expression programs of cell-type identity and cellular activity with single-cell RNA-Seq." In: *eLife* 8 (2019), e43803. DOI: 10.7554/eLife.43803.

[52] Avishai Gavish et al. "Hallmarks of transcriptional intratumour heterogeneity across a thousand tumours." In: *Nature* 618.7965 (2023), pp. 598–606. DOI: 10.1038/s41586-023-06130-4.

[53] Hassaan Maan et al. "Characterizing the impacts of dataset imbalance on single-cell data integration." In: *Nature Biotechnology* 42.12 (2024), pp. 1899–1908.

[54] Yang Zhou et al. "Quantifying batch effects for individual genes in single-cell data." In: *Nature Computational Science* (2025), pp. 1–9.

[55] Pia Rautenstrauch and Uwe Ohler. "Shortcomings of silhouette in single-cell integration benchmarking." In: *Nature Biotechnology* (2025), pp. 1–5.

[56] Ilya Korsunsky et al. "Fast, sensitive and accurate integration of single-cell data with Harmony." In: *Nature methods* 16.12 (2019), pp. 1289–1296.

[57] Nghia Millard et al. "Batch correcting single-cell spatial transcriptomics count data with Crescendo improves visualization and detection of spatial gene patterns." In: *Genome biology* 26.1 (2025), p. 36.

[58] Romain Lopez et al. “Deep generative modeling for single-cell transcriptomics.” In: *Nature methods* 15.12 (2018), pp. 1053–1058.

[59] Laleh Haghverdi et al. “Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors.” In: *Nature biotechnology* 36.5 (2018), pp. 421–427.

[60] Jialin Liu et al. “Jointly defining cell types from multiple single-cell datasets using LIGER.” In: *Nature protocols* 15.11 (2020), pp. 3632–3662.

[61] Constantin Ahlmann-Eltze and Wolfgang Huber. “Comparison of transformations for single-cell RNA-seq data.” In: *Nature Methods* (2023), pp. 1–8.

[62] Hoa Thi Nhu Tran et al. “A benchmark of batch-effect correction methods for single-cell RNA sequencing data.” In: *Genome biology* 21 (2020), pp. 1–32.

[63] Malte D Luecken et al. “Benchmarking atlas-level data integration in single-cell genomics.” In: *Nature methods* 19.1 (2022), pp. 41–50.

[64] Haiyang Huang et al. “Towards a comprehensive evaluation of dimension reduction methods for transcriptomic data visualization.” In: *Communications biology* 5.1 (2022), pp. 1–11.

[65] Nicholas Schaum et al. “Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris: The Tabula Muris Consortium.” In: *Nature* 562.7727 (2018), p. 367.

[66] Aviv Regev et al. “The human cell atlas.” In: *elife* 6 (2017), e27041.

[67] Dongyuan Song et al. “scDesign3 generates realistic in silico data for multimodal single-cell and spatial omics.” In: *Nature Biotechnology* (2023), pp. 1–6.

[68] Trevor Atkeson et al. “A novel paradigm for single-cell annotation in stem cell research.” In: *Stem Cell Reports* (2025).

[69] Lijia Yu et al. “Benchmarking clustering algorithms on estimating the number of cell types from single-cell RNA-sequencing data.” In: *Genome biology* 23.1 (2022), p. 49.

[70] Qiqing Fu et al. “A comparison of scRNA-seq annotation methods based on experimentally labeled immune cell subtype dataset.” In: *Briefings in Bioinformatics* 25.5 (2024), bbae392.

[71] Kexin Huang et al. “Biomni: A General-Purpose Biomedical AI Agent.” In: Cold Spring Harbor Laboratory, 2025. DOI: 10.1101/2025.05.30.656746. eprint: https://www.biorxiv.org/content/early/2025/06/02/2025.05.30.656746.full.pdf. URL: https://www.biorxiv.org/content/early/2025/06/02/2025.05.30.656746.

[72] Paolo Strati et al. “Prolonged cytopenia following CD19 CAR T cell therapy is linked with bone marrow infiltration of clonally expanded IFNγ-expressing CD8 T cells.” In: *Cell Reports Medicine* 4.8 (2023).

[73] Maria Fasolino et al. “Single-cell multi-omics analysis of human pancreatic islets reveals novel cellular states in type 1 diabetes.” In: *Nature metabolism* 4.2 (2022), pp. 284–299.

[74] GW Brown and AM Mood. “Homogeneity of several samples.” In: *The American Statistician* 2.3 (1948).

[75] WW Daniel. “Jonckheere–Terpstra test for ordered alternatives.” In: *Applied nonparametric statistics* (1990), pp. 234–240.

[76] Orit Rozenblatt-Rosen et al. “The Human Cell Atlas: from vision to reality.” In: *Nature* 550.7677 (2017), pp. 451–453.

[77] Rasa Elmentaite et al. “Single-cell atlases: shared and tissue-specific cell types across human organs.” In: *Nature Reviews Genetics* 23.7 (2022), pp. 395–410.

[78] Lisa Sikkema et al. “An integrated cell atlas of the lung in health and disease.” In: *Nature medicine* 29.6 (2023), pp. 1563–1577.

[79] Samantha A Morris. “The evolving concept of cell identity in the single cell era.” In: *Development* 146.12 (2019), dev169748.

[80] Tomasz Kujawa, Michał Marczyk, and Joanna Polanska. "Influence of single-cell RNA sequencing data integration on the performance of differential gene expression analysis." In: *Frontiers in Genetics* 13 (2022), p. 1009316.

[81] Nikolaos Patikas et al. "Integration of large, complex single-cell datasets with Harmony2." In: *bioRxiv* (2026), pp. 2026–03.

[82] Chenling Xu et al. "Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models." In: *Molecular systems biology* 17.1 (2021), MSB20209620.

[83] Brian Hie, Bryan Bryson, and Bonnie Berger. "Efficient integration of heterogeneous single-cell transcriptomes using Scanorama." In: *Nature biotechnology* 37.6 (2019), pp. 685–691.

[84] Zhaojun Zhang et al. "Recovery of biological signals lost in single-cell batch integration with CellANOVA." In: *Nature Biotechnology* (2024), pp. 1–17.

[85] Megan Crow et al. "Characterizing the replicability of cell types defined by single cell RNA-sequencing data using MetaNeighbor." In: *Nature Communications* 9.1 (2018), p. 884. DOI: 10.1038/s41467-018-03282-0.

[86] The Tabula Sapiens Consortium. "The Tabula Sapiens: A multiple-organ, single-cell transcriptomic atlas of humans." In: *Science* 376.6594 (2022), eabl4896. DOI: 10.1126/science.abl4896.

[87] Maria Antonietta Tosches et al. "Evolution of pallium, hippocampus, and cortical cell types revealed by single-cell transcriptomics in reptiles." In: *Science* 360.6391 (2018), pp. 881–888. DOI: 10.1126/science.aar4237.

[88] Trygve E Bakken et al. "Comparative cellular analysis of motor cortex in human, marmoset and mouse." In: *Nature* 598.7879 (2021), pp. 111–119.

[89] Lawrence Hubert and Phipps Arabie. "Comparing partitions." In: *Journal of Classification* 2 (1985), pp. 193–218.

[90] Rachel D Edgar et al. “Single-cell atlas of human pediatric liver reveals age-related hepatic gene signatures.” In: *Hepatology Communications* 9.11 (2025), e0813.

[91] Ville Satopaa et al. “Finding a” kneedle” in a haystack: Detecting knee points in system behavior.” In: *2011 31st international conference on distributed computing systems workshops*. IEEE. 2011, pp. 166–171.

[92] D. Mari, R. Coppola, and R. Provenzano. “Hemostasis Factors and Aging.” In: *Experimental Gerontology* 43.2 (2008), pp. 66–73.

[93] S. S. Kale et al. “Effects of Aging on Inflammation and Hemostasis through the Continuum of Critical Illness.” In: *Aging and Disease* 2.6 (2011), pp. 501–511.

[94] Pedro G. Ferreira et al. “The Effects of Death and Post-Mortem Cold Ischemia on Human Tissue Transcriptomes.” In: *Nature Communications* 9 (2018), p. 490. DOI: 10.1038/s41467-017-02772-x.

[95] Hyun Min Kang et al. “Multiplexed droplet single-cell RNA-sequencing using natural genetic variation.” In: *Nature biotechnology* 36.1 (2018), pp. 89–94.

[96] Madhav Mantri et al. “Spatiotemporal single-cell RNA sequencing of developing chicken hearts identifies interplay between cellular differentiation and morphogenesis.” In: *Nature communications* 12.1 (2021), p. 1771.

[97] Meng Zhang et al. “Molecularly defined and spatially resolved cell atlas of the whole mouse brain.” In: *Nature* 624 (2023), pp. 343–354.

[98] Maayan Baron et al. “A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure.” In: *Cell systems* 3.4 (2016), pp. 346–360.

[99] Alexander Strehl and Joydeep Ghosh. “Cluster ensembles—a knowledge reuse framework for combining multiple partitions.” In: *Journal of Machine Learning Research* 3 (2002), pp. 583–617.

[100] Michael Ashburner et al. “Gene ontology: tool for the unification of biology.” In: *Nature Genetics* 25.1 (2000), pp. 25–29.

[101] The Gene Ontology Consortium. "The Gene Ontology Resource: 20 years and still GOing strong." In: *Nucleic Acids Research* 47.D1 (2019), pp. D330–D338.

[102] Guangchuang Yu et al. "clusterProfiler: an R package for comparing biological themes among gene clusters." In: *OMICS: A Journal of Integrative Biology* 16.5 (2012), pp. 284–287.

[103] Dmitry I. Gabrilovich, Suzanne Ostrand-Rosenberg, and Vincenzo Bronte. "Coordinated regulation of myeloid cells by tumours." In: *Nature Reviews Immunology* 12.4 (2012), pp. 253–268.

[104] Mikhail Binnewies et al. "Understanding the tumor immune microenvironment (TIME) for effective therapy." In: *Nature Medicine* 24.5 (2018), pp. 541–550.

[105] Daniel Rodríguez-Pinto. "B cells as antigen presenting cells." In: *Cellular Immunology* 238.2 (2005), pp. 67–75.

[106] Dipankar Ghosh et al. "New insights into B cells as antigen presenting cells." In: *Current Opinion in Immunology* 70 (2021), pp. 129–137.

[107] Indu Rastogi et al. "Role of B cells as antigen presenting cells." In: *Frontiers in Immunology* 13 (2022), p. 954936.

[108] Mohamad A. Salkeni and Aung Naing. "Interleukin-10 in cancer immunotherapy: from bench to bedside." In: *Trends in Cancer* 9.9 (2023), pp. 716–725.

[109] Peng Chen et al. "Tumour-reactive plasma cells in antitumour immunity." In: *Nature Reviews Immunology* (2024).

[110] Ann Wasiuk et al. "Mast cells as regulators of adaptive immunity to tumours." In: *Clinical and Experimental Immunology* 155.2 (2009), pp. 140–146.

[111] Xiaoyan Guo et al. "Role of mast cells activation in the tumor immune microenvironment." In: *European Journal of Pharmacology* 960 (2023), p. 176122.

[112] Jennifer N. Lichterman and Saraswati M. Reddy. "Mast Cells: A New Frontier for Cancer Immunotherapy." In: *Cells* 10.6 (2021), p. 1270.

[113] Dongyuan Song et al. “scDesign3 generates realistic in silico data for multimodal single-cell and spatial omics.” In: *Nature Biotechnology* 42.2 (2024), pp. 247–252.

[114] Zihao Chen et al. “Directly selecting cell-type marker genes for single-cell clustering analyses.” In: *Cell Reports Methods* 4.7 (2024).

[115] Frank J Massey Jr. “The Kolmogorov-Smirnov test for goodness of fit.” In: *Journal of the American statistical Association* 46.253 (1951), pp. 68–78.

[116] Jorge Faber and Lilian Martins Fonseca. “How sample size influences research outcomes.” In: *Dental press journal of orthodontics* 19 (2014), pp. 27–29.

[117] Persi Diaconis and David Freedman. “Asymptotics of graphical projection pursuit.” In: *The annals of statistics* (1984), pp. 793–815.

[118] Adam Gayoso et al. “A Python library for probabilistic analysis of single-cell omics data.” In: *Nature Biotechnology* 40 (2022), pp. 163–166.

[119] Laleh Haghverdi et al. “Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors.” In: *Nature biotechnology* 36.5 (2018), pp. 421–427.

[120] Ilya Korsunsky et al. “Fast, sensitive and accurate integration of single-cell data with Harmony.” In: *Nature methods* 16.12 (2019), pp. 1289–1296.

[121] Adam Gayoso et al. “A Python library for probabilistic analysis of single-cell omics data.” In: *Nature biotechnology* 40.2 (2022), pp. 163–166.

[122] Jialin Liu et al. “Jointly defining cell types from multiple single-cell datasets using LIGER.” In: *Nature protocols* 15.11 (2020), pp. 3632–3662.

[123] Rongrong Wang and Xiaopeng Zhang. “Capacity preserving mapping for high-dimensional data visualization.” In: *arXiv preprint arXiv:1909.13322* (2019).

[124] Jiarui Ding et al. “Systematic comparison of single-cell and single-nucleus RNA-sequencing methods.” In: *Nature biotechnology* 38.6 (2020), pp. 737–746.

[125] Marcos Otero-Garcia et al. “Molecular signatures underlying neurofibrillary tangle susceptibility in Alzheimer’s disease.” In: *Neuron* 110.18 (2022), 2929–2948.e8.

[126] Laura Jardine et al. “Blood and immune development in human fetal bone marrow and Down syndrome.” In: *Nature* 598.7880 (2021), pp. 327–331.

[127] Yi-Rong Peng et al. “Molecular classification and comparative taxonomics of foveal and peripheral cells in primate retina.” In: *Cell* 176.5 (2019), pp. 1222–1237.

[128] Xinyi Xu et al. “Propensity score matching enables batch-effect-corrected imputation in single-cell RNA-seq analysis.” In: *Briefings in Bioinformatics* 23.4 (2022).

[129] Constantin Ahlmann-Eltze and Wolfgang Huber. “Comparison of transformations for single-cell RNA-seq data.” In: *Nature Methods* 20.5 (2023), pp. 665–672.

[130] Jérémie Breda, Mihaela Zavolan, and Erik van Nimwegen. “Bayesian inference of gene expression states from single-cell RNA-seq data.” In: *Nature Biotechnology* 39.8 (2021), pp. 1008–1016.

[131] Laurens Van Der Maaten. “Accelerating t-SNE using tree-based algorithms.” In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 3221–3245.

[132] George C Linderman et al. “Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data.” In: *Nature methods* 16.3 (2019), pp. 243–245.

[133] Mathieu Jacomy et al. “ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software.” In: *PloS one* 9.6 (2014), e98679.

[134] Joseph M Rich et al. “The impact of package selection and versioning on single-cell RNA-seq analysis.” In: *bioRxiv* (2024).

[135] Yingying Cao et al. “UMI or not UMI, that is the question for scRNA-seq zero-inflation.” In: *Nature Biotechnology* 39.2 (2021), pp. 158–159.

[136] Davis J McCarthy, Yunshun Chen, and Gordon K Smyth. “Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation.” In: *Nucleic acids research* 40.10 (2012), pp. 4288–4297.

[137] Abhishek Sarkar and Matthew Stephens. "Separating measurement and expression models clarifies confusion in single-cell RNA sequencing analysis." In: *Nature genetics* 53.6 (2021), pp. 770–777.

[138] Jingshu Wang et al. "Gene expression distribution deconvolution in single-cell RNA sequencing." In: *Proceedings of the National Academy of Sciences* 115.28 (2018), E6437–E6446.

[139] Florian Wagner, Yun Yan, and Itai Yanai. "K-nearest neighbor smoothing for high-throughput single-cell RNA-Seq data." In: *BioRxiv* (2018), p. 217737.

[140] Naomi Altman and Martin Krzywinski. "The curse (s) of dimensionality." In: *Nat Methods* 15.6 (2018), pp. 399–400.

[141] Ashwin Narayan, Bonnie Berger, and Hyunghoon Cho. "Assessing single-cell transcriptomic variability through density-preserving data visualization." In: *Nature Biotechnology* 39.6 (2021), pp. 765–774.

[142] Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. "Measuring and testing dependence by correlation of distances." In: (2007).

[143] Lucy Xia, Christy Lee, and Jingyi Jessica Li. "Statistical method scDEED for detecting dubious 2D single-cell embeddings and optimizing t-SNE and UMAP hyperparameters." In: *Nature Communications* 15.1 (2024), p. 1753.

[144] Izaskun Mallona et al. "Omnibenchmark (alpha) for continuous and open benchmarking in bioinformatics." In: *arXiv preprint arXiv:2409.17038* (2024).

[145] Po-Yuan Tung et al. "Batch effects and the effective design of single-cell gene expression studies." In: *Scientific Reports* 7 (2017), p. 39921. DOI: 10.1038/srep39921.

[146] Fan Song et al. "Flexible experimental designs for valid single-cell RNA-sequencing experiments allowing batch effects correction." In: *Nature Communications* 11.1 (2020), p. 3274. DOI: 10.1038/s41467-020-16905-2.

[147] Mohammad Lotfollahi et al. "Mapping single-cell data to reference atlases by transfer learning." In: *Nature biotechnology* 40.1 (2022), pp. 121–130.