Distribution Agreement

Signature:

____________________________________ ____________

Wenjing Ma Date

Cell type identification in single-cell genomics and its applications

By

Wenjing Ma
Doctor of Philosophy

Computer Science and Informatics

Hao Wu, Ph.D.
Advisor

Joyce Ho, Ph.D.
Committee Member

Peng Jin, Ph.D.
Committee Member

Mingyao Li, Ph.D.
Committee Member

Carl Yang, Ph.D.
Committee Member

Accepted:

Kimberly Jacob Arriola, Ph.D, MPH
Dean of the James T. Laney School of Graduate Studies

Date

Cell type identification in single-cell genomics and its applications

By

Wenjing Ma
B.S., Beijing University of Posts and Telecommunications, 2014
M.S., Beijing University of Posts and Telecommunications, 2017

Advisor: Hao Wu, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2023

Abstract

Cell type identification in single-cell genomics and its applications

By Wenjing Ma

Advances in techniques for measuring genomics in cell-level resolution provide great opportunities to uncover cellular heterogeneity in genomic features of interest at the level of individual cells. Initiated by the introduction of single-cell RNA-sequencing (scRNA-seq), which measure transcriptomics information, single-cell techniques have been expanded to encompass other epigenomic modalities as well. Among all scientific goals in single-cell genomics studies, precise cell type identification (celltyping) is a fundamental and crucial step in analyzing single-cell genomics data. Supervised cell typing methods have become increasingly popular due to their superior accuracy, robustness, and efficiency. In our dissertation, we primarily focus on the development and application of supervised cell typing methods.

The dissertation starts with evaluating key factors for supervised celltyping methods developed for scRNA-seq data. After performing extensive real data analyses, we suggest combining all individuals from available datasets to construct the reference dataset and using the multi-layer perceptron (MLP) as the classifier, along with F-test as the feature selection method. This benchmark study not only offers valuable insights and suggestions for method developers but also lays the groundwork for our subsequent research endeavors.

We then developed a novel computational method with open-source software called Cellcano, which is specifically designed for the single-cell technique that profiles chromatin accessibility (scATAC-seq). Cellcano is based on a two-round supervised learning algorithm and provides significantly improved accuracy, robustness, and computational efficiency compared to existing tools. We have also explored the possibilities of using scRNA-seq data as references to perform a supervised manner of celltyping and data integration for scATAC-seq.

Upon accurate identification of distinct cell types, specific markers unique to each cell type can be extracted to enable diverse applications and downstream analyses. Based on cell-type-specific marker genes, we developed a method named LRcell to identify cellular activities associated with psychiatric disorders.

The computational and statistical methods employed in this dissertation are designed to provide a comprehensive understanding of cell-type-specificity. We anticipate that this research will contribute to the understanding of cellular functions in biological mechanisms and disease progression, potentially providing valuable insights for biomedical researchers.

Cell type identification in single-cell genomics and its applications

By

Wenjing Ma
B.S., Beijing University of Posts and Telecommunications, 2014
M.S., Beijing University of Posts and Telecommunications, 2017

Advisor: Hao Wu, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2023

Acknowledgments

I express my sincere appreciation to Emory University and the Computer Science department for providing me with this incredible opportunity to embark on my Ph.D. journey and gain diverse experiences. I am truly grateful for the unwavering support of our wonderful community. I take great pleasure in spending time with my cohort and friends from other departments during occasional gatherings, engaging in activities such as baking and playing a variety of sports. It's wonderful to have a supportive and enjoyable community that shares similar interests and passions. These activities not only provide a break from academic pursuits but also create cherished memories.

I would like to show my heartfelt gratitude to my advisor Dr. Hao Wu for making my Ph.D. grind more enjoyable, easier, and more fulfilling. Dr. Wu possesses a keen intellect and a passion for simplifying complex concepts. I have learned invaluable skills from him in applying advanced deep learning and statistical learning techniques to single-cell genomics data, ultimately developing practical tools and methods for biomedical research. His ideas always work and are presented in an elegant manner. In addition to his academic support, Dr. Wu has also been a mentor in cultivating my independence and responsibility as a researcher. During the early stages of my research, I often worried about being scooped, but he encouraged me to adopt an abundance mindset and foster collaborations to expand the field instead of engaging in cutthroat competition. As I become more familiar with my research area, Dr. Wu guides me toward critical thinking rather than aimless work. He skillfully transitioned from a hands-on to a hands-off mentoring style, empowering me to develop my own research independence. I am truly grateful for Dr. Wu's guidance, motivation, and unwavering support throughout my Ph.D. journey.

I also appreciate the invaluable contributions and constructive suggestions from my committee members, Dr. Joyce Ho, Dr. Peng Jin, Dr. Mingyao Li, and Dr. Carl Yang. I am grateful to Dr. Joyce Ho for providing me with valuable resources on deep learning, which has been instrumental in advancing my research. Dr. Peng Jin's expertise and insights from a biological perspective have challenged me to think critically about important questions that biomedical researchers are interested in. It has been a valuable experience collaborating with Dr. Peng Jin and his postdoc Dr. Yulin Jin. I express my gratitude to Dr. Mingyao Li for offering a unique perspective on the importance of data integration in our research. Her suggestions have expanded our thinking beyond celltyping, opening up new avenues for exploration. I also appreciate Dr. Carl Yang for his expertise in adversarial learning and his unwavering support in sharing ideas and insights, which have enriched my research experience.

Additional thanks should go to my rotation project advisor Dr. Zhaohui Qin, for recognizing my potential in research and nominating me for my first fellowship. During my rotation, he makes every effort to support my research endeavors. He has provided invaluable guidance in helping me find research topics and facilitate collaborations with other researchers. His support and belief in my abilities have been instrumental in my academic journey, and I am grateful for the opportunity to work with him. I would also like to express my thanks to Dr. Chongzhi Zhang, my visiting advisor, for his constant support throughout my Ph.D. journey. I am always grateful for his willingness to host me as a visiting scholar at the University of Virginia and this provides me with the opportunity to start my academic career. Upon graduation, he generously provided me with valuable insights and guidance on shaping my future academic plans.

Last but certainly not least, I would like to express my deepest gratitude to my beloved family, including my husband Jiaying Lu, my parents, and my cat Yu. I would like to extend special thanks to Jiaying, who has been a constant source of encouragement in my pursuit of challenging endeavors and has inspired me to be an independent and empowered researcher. He has always been supportive and provided me with instrumental discussions on obtaining the knowledge and skills to learn and apply deep learning techniques. Without his guidance and assistance, I would not have been able to come this far in my academic pursuits. I would also want to express my deepest appreciation to my father and my mother. My father has provided me with endless support and encouragement for me to persevere on my journey. My mother even flew here from China to provide me with in-person support, despite not being able to speak any English. Their words and actions have been a significant source of motivation and inspiration for me. Additionally, I am grateful to have an adorable and lovely cat named Yu. Her presence has brought me comfort during stressful times and has provided me with countless happy moments. Finally, words cannot express my feelings for all companionship from my family and my friends. It has been a source of joy and relaxation throughout my Ph.D. journey, and I am thankful to have them by my side.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Single-cell genomics

Recent years have witnessed the rapid growth of techniques in measuring genomics in a single-cell resolution, which has greatly enhanced the understanding of biological mechanisms in complex tissues [64] [29]. The advent of single-cell genomics was initiated by the introduction of single-cell RNA-sequencing (scRNA-seq), which enables the measurement of transcriptomic information from individual cells. To gain a more holistic understanding of gene regulation and cellular function, this approach has been expanded to encompass other epigenomic modalities, including chromatin accessibility [14], DNA methylation [71], histone modifications [9], and more. This broader application of single-cell techniques has provided researchers with a comprehensive perspective on the molecular landscape of individual cells, shedding light on their regulatory mechanisms and functional diversity. Unlike bulk experiments that measure the average genomics profile from a collection of cells, single-cell genomics offers a higher-resolution approach by profiling genomic features of interest for each individual cell. This provides richer and more detailed information that cannot be obtained from bulk data alone. Single-cell genomics allows for studying the composition of cell types in complex tissues, uncovering cell-to-cell heterogeneity, and exploring the dynamics of biological processes such as development, differentiation, and disease progression. The ability to capture data at the level of individual cells opens up new possibilities for addressing important research questions and gaining deeper insights into complex biological systems.

Single-cell genomics studies aim to achieve several scientific goals. The most fundamental and critical one is to unravel the cellular composition of complex tissues, which involves identifying the various cell types and subtypes (referred to as ”celltyping” hereafter) present in a tissue sample, as well as determining their relative proportions. Understanding the cellular composition of tissues can have significant implications in biological and clinical practices. For instance, in the context of tumor research, analyzing the composition of tumor-infiltrating immune cells can provide valuable insights into anti-tumor immune responses and inform treatment strategies [59]. After revealing cell types, cell-type-specific genomic features can be obtained and are also of great interest as they enhance the understanding of cell signatures [79].

1.2 Supervised celltyping in single-cell genomics

There are two commonly used approaches for performing celltyping in single-cell genomics studies: experimental procedures and computational methods. Experimental procedures typically involve the use of fluorescence-activated cell sorting (FACS), which targets specific antigens on the cell surface to sort cells into different populations. While FACS can sort multiple cell populations simultaneously, it can be expensive and requires sophisticated instruments and careful experimental design, making it impractical for large-scale studies [26].

On the other hand, computational methods offer a more practical and efficient approach to celltyping. Traditional approaches for celltyping typically rely on unsupervised clustering methods [49] [35]. These methods group cells into clusters based on their molecular profiles, such as gene expression or epigenetic marks. Subsequently, domain knowledge, such as marker genes or cell-type-specific chromatin openness, is employed to iteratively refine and curate the clusters and their labels. However, this whole process can be labor-intensive and time-consuming, requiring manual curation and high expertise to accurately assign cell types. Moreover, the unsupervised methods often do not scale up well computationally with cell numbers [51]. Given the significant efforts invested in annotating single-cell genomics data, a natural thought would be whether we can leverage the high-quality and well-annotated datasets to better perform celltyping. This motivates the development of supervised celltyping methods, making it a highly active and evolving research area in recent years [106] [86]. These methods first construct a classifier from a reference dataset with known cell types. Then, for a given target dataset, they assign cell types for every single cell based on the trained classifier. Additionally, there also exist a few "semi-supervised" methods [48] [20], where they still perform unsupervised clustering but obtain initial values of the parameters from a reference dataset.

Supervised celltyping methods offer several advantages. Firstly, they often yield superior performance compared to unsupervised methods. Secondly, they are not influenced by the sample size (number of cells) of the target data, as they predict cell types for each cell individually. In contrast, unsupervised clustering methods may require larger datasets and their performance may be affected by the number of cells. Additionally, supervised methods are more effective in handling data with imbalanced cell type proportions compared to unsupervised methods, which can struggle with accuracy when dealing with highly imbalanced cluster sizes [62].

Despite the advantages of supervised celltyping in single-cell genomics, there are also challenges to overcome. Supervised methods heavily rely on various factors, such as the selection of predictive features, the construction of the prediction model, and the choice of the reference dataset. Additionally, when performing cross-modality prediction, the feature space may not be unified, as different single-cell genomics profiles may have distinct data distributions. For example, scATAC-seq data can be summarized into genome-wide fixed-size bins, peaks representing accessible regions, or genes [19]. Even when summarizing scATAC-seq data into a common gene-level feature space, the data distribution may differ among different single-cell genomics profiles. These challenges have motivated the work in this dissertation, which focuses on evaluating the factors involved in supervised celltyping for scRNA-seq and developing supervised celltyping methods for scATAC-seq.

1.3 Applications after accurate celltyping

Once accurate cell typing has been achieved, it allows for obtaining cell-type-specific information, which in turn enables downstream analyses to be performed. Several methods have been developed to extract cell-type-specific genomic features [89] [91] and investigate cell dynamics [17][100]. The valuable cell-type-specific information can be extensively utilized in tasks such as single-cell genomics data integration and deciphering bulk experiments.

On one hand, current approaches for integrating single-cell genomics data involve projecting the datasets into a shared embedding space and then performing cell label transfer to integrate the cell type information across different datasets [70] [40] [63]. However, without the incorporation of cell-type-specificity information as guidance, these methods may inadvertently mix cell types, leading to inaccurate downstream analysis results. On the other hand, cell-type-specificity can also be utilized to uncover previously unseen information from bulk experiments. For instance, deconvolution, which involves inferring the cell type composition from bulk experiments using a cell-type-specific marker gene panel, can provide insights into cell population differences and potential therapeutic targets for disease treatment [6]. This dissertation employs cell-type-specificity to aid in the integration of multiple scRNA-seq and scATAC-seq datasets. In addition, we utilize it to uncover the potential cellular activity associated with differentially expressed genes in bulk studies.

1.4 Outline

This dissertation presents computational techniques for identifying and exploiting cell-type-specificity. Chapters 2 and 3 concentrate on evaluating and developing supervised celltyping approaches for single-cell genomics data. In Chapter 2, we assess important components of supervised celltyping in scRNA-seq, which serves as the groundwork for our novel supervised celltyping method, Cellcano, presented in Chapter 3. Cellcano is specifically designed for scATAC-seq data and utilizes a two-round supervised learning algorithm, providing superior accuracy, robustness, and computational efficiency in comparison to existing tools. In Chapter 4, we focus on applications of utilizing cell-type-specificity. In the first half, we describe our methodology named CellAMA for integrating multiple single-cell genomics datasets. Then, in the second half, we introduce our method named LRcell, which uses cell-type-specific marker genes to detect potential cellular activities in bulk differential expression studies. Finally, in the last chapter, we discuss current challenges and future research plans in celltyping and its applications.

Chapter 2

Evaluating key factors of supervised celltyping for scRNA-seq data

2.1 Introduction

As outlined in the Introduction, celltyping is a crucial and fundamental component of single-cell genomics analysis and it has been established that supervised celltyping methods outperform unsupervised approaches in terms of accuracy, robustness, and efficiency. Nevertheless, several challenges persist in identifying the key factors necessary to achieve superior prediction performance in supervised celltyping for scRNA-seq. Important questions include the selection of predictive features, the construction of the prediction model, and the choice of the reference dataset. Investigating these issues is crucial for researchers utilizing supervised methods for cell type prediction in practice. While existing publications often showcase their results by selecting a single dataset [1] [45] or combining multiple datasets after removing batch effects [101], these approaches have limitations. A single dataset may introduce bias and with the increasing number of scRNA-seq datasets being generated, there is an urgent need for proper guidance on how to maximize the utility of existing datasets to construct reference datasets.

In this work, we perform extensive and comprehensive real data analyses to systematically evaluate the strategies in supervised celltyping in terms of feature selection, prediction model, and choice of reference datasets [73]. We have also discussed the impact of data preprocessing including batch effect removal and data imputation. Although there are a few benchmark papers for comparing the performances of supervised celltyping methods [1] [45] [101] [86], they only compare "off-the-shelf" available tools, while we take a step further to evaluate the combinations of different strategies. More importantly, we evaluate the impact of the reference data and potential strategies for processing the reference data, which have never been investigated before to the best of our knowledge. Based on our analyses, we provide a guideline and rule of thumb for using the supervised celltyping methods.

2.2 Factors under evaluation

2.2.1 The choice of prediction model

We include the following nine supervised celltyping methods in the comparison, which cover a wide range of different strategies for supervised celltyping:

- Three off-the-shelf supervised learning methods: random forest [87], SVM with linear kernel, and SVM with radial basis function kernel [87]

- Two supervised celltyping methods specifically designed for scRNA-seq data based on the correlation between target data and reference data: scmap [50] and CHETAH [27]

- Two supervised deep learning methods: multi-layer perceptron (MLP) [92] and graph-embedded deep neural network (GEDFN) [52]
- Two semi-supervised deep learning method: ItClust [43] based on transfer learning and MARS with meta-learning concepts [12]

There are several other supervised celltyping methods available for scRNA-seq. For example, scSorter [37] borrows information from lowly expressed marker genes to assign cells; scPred [2] adopts a principal component analysis (PCA)-based feature selection; SingleCellNet [108] uses top-pair transformation on gene space and selects informative paired genes as features; CellAssign [120] builds a probabilistic model with some prior knowledge of cell markers, etc. But according to a recent comparison [1], SVM with rejection, scmap, and CHEAH are among the best performers, so we decide not to include more such methods. GEDFN is a method designed for predicting phenotype from bulk expression but can be directly applied to scRNA-seq celltyping. We include it because we want to understand whether incorporating protein-protein interaction network information can improve the results. ItClust is a semi-supervised method that uses the reference data to obtain initial values for unsupervised clustering in target data. MARS uses a meta-learning concept to construct cell-type-specific landmarks by jointly embedding both annotated and unannotated data without removing the batch effects and then assigning cell types based on the learned embedding space. We want to evaluate the performances of these semi-supervised methods under different scenarios.

2.2.2 Choice of predictive features

It is known that feature selection plays an important role in many high-throughput data analyses, including scRNA-seq cell clustering and supervised celltyping. Since most genes are not cell-type-specific, including them in the prediction model will dilute the signal and impair the prediction accuracy. Most celltyping methods have a feature selection step. When one evaluates the performance of a method, it is unclear whether the performance gain/loss comes from the feature selection or the method itself. We want to merely investigate the impact of feature selection, so we decouple this step from the prediction. We include two unsupervised feature selection methods: one is Seurat V2.0 [15], which is based on marginal gene expression and variation, and the other is FEAST [103], which is based on unsupervised consensus clustering followed by F-test for ranking features. Briefly speaking, FEAST first performs unsupervised consensus clustering (similar to that in SC3 [49]) and then performs F-test on the clusters to calculate the feature significance and rank features. We also include one supervised method using F-test to select features from the reference dataset where the cell types are known.

Another aspect of the problem is whether to select features from the reference or target datasets. It is obviously more desirable to select features from the reference data since one only needs to perform feature selection and prediction model construction once for each reference dataset. On the other hand, selecting features from the target data might be able to capture the target data characteristics more accurately and improve the prediction accuracy. In fact, ItClust suggests selecting features from the target data. In such a case, re-training the prediction model for each target data might be worth the extra computational burden. Thus, we evaluate the Seurat and FEAST feature selection in both reference and target datasets. Note that we have investigated the impact of feature number and decided to pick the top 1000 features for downstream analysis (Appendix A Section 4) in all feature selection procedures. As a baseline, we also include results from not selecting features at all. Altogether, we test 6 feature selection procedures.

2.2.3 Datasets

All datasets used in this study are listed in Appendix Tables A.1, A.2, and A.3. Briefly, we include multiple datasets from human peripheral blood mononuclear cells (PBMCs), human pancreas, and mouse brain. For human PBMCs datasets, we include studies from lupus patients [47] using 10X Chromium (denoted as "Human PBMCs lupus") and frozen (PBMCs1) and fresh (PBMCs2) samples [28] processed by three protocols including 10X Chromium, Smart-seq2, and CEL-seq2. For human pancreas datasets, we include three human pancreas datasets [81] [97] [116]. In mouse brains, the cell type composition is more complex and has variations among the brain regions. To simplify, we focus on the frontal cortex and hippocampus regions from adult mouse whole brain study [96] using Drop-seq (denoted as "Mouse brain FC" and "Mouse brain HC"), prefrontal cortex region from adolescence and addiction study [10] using 10X Chromium (denoted as "Mouse brain pFC"), cortex samples from [28] processed by DroNc-seq (denoted as "Mouse brain cortex"), and samples with frontal cortex regions extracted from [117] processed by 10X Chromium (denoted as "Mouse brain Allen"). The cell types from the above datasets are annotated in the literature by unsupervised clustering and known marker gene expression. To ascertain the computationally derived annotations do not bias toward certain computational prediction methods, we also include human PBMCs datasets with 10 cell subpopulations from a healthy donor, where the cell types were identified by FACS sorting [123].

The chosen datasets enable us to investigate different scenarios in terms of reference data selection. We conduct many tests with different scenarios for the reference and target data discrepancies, including the following:

- Individual difference: when reference and target data are from different individuals. In this case, the discrepancy only comes from biological variations.
- Condition difference: when reference and target data are from different condi-

tions, including protocol difference (10X Chromium vs. Smart-Seq2), sample collection difference (e.g., frozen and fresh tissues), lab effect (data generated by different laboratories), biological difference (e.g., different brain regions), and clinical difference (e.g., different disease status). These tests cover a wide range of biological, clinical, and technical discrepancies between reference and target datasets.

In addition to the discrepancies between reference and target, we also investigate the strategy of using a "pooled" reference: to combine data from many individuals with the same or different conditions together. Such a strategy can increase the reference data size and potentially average out the individual or condition variation ("pooling effect"). In order to distinguish whether the performance gain/loss comes from the increased reference data size or the pooling effect, we also perform downsampling on the reference data to make a fair comparison. Meanwhile, we are also curious about whether purifying the reference dataset can improve prediction performance. We adopt two strategies to remove "noisy cells" (cells that are not tightly clustered) in the reference and investigate the prediction performance with the purified reference.

2.2.4 Evaluation metrics

We use three metrics to evaluate the prediction results and benchmark the computational performance of all methods:

- Accuracy (Acc), which is the proportion of correct cell type assignments among all cells, directly evaluates the overall final celltyping accuracy.
- Adjusted Rand Index (ARI), which evaluates the clustering similarity between ground truth and prediction, without considering the accuracy of the assignment of cell types for clusters.

- Macro F1, which is a harmonized factor weighing precision and recall rate while considering all classes having equal contributions. It is a suitable metric when the cell type proportions are highly imbalanced.

More detailed information can be found in our published paper if interested [73].

2.3 Results

We evaluate all the combinations of the aforementioned factors in the supervised cell-typing: different prediction methods, feature selection methods, and choices of reference data. Overall, we obtain results for 29 predictions, 6 feature selection strategies, 9 prediction methods, and 4 metrics (including running time as an additional metric), which produce a total of over 5000 results.

2.3.1 F-test on reference datasets along with MLP achieves the best overall performance

We first evaluate the overall impact of different feature selection and prediction methods across all experiments. Since each experiment has a different baseline performance, i.e., the prediction accuracies are higher in some experiments than others, we remove such baselines to compute the performance gains or losses merely induced by feature selection and prediction methods. By doing so, the results from all experiments can be summarized altogether. More details about the procedure are provided in Appendix A Section 1.

We summarize the performance gains/losses of all combinations of feature selection methods and classifiers in Figure 2.1. The heatmap shows the results for the combinations, and the boxplots on the sides show the marginal gains/losses from each feature selection and classifier alone. The heatmaps are sorted by the average values of the rows and the columns so that the entry in the top left corner represents the best overall performer. For example, the vertical boxplots in Figure 2.1A show that the median gain in the accuracy of using MLP as the predictor is 0.053, and the horizontal boxplots in Figure 2.1A show that the median gain for using F-test on reference data to selection feature is 0.013. The heatmap shows that combining F-test on reference and MLP, which is the best combination, provides a gain of the accuracy of 0.09. Overall, we observe that using F-test on reference data as a feature selection method is the best, whereas using Seurat on reference data, Seurat on target data, and no feature selection are among the worst performers. The results also reveal that FEAST produces better feature selection than Seurat not only in unsupervised clustering tasks [103] but also in supervised celltyping. In terms of classifiers, MLP is the best overall, but SVM with both linear and RBF kernels provides comparable results. These results are consistent with the ones reported in [1], where the SVM with rejection has the best performance. These conclusions in general hold for other metrics (ARI and Macro F1), only that the SVM with linear kernel has a slight edge over MLP in Macro F1. Among the two semi-supervised methods, ItClust performs reasonably well and ranks 3rd when using ARI as measurement, only slightly behind MLP and SVM. MARS has poor performances based on our tests: it ranks last on average accuracy and ARI, and the results are highly variable, indicating poor robustness.

2.3.2 Impact of data preprocessing

To alleviate the noises in scRNA-seq data, a number of methods have been developed for scRNA-seq data preprocessing, including batch effect removal and missing data imputation. We perform a series of analyses to evaluate whether the preprocessing helps supervised cell type identification.

We first evaluate the impact of missing data imputation. In a recent study [42], several imputation methods were evaluated to assess the accuracy and the usability

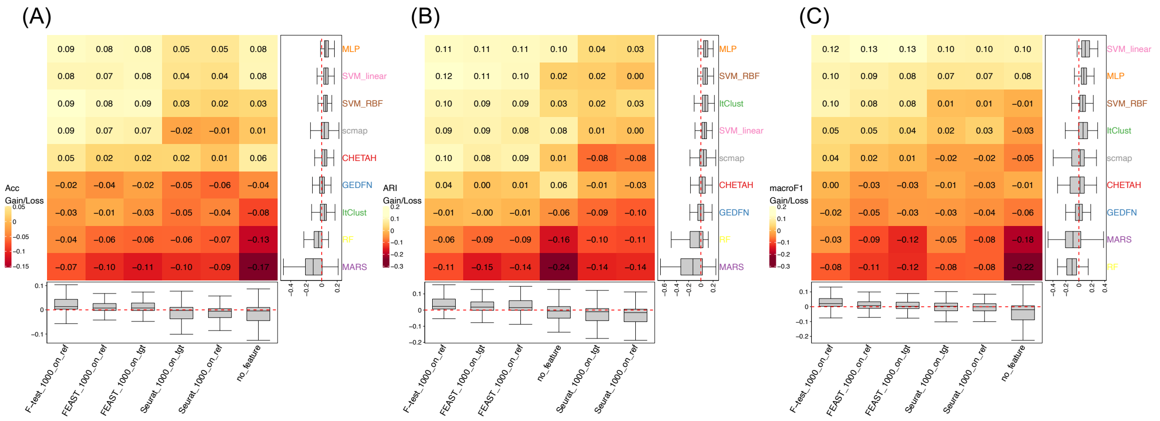


Figure 2.1: Prediction performance gains/losses with different combinations of classifiers and feature selection strategies on all experiments. Accuracy. B ARI. C Macro F1. The performance gains/losses for all combinations are illustrated by the heatmap. The heatmaps are sorted by the average values of the rows and the columns, and thus, the entry at the top left corner represents the most performance gain combination. The boxplots on the right and bottom sides illustrate the marginal performance gains/losses from classifiers and feature selection methods. The red dotted lines in the boxplots are reference lines at 0 (no gain nor loss).

of downstream analysis. We choose three outperforming methods MAGIC (smooth-based) [113], SAVER (model-based) [44], and scVI (data reconstruction based on deep learning) [68] to impute both reference and target datasets and then train a classifier for cell type prediction. Since we observe that the MLP classifier with F-test feature selection produces the best prediction, we only evaluate the impact of imputation on this combination. Our results (Appendix Figure A.1) indicate that no imputation method steadily outperforms the one without imputation under all scenarios. Thus, we believe that imputation may not be a necessary preprocessing step for supervised celltyping.

We next evaluate the impact of batch effect removal. There are several methods specifically designed for scRNA-seq to remove the batch effect, and they are comprehensively compared in [109]. Here, we apply two popular batch effect removal methods: Harmony [53] and fastMNN [38] on the data, and again compare the prediction performance to the original ones without removal. The same as in imputation, we only perform such comparison on the MLP with F-test combination. Our results (Appendix Figure A.2) show that there are no significant differences with or without removing the batch effect. In fact, the ones without batch effect removal have slightly better performances in most cases. Therefore, we conclude that batch effect does not affect the prediction performance and the correction may not be required, and we directly concatenate the datasets to perform the following analyses.

2.3.3 Condition effect

Next, we want to know how the difference between the reference and target datasets will affect the prediction. As stated in the previous “Datasets” section, we categorize the discrepancies between the reference and target datasets into individual effects and condition effects. In our definition, the individual effect describes individuals from the same dataset under the same technical and clinical conditions, so the difference between reference and target data only comes from biological variations. The condition effect is broader, including technical artifacts such as batch effect as well as other biological and clinical condition differences. Thus, the impact of the individual effect should be considerably smaller than the condition effect. In our design, we use individual effects as a baseline and benchmark different types of condition effects toward it. Within this section, we only present the results from using F-test on the reference dataset for feature selection and using MLP as the classifier, since they are proven to have the best results in previous sections.

Our analysis and results from mouse brain and human PBMCs datasets reveal several important points. First, the individual effects (caused by biological variance) are small, evidenced by the best performance from using subjects from the same dataset under the same condition as a reference. Secondly, the biological effect is also not significant for predicting major cell types, e.g., using the hippocampus from the same dataset as a reference can accurately predict major cell types in the frontal cortex. This indicates the similarities in the gene expression profiles of major cell types between the frontal cortex and hippocampus and that the major cell type differences are much stronger than the brain region differences. However, despite the individual effect and the region effect both achieving high performances, these two cases are impractical in real data scenarios since most of the prediction will happen across datasets in practice. When predicting across datasets, the performance becomes worse. This is reasonable since the dataset effect contains both technical and biological/clinical effects. However, our results indicate that the performance reduction in predicting major cell types across datasets is not severe: the accuracy only drops by less than 0.02 in both datasets. When reference and target data have significant clinical differences, there will be some but not dramatic performance reductions. In general, we conclude that the dataset difference, when there is no strong clinical difference, does not have a significant impact on predicting major cell types. With clinical differences, one should expect some performance reductions. However, in those cases when investigators cannot find the reference data with a matching clinical condition, using data from normal control as the reference is not a terribly bad idea.

2.3.4 Pooling references improves the prediction results

After obtaining a better understanding of how the discrepancies between reference and target datasets affect the prediction, a natural thought is to combine reference datasets to reduce bias. To validate this, we perform reference dataset "pooling" to investigate whether it can improve the prediction. We fix the target dataset as one subject and pool data from multiple individuals and different conditions to create a larger reference for prediction. In order to understand whether the prediction improvement is from the increased reference data size or data pooling effect, we also down-sample the pooled reference to eliminate the reference size effect. We choose the results from individual effects as baselines in these comparisons. Reference pooling is conducted under both intra-dataset and inter-dataset scenarios. We choose individuals or subjects from “Human PBMCs lupus,” “Mouse brain FC” with major cell types, and “Mouse brain FC” with sub-cell types to perform intra-dataset prediction. For inter-dataset prediction, we use mice from “Mouse brain FC” to predict mice in “Mouse brain pFC.” More details about the dataset selection and processing are provided in Appendix A Section 2.

Individual effect, pooling effect, and downsampled pooling effect

As shown in Figure 2.2A–C, under intra-dataset setting, combining individuals together (black line) achieves significantly higher overall accuracy compared to the other two strategies in all datasets. The same trends can be observed in ARI and Macro F1 (Appendix Figure A.3). As for the down-sampling strategy, we can also observe a slight increase in the mean performance with lower variance, indicating that the benefit of pooling is not only from the increased reference data size. Another finding from the figures is the significant increase in performance when predicting sub-cell types in the mouse brain dataset (Figure 2.2C). This indicates that, for a large number of sub-cell types, an increased sample size is particularly beneficial. Figure 2.2D and Appendix Figure A.3D show the comparison under the inter-dataset setting, where “pooling” brings slightly better performances, similar to that in the intra-dataset experiments.

Pooling reference from different conditions can improve the prediction results

Next, we wonder how “pooling” subjects with different conditions will impact the prediction performance. We combine subjects from different brain regions and different datasets in mouse brain data, as well as individuals from different batches and

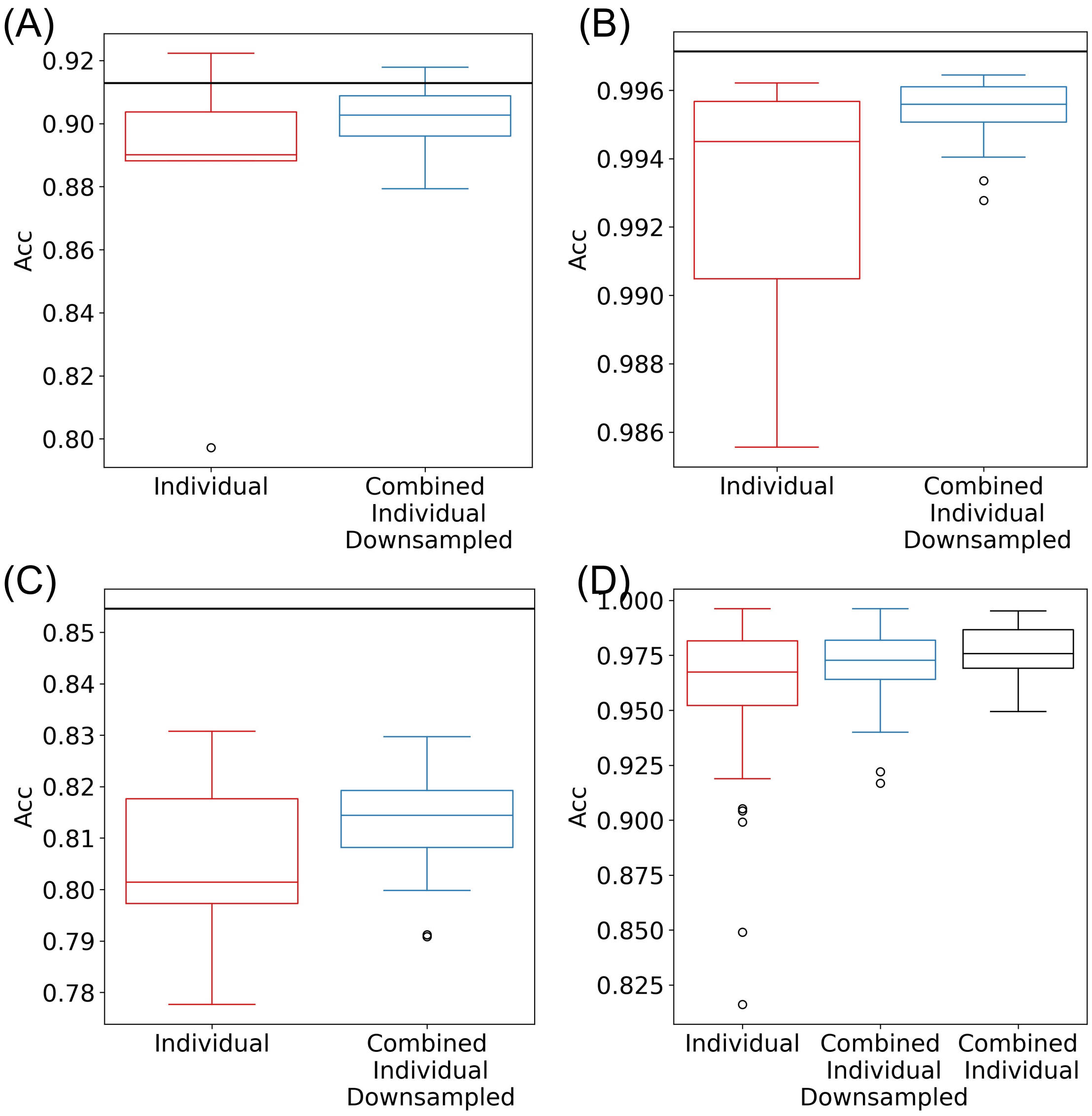


Figure 2.2: Impact of "pooling" on individual effect under intra-dataset and inter-dataset scenarios. Accuracy comparisons among individual effect (red box), down-sampling strategy (blue box), and "pooling" all individuals (black line for intra-dataset, black box for inter-dataset). **A** "Human PBMCs lupus": 8 lupus patients from batch 1 under the same condition. **B** "Mouse brain FC" major cell types: 7 mouse subjects from the same frontal cortex region under the same condition. **C** "Mouse brain FC" sub-cell types: 7 mouse subjects from the same frontal cortex region under the same condition. Down-sample boxes in A–C each contains 30 results. **D** "Mouse brain FC" to predict "Mouse brain pFC" on major cell types: use 7 mouse subjects respectively from "Mouse brain FC" as a reference to predict 6 mouse subjects from "Mouse brain pFC". Individual effect box contains 42 results; down-sample box contains 60 results, and "pooling" box contains 6 results.

clinical conditions in human PBMCs. For the dataset effect, we first try to combine individuals in each dataset respectively and then merge the two datasets together to predict.

All accuracy and Macro F1 metrics are improved by combining individuals together (Table 2.1), although some experiments have slight drops in ARI. These results indicate that by "pooling" individuals together, some noises caused by individual variations can be averaged out. We also find when combining "Mouse brain pFC" and "Mouse brain cortex" to predict the target in "Mouse brain FC," we can achieve a better result in Macro F1 than combining individuals from "Mouse brain pFC" and "Mouse brain cortex," respectively. We further visualize the cell type annotations after combining the "Mouse brain pFC" and "Mouse brain cortex" using tSNE [112]. We observe the two cortex datasets actually do not blend well; instead, a cluster of interneurons and neurons from "Mouse brain cortex" is mixed together (Appendix Figure A.4). Even though there is a clear separation between datasets as shown in these tSNE plots, our results show that combining datasets can still improve the prediction performance.

After showing that pooling reference data can improve prediction performance, we wonder if there is a saturation point when we keep enlarging the reference datasets. We conduct three analyses in mouse brain data to investigate the pooling saturation point on three perspectives: (1) predict major cell type within the same dataset, (2) predict major cell type across different datasets, and (3) predict sub-cell type within the same dataset. The saturation analyses are done in two ways. We first combine cells from different numbers of individuals as a reference. Furthermore, we pool the cells from all individuals and randomly sample different numbers of cells as a reference. The second averages out the individual effects and only investigates whether there will be saturation with more cells in the reference. More details about datasets used in this analysis are provided in Appendix A Section 2, and analysis

Table 2.1: Before and after “pooling”

	Before (mean Acc)	After (Acc)	Before (mean ARI)	After (ARI)	Before (mean Macro F1)	After (Macro F1)
Mouse brain region effect	0.993	**0.996**	0.995	**0.997**	0.915	**0.959**
Mouse brain dataset effect (pFC)	0.971	**0.988**	0.947	**0.967**	0.892	**0.904**
Mouse brain dataset effect (cortex)	0.977	**0.982**	0.955	**0.965**	0.918	**0.927**
Mouse brain dataset effect (combine pFC and cortex)	-	**0.986**	-	**0.963**	-	**0.933**
Human PBMC lupus batch effect	0.872	**0.893**	0.779	**0.789**	0.717	**0.790**
Human PBMC lupus clinical difference	0.838	**0.850**	0.724	0.716	0.673	**0.701**
Human PBMC lupus (combine batch effect and clinical difference)	-	**0.896**	-	**0.782**	-	**0.790**

Performance comparisons between before and after “pooling” individuals with condition effect. “–” indicates the data is unavailable. The bold data indicates a performance improvement.

details are provided in Appendix A Section 3.

We notice that for major cell type prediction (Appendix Figure A.5A, B), performance saturation clearly exists with larger reference data. For sub-cell type prediction (Appendix Figure A.5C), we do not observe a clear saturation point, and it is likely that the performance can further improve with a larger reference. The low signal-to-noise ratio among the subtypes requires an even larger reference for us to observe the saturation. Another finding from this analysis is that pooling individuals can potentially lead to faster saturation. This is very pronounced in sub-cell type prediction (Appendix Figure A.5B). The right panel shows that the performance is saturated from the start (3000 cells) when cells are sampled from a pool of individuals. When adding each individual at a time (left panel), it requires 4 individuals (around 40,000 cells) to reach saturation. These results are consistent with our findings that pooling individuals can achieve better prediction performance.

2.3.5 Purifying references does not improve the prediction results

Furthermore, we investigate whether purifying the reference dataset can achieve better predictions. Intuitively, cells on the edge of the cluster can be easily misclassified, and including them in the reference can contaminate the signals. We adopt two strategies for purifying the reference data: (1) Euclidean distance-based and (2) probability-based. The distance-based purification first computes the centroids for each cell cluster and then removes the 10% of cells with the largest distance to the centroid. For probability-based purification, we first adopt an SVM classifier with RBF kernel to fit the reference data and then generate probability scores of each cell belonging to cell types. For each cell type, 10% of cells with the lowest probability scores are removed. We conduct both purifica- tions in four designed analyses. More details can be found in Appendix A Section 2.

We first visualize those cells removed from the reference dataset (Appendix Figure A.6) and find that distance-based purification evenly removes cells on the edge of the clusters while probability-based purification removes more cells lying in between different clusters. Table 2.2 presents the overall accuracies of the four comparisons before and after purification. The results vary in different analyses. In predicting mouse brain sub-cell types, both purifications only lead to slightly improved performances. The reason might be that the purification removes wrongly labeled cells and increases the separations among cell clusters. Overall, cell purification does not improve the performance when predicting major cell types because the outliers of cell clusters do not have a large impact on assigning labels. However, when there exist sub-cell types, outliers among cell clusters act as noises, and by removing those, the prediction can be slightly improved.

Table 2.2: Before and after "purification"

	Original performance	After distance-based purification	After probability-based purification
Human PBMC lupus: one individual predicts another individual	0.797	0.748	**0.799**
Human PBMC lupus: one batch predicts another batch	0.924	0.920	0.921
Human PBMC lupus: one status predicts another status	0.931	**0.932**	**0.934**
Mouse brain FC: one subject predicts another subject (sub-cell types)	0.783	**0.813**	**0.802**

Performance comparisons before and after purifications on reference dataset. Here, we only demonstrate the performance of overall accuracy. There are in total four experiments: (1) "Human PBMC lupus" one individual predicts another individual: uses one lupus patient from batch 1 to predict another from the same batch under the same condition; (2) "Human PBMC lupus" one batch predicts another batch: uses samples from batch 1 to predict samples from batch 2 under the same condition; (3) "Human PBMC lupus" one status predicts another status: uses lupus samples from batch 2 to predict IFN-β stimulated samples from the same batch; and (4) "Mouse brain FC" one subject predicts another subject: uses one mouse subject to predict sub-cell types of another subject from the same region and the same dataset. The bold data indicates a performance improvement

2.3.6 Computational performance

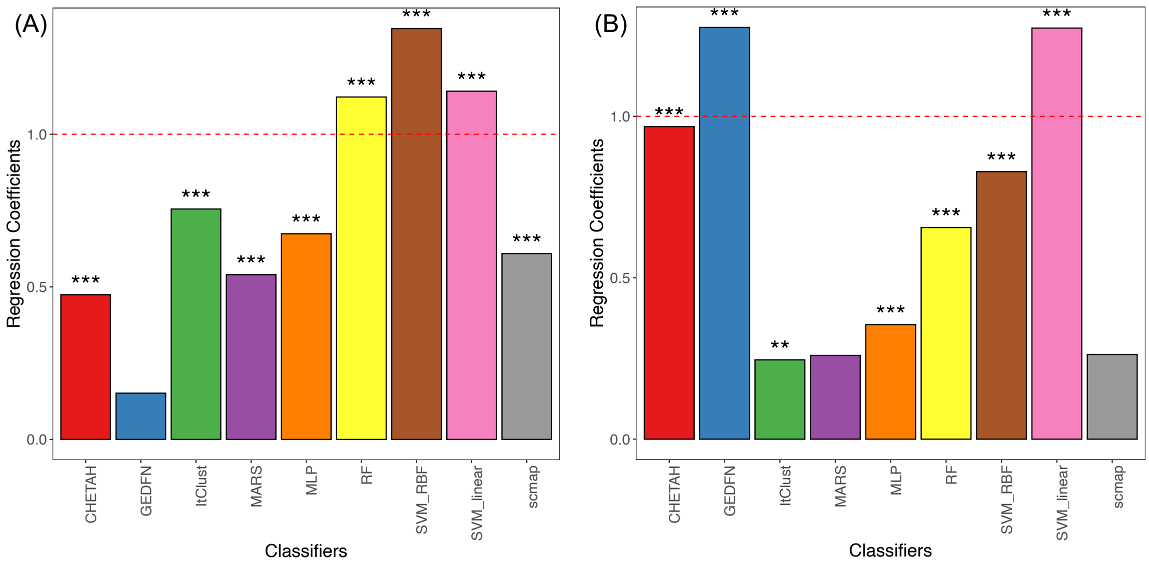


Figure 2.3: Computation performance of each method. The horizontal dotted red line denotes 1 and indicates a linear relation. The star denotes the p-value of the estimates (**p-value < 0.01; ***p-value < 0.001). **A** Regression coefficients of each method describe the relationship between training time and reference data size. As shown in the figure, the training time of SVM and the random forest grows faster than the increase of reference data size, and all others are slower. Among all classifiers, the coefficient estimation of GEDFN is not significant. **B** Regression coefficients of each method describe the relationship between training time and the number of cell types. The training time of GEDFN and SVM with linear kernel grows faster than the increase in the number of cell types. Coefficient estimations of scmap and MARS are not significant.

Besides prediction performances, we also keep records of the training time for each experiment. The training time can be affected by both reference size and the number of cell types, which are moderately positively correlated with the Pearson correlation coefficient being 0.44. To fully evaluate how training time is affected by each classifier, we construct a linear regression model using the log-transformed training time t as a response and the log-transformed reference size s along with the log-transformed number of cell types c as explanatory variables. This regression model can be further denoted as $log(t) \sim \beta_1 log(s) + \beta_2 log(c)$. We estimate β for each classifier. If β is greater than 1, the training time grows faster than linear and vice versa. Regression coefficients are summarized in Figure 2.3. It shows that RF and SVM with both kernels have the worst computational performances in terms of training data size, and GEDFN and SVM (linear kernel) are the worst in terms of the number of cell types. Overall, ItClust and MLP show the best scalability with the coefficients of both reference size and the number of cell types being less than 1. This might be caused by the design of the loss function which directly takes all classes into account. Compared to scmap, another correlation-based method, CHETAH, is largely affected by the number of cell types because it adopts a hierarchical structure and needs to derive gene profiles for each branch until discovering all cell types. With these observations, we again promote the usage of MLP with its comparably better performance and high scalability. When "pooling" all cells together, training an MLP classifier will not consume too much training time. Once the classifier is trained, parameters can be stored and directly used for predicting cells in the newly generated scRNA-seq datasets. Prediction can be done in a very short time period.

2.4 Discussion

Supervised celltyping for scRNA-seq has gained tremendous interest in recent years, and we believe it is the direction to go for identifying cell types in scRNA-seq data. In this paper, we comprehensively evaluate several important aspects of supervised celltyping: feature selection, prediction method, data preprocessing, and the impact of discrepancies between reference and target dataset, which are important choices to make for investigators. Even though there are a number of methods and several comparison studies, no one has investigated the combined effects of these procedures, in particular, the choice of reference datasets. Moreover, we also investigate the strategies for processing reference data, including reference pooling and purification.

Based on our results, we make the following main recommendations. First, applying F-test on the reference dataset to select features and using MLP as a classifier is the best performer overall when the reference data is reasonably large (e.g., > 5000 cells). In fact, MLP can be replaced by SVM with either linear or RBF kernel, which produces comparable results. However, due to the computational burden of training SVM, especially in large datasets or having many cell types, we recommend using MLP. When the reference data is small, using a correlation-based method such as scmap is a better strategy. We consider certain pre-processing (e.g., imputation and batch effect correction) steps unnecessary because we do not observe a significant performance increase by doing so. Secondly, it is always desirable to pick reference data with matching biological and clinical conditions with the target data. However, the discrepancy between the reference and target data only has a slight impact on predicting major cell types. Thus, it is not terribly bad to use a reference dataset with slight condition differences, even though significant clinical differences could lead to non-trivial performance reductions. Thirdly, pooling references from different datasets improves the prediction results. This is not only because of the increased reference data size but also because that pooling can average out some biological and technical variations (evidenced by our downsampling prediction results and pooling saturation analysis). In the pooling saturation analyses, we sometimes observe that adding certain cells or individuals may result in worse performance (Appendix Figure A.5C). This leads to an important question on how to assess and select high-quality references. This is beyond the scope of this work, but we will explore it in the near future. Moreover, we find that purifying the reference data does not significantly improve the results, so we recommend against such a procedure.

From our investigations, the major cell type prediction is a relatively easier task as all analyses achieve satisfying results. However, supervised prediction for sub-cell types is much more difficult mainly due to the inconsistent subtype definition from different datasets. In fact, it is highly possible that the sub-cell types are indeed different under distinct biological and clinical conditions. Due to these reasons, we recommend against supervised prediction for sub-cell types and suggest a two-step hybrid approach: first applying a supervised prediction for major cell types and then using unsupervised approaches for subtypes. This might require further development and evaluation of the unsupervised clustering methods for similar sub-cell types since most current methods focus on clustering major cell types. When the goal of an unsupervised method is to distinguish many very similar cell subtypes, we might need new algorithms for feature selection, cell clustering, and new/rare subtype identification. It is worth mentioning that even though the two semi-supervised methods we test (ItClust and MARS) do not perform as well as the supervised ones in predicting known cell types, they have the potential advantage to discover new cell types, which could be useful in subtype prediction.

With the increased application of scRNA-seq, especially in large-scale, population-level studies, cell type identification continues to be one of the most important questions in scRNA-seq data analysis, for which we believe the supervised celltyping method will be a better answer. We perform extensive evaluations on several important factors in such an approach and provide some recommendations. More importantly, we evaluate the impact of the reference data and potential strategies for processing reference data, which have never been done before. Our study not only provides performance evaluation and recommendations but also points out potential research directions in this field.

Chapter 3

Cellcano: a supervised celltyping method for scATAC-seq

3.1 Introduction

Gene expression can be regulated by several factors. Among them, chromatin accessibility is essential for the interaction between DNA and regulatory elements and provides important information for understanding the transcriptional regulatory mechanism [110]. Recent years have also witnessed the shift from measuring chromatin accessibility in bulk samples to single-cell level by single-cell sequencing assay for transposase-accessible chromatin (scATAC-seq) [14]. Like in scRNA-seq, celltyping is also an important question in scATAC-seq data analysis. However, scATAC-seq data have certain characteristics that make celltyping more difficult. First of all, scATAC-seq data are much sparser due to low read counts [7], which results in weaker signals for distinguishing cell types. Secondly, unlike scRNA-seq, feature space is not well-defined in scATAC-seq data, which poses difficulties in extracting useful information. The raw scATAC-seq data can be summarized to counts on genome-wide fixed-size bins, peaks representing the accessible regions, or genes [19]. The determination of feature space is an additional important step in scATAC-seq celltyping. Although it is possible to do celltyping through experimental procedures such as Fluorescence-activated cell sorting (FACS) [26] or leveraging information from multi-omics sequencing techniques such as SNARE-seq [21], these datasets are expensive and limited. Therefore, methods specifically developed for scATAC-seq celltyping are in urgent need.

Most existing computational celltyping methods are unsupervised and based on prior knowledge [11] [119] [8] [101]. As of now, a lot of methods have been developed for single-cell omics integration 1 while limited methods have been specifically developed for scATAC-seq celltyping. Seurat V3 [101] and scJoint [63] use scRNA-seq datasets as references to transfer cell labels to scATAC-seq. Due to the strong data distributional shift between different measurements, the two methods can significantly underperform. Although SnapATAC [30] performs analysis with scATAC-seq datasets as references, the functionality is not implemented in the pipeline. Only recently, EpiAnno was published to perform supervised celltyping in scATAC-seq using scATAC-seq as reference using peaks as input [22]. A major problem is that the peaks are not well-defined and are highly data-dependent. Due to technical and biological artifacts, concordance of peaks can be low between reference and target [31], which would result in a loss of information and undesirable celltyping results. Additionally, EpiAnno is not scalable for large datasets.

In this work, we develop a novel computational celltyping method for scATAC-seq, named Cellcano. Cellcano implements a two-round supervised learning algorithm. It first trains a multi-layer perceptron (MLP) on the reference dataset and predicts cell types in target data. From the prediction results, we can acquire the prediction probabilities where we can further compute entropy. We select cells with lower entropy as anchors to form a new training set. Next, Cellcano trains a self-Knowledge Distiller model (KD model) [66] on anchors using the predicted pseudo labels and predict cell types in remaining non-anchors. The KD model alleviates noises in anchors by softening the label distribution. Through extensive real data analyses, we demonstrate that Cellcano is significantly more accurate, computationally efficient, and scalable compared to existing methods. Cellcano is well-documented and freely available at `https://marvinquiet.github.io/Cellcano/`.

3.2 Methods

3.2.1 Cellcano model

Cellcano takes scATAC-seq raw data (fragment files or bam files) as inputs and calls ArchR to generate the gene score matrices (details in Appendix B Section 1). Assume there are G genes and N cells in the reference, and M cells in the target data, we define the gene score matrices in reference and target data as $X_{ref} \in \mathbb{R}^{G \times N}$ and $X_{tgt} \in \mathbb{R}^{G \times M}$, respectively. In the reference gene scores, we first perform a feature selection step to select representative features. The features are selected by F-test with known cell type labels, represented as $C_{ref} \in \mathbb{R}^{N \times 1}$. We have previously shown that features selected by F-test in reference data can provide the best results in supervised scRNA-seq celltyping [73]. By default, we select the top 3,000 genes with the largest F-statistics. We obtain the reference and target gene scores for the selected features and perform data normalization. To be specific, we normalize the cell-wise gene scores so that the total gene scores sum to 10,000 for each cell. We then take log-transformation on the normalized gene scores plus 1. After that, we perform gene-wise standardization on the log normalized gene scores so that each gene will have a zero-mean and unit-variance charateristic. The standardization is a recommended procedure for performing efficient backpropagation in neural networks [58].

In Cellcano's first-round prediction, we first train an MLP model with a ReLU activation function to capture the non-linear mapping between the X_{ref} and C_{ref}. For a multi-class classification with K cell types, the cell type label C_{ref} is one-hot encoded to a binary matrix with dimension $N \times K$. The one-hot encoding labels the corresponding class as 1 and all others as 0 for each cell. The last layer of MLP is connected to a softmax function to convert the outputs from the last layer of the MLP to probabilities. The softmax function is represented by

$$\sigma(Z_i) = \frac{exp(\frac{Z_i}{T})}{\sum_{k=1}^{K} exp(\frac{Z_k}{T})}.$$

Here, Z_i represents the outputs from the last layer of the MLP, and T is a hyperparameter representing the temperature of the softmax function. The larger the T is, the smoother the $\sigma(Z_i)$ will be. We set $T = 1$ in the first-round MLP model. During training, we use cross-entropy as the loss function to minimize the distributional difference between the one-hot encoded cell type label p and the predicted cell type probabilities $\sigma(Z)$:

$$H(p, \sigma(Z)) = -\sum_{i=1}^{N}\sum_{k=1}^{K} p_{ik} log(\sigma(Z_i)_k).$$

After training the MLP model, we apply the trained MLP model to the target data to obtain the probabilities for each cell being in each cell type.

When the target data size is small, Cellcano takes the class with the largest probability as the final predicted cell type for each cell and stops. When the target size is large (over 1,000 cells by default), we perform a second-round prediction. We first select anchors from the target, and we aim at selecting accurate anchors which can also capture the full scope of target distribution to guide the second-round prediction. With the first-round predicted probabilities, denoted as q_{ik} for cell i being in cell type k, we calculate the entropy $E^{M\times 1}$ for all M cells as:

$$E_i = -\sum_{k=1}^{K} q_{ik} log(q_{ik}).$$

When a cell label is more confidently assigned, its entropy over the predicted probabilities is lower, and the prediction is in general more accurate. Once we have entropies for all cells, we select 40% cells with the lowest entropies as anchors for each cell type to form the new reference dataset for second-round training. This can assure the existence of every cell type in the anchors, as well as keep the cell type proportion consistent between anchors and non-anchors in the target data. Since some anchors will be mistakenly predicted, we apply the KD model in the second-round training to deal with the issue, detailed in next section. The model trained in the second round will be used to predict cell types for non-anchors. Finally, we combine the cell types predicted for the anchors (from the first round) and non-anchors (from the second round) as our final cell type calls.

3.2.2 The Knowledge Distiller model

Although the anchors cannot be perfectly predicted from the first round, they are important complementary training data for improving prediction, since these cells are from the exact same target domain where we previously lack supervision. To deal with training data with noisy labels, we implement a self-Knowledge Distiller (KD) model in the second-round training. The KD technique was originally proposed to transfer the knowledge learned from a sophisticated teacher model to a light-weighted student model, by treating the prediction results produced from the teacher model as the "soft labels" for training the student model [41]. Inspired by this and several recent works [66] [118], we propose to use the teacher-student interaction to alleviate the noisy label problem. Specifically, the teacher model distills knowledge from both clean supervision and noisy supervision by producing "soft labels" as the training targets of the student model. Compared to the "hard labels" that only contain over-confident 1's and 0's, "soft labels" are smoothed and thus more noise-tolerated [80]. Also, there are cell types sharing similar profiles during celltyping which fits the fine-grained classification setting in the KD model. In Cellcano, we apply a "self-KD model" where we have the exact same structure for the teacher model and the student model. We set them to be vanilla MLPs of two hidden layers with 64 and 16 nodes respectively. To let the model be more generalizable, we put the dropout layer right after the input layer. We use ReLU as the activation function.

We first train the teacher model with the anchors as input. To make the label "softer", we set the temperature T of the softmax function to be larger. We use the cross-entropy loss for the teacher model, then train the student model with the teacher's "soft labels" as well as the one-hot encoded "hard labels". The idea is to learn a label smoothing regularization so that the label distribution can be better captured. The KD loss function for the student model is a weighted average of two losses, which is shown in the equation below:

$$L_{KD} = \alpha H(p, q_s^{T_1}) + (1 - \alpha) KL(q_t^{T_2}, q_s^{T_2}).$$

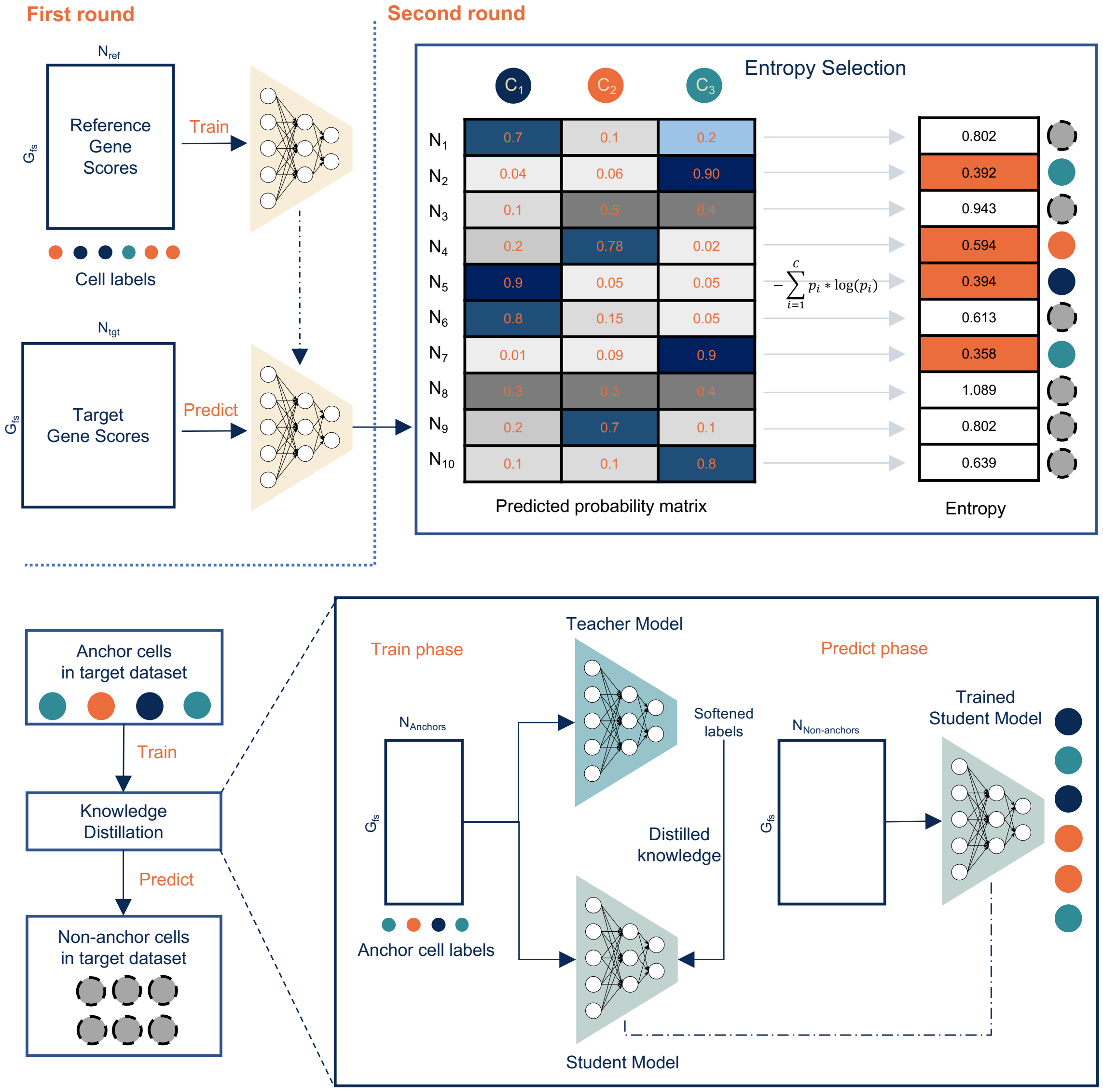
Here, T_1 and T_2 are temperatures in the softmax functions, and α is a hyperparameter for balancing the two losses. The first part of the KD loss is a cross-entropy loss where the student prediction q_s is guided by "hard labels" (anchor cell types from first-round prediction), and we set the T_1 as 1. The second part represents the Kullback-Leibler (KL) divergence loss which measures the probability distribution distances between the soft teacher prediction q_t and the soft student prediction q_s, where T_2 can be adjusted. We set $T_2 = 3$ for the second part to soften the label distribution. Overall, we set α as 0.1 to value more on the teacher model's "soft labels". The KD model is trained for 30 epochs.

More detailed information about the overall scheme of data processing and analysis, methods that have been benchmarked and evaluation metrics can be found in the Cellcano publication [75].

3.3 Results

3.3.1 The Cellcano framework

Cellcano uses gene-level summaries of the raw scATAC-seq data as inputs. Given the raw data, Cellcano incorporates ArchR [35] pipeline to process the raw data and obtain gene scores for both reference and target datasets. The choice of input is carefully investigated, and the results show that using gene scores provides good prediction accuracy and computational efficiency (details in a later section). Then Cellcano applies F-test on reference gene scores to select cell-type-specific genes as features for model construction [103]. After obtaining the reference and target gene scores for the selected features, Cellcano adopts a two-round supervised celltyping strategy, shown in Figure 3.1. In the first round, Cellcano trains an MLP model with reference gene scores and predicts cell types in target data. If the target size is too small, Cellcano stops and returns the prediction results. When the target size is large enough (e.g., over 1,000 cells), Cellcano performs another round of model training to improve the prediction results. The second round starts with selecting anchor cells. For that, we first calculate entropy for each cell based on the prediction probabilities from the first-round prediction and then select cells with lower entropies as anchors. The assumption is that the cells with lower prediction entropies are more likely to be accurately predicted. We carefully investigate the anchor cell properties and their impact on the prediction results (details in a later section) and demonstrate that the assumption holds well in real data. We then use the anchors with their predicted cell types as new reference data to train another classifier to predict the non-anchor cells. Here, we use a KD model as the classifier since it works better when reference data have imperfect labels. The assumption in the second round is that the classifier trained on anchors (which are from the target data) can better capture the data distribution in the target dataset compared to the classifier trained on the reference dataset, thus improving the prediction performance.



	C_1	C_2	C_3	Entropy
N_1	0.7	0.1	0.2	0.802
N_2	0.04	0.06	0.90	0.392
N_3	0.1	0.5	0.4	0.943
N_4	0.2	0.78	0.02	0.594
N_5	0.9	0.05	0.05	0.394
N_6	0.8	0.15	0.05	0.613
N_7	0.01	0.09	0.9	0.358
N_8	0.3	0.3	0.4	1.089
N_9	0.2	0.7	0.1	0.802
N_{10}	0.1	0.1	0.8	0.639

Figure 3.1: Cellcano adopts a two-round prediction strategy. In the first round, Cellcano trains an MLP model on reference gene scores with known cell labels. Then, Cellcano uses the trained MLP to predict cell types on target gene scores. When the target size is sufficiently large, Cellcano starts the second round by selecting anchors. With the predicted probability matrix obtained from the first-round prediction, entropies are calculated for each cell. Cells with relatively low entropies are selected as anchors to train a knowledge distillation (KD) model. The trained KD model is used to predict cell types in remaining non-anchors.

3.3.2 The choice of using gene score as input

As mentioned before, scATAC-seq data can be represented in three different feature spaces: genome-wide fixed-size bins, peaks, and genes. Genome-wide fixed-size bins have a very large feature space, which poses a heavy computational burden. The peaks are not pre-defined and require additional steps in calling and unifying peaks. More importantly, since the peaks will be different for each dataset, one cannot reuse a pre-trained prediction model for new target data. In this work, we choose gene scores as input because they are well-defined and have a small feature space. Also, it is possible to further connect the model trained on gene scores to scRNA-seq models, and vice versa. There are different ways of summarizing gene scores [19] [35] and our first question is how to utilize these gene score models. In total, ArchR provides 54 variations of gene score models (details in Appendix B Section 2 and Section 3), and its recommended one is shown to be the most accurate to infer gene expression in matched scATAC-seq and scRNA-seq data. From the real data analysis, we show that using the ArchR-recommended gene score model achieves good celltyping performances from Cellcano.

We next evaluate Cellcano with the recommended gene score or fixed-size 500-bp bin counts as input in both human PBMCs and mouse brain celltyping tasks. The comparison of prediction performances from human PBMCs is shown in Figure 3.2a and Appendix Figure B.1a-b. The two types of inputs produce comparable prediction accuracies in most celltyping tasks, while results in ARI and macroF1 show that using gene scores is significantly better. In mouse brain celltyping tasks, Cellcano with gene scores as input is better than using fixed-size bins in 62 out of 63 prediction results (Figure 3.2b, Appendix Figure B.1c-d), except one in mouse brain celltyping task using ARI as measurement. Overall, these results demonstrate that using gene scores as inputs works better than using bin counts. In addition, the computational time for using gene scores as input is much shorter (Figure 3.2c).

Considering both computational and prediction performances, we decide to use the ArchR-recommended gene scores as Cellcano's default input.

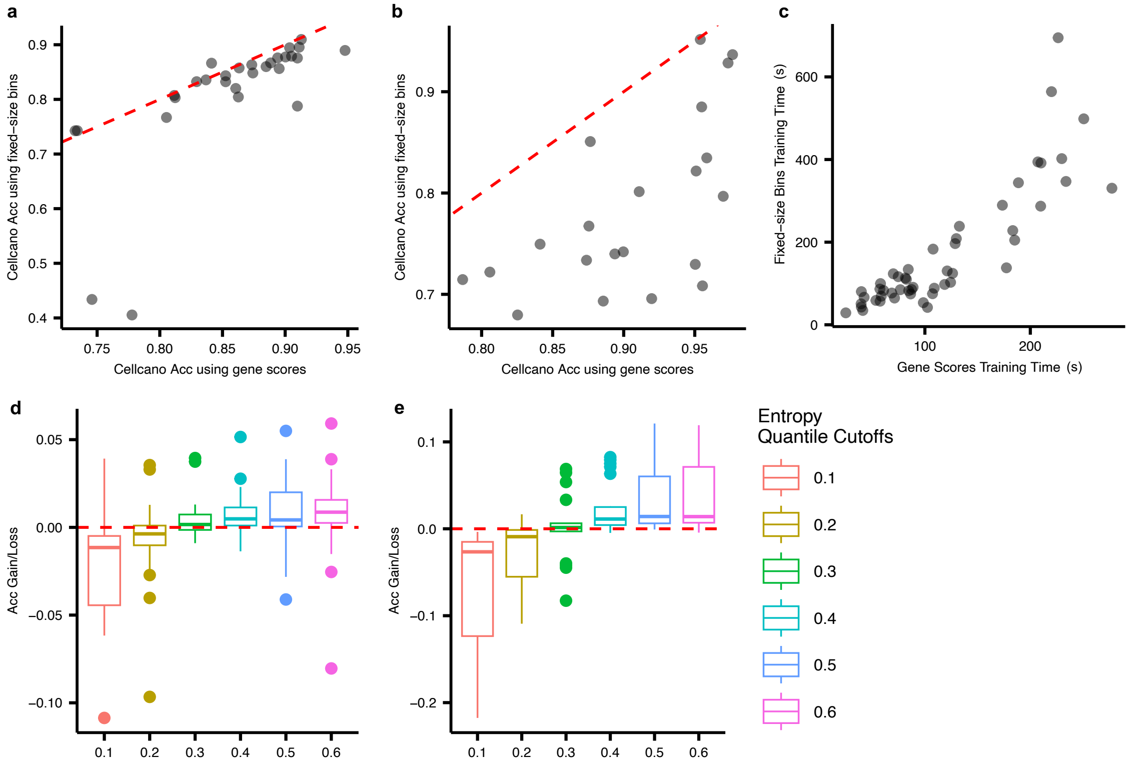


Figure 3.2: Focus on exploring performances between using different inputs for Cellcano. a, b Accuracies comparison on Cellcano using genome-wide fixed-size bins and gene scores as input from (a) n = 29 human PBMCs celltyping tasks and (b) n = 21 mouse brain celltyping tasks. The red dotted lines are identity lines. c Model training time comparison using the two different inputs on all n = 50 celltyping tasks. d, e demonstrate the selection of the appropriate number of anchors. d, e Accuracy gains/losses using different entropy cutoffs on (d) n = 29 human PBMCs celltyping tasks and (e) n = 21 mouse brain celltyping tasks.

3.3.3 Properties of Cellcano anchors

Cellcano selects anchor cells from the target dataset based on the prediction entropy from the first round (details in the Methods section) and uses them as reference to predict cell types for non-anchors in the second round. The number of anchors is specified by user as a cutoff for the quantiles of entropies. For example, when using

0.3 entropy quantile cutoff, 30% of the cells in the target dataset will be selected as anchor cells. As an exploration, we first compare the performance between anchors and non-anchors under different quantile cutoffs (0.1 to 0.6 with step size 0.1) in human PBMCs celltyping tasks and mouse brain celltyping tasks. Results show that the final prediction performance depends on a balance between anchor numbers and anchor accuracy.

We then summarize the final prediction performances using different entropy quantiles in human PBMCs celltyping tasks (Figure 3.2d, Appendix Figure B.2a-b) and mouse brain celltyping tasks (Figure 3.22, Appendix Figure B.2c-d). Each celltyping task has a prediction baseline which is calculated as the average performance by using different quantile cutoffs. We calculate the gains/losses using each quantile cutoff against the average performance. Overall, the performances are stable when using 0.2 or above as quantile cutoffs (the median Acc varies within -0.4%   +0.9% in human PBMCs celltyping tasks and -0.9%   +1.4% in mouse brain celltyping tasks). The worst performance occurs when using 0.1 as the quantile cutoff. This can be explained by the small training size in the second round and the failure of capturing the target data distribution. In conclusion, when using a moderate number of anchor cells, Cellcano can produce comparable prediction results. By default, we use 0.4 as the entropy quantile cutoff in our software implementation. Moreover, since Seurat also has an anchor selection step, we perform comparisons and show that Cellcano anchors are more accurate and can better capture the full scope of target data distribution.

Similar to Seurat, Cellcano selects anchors from the target dataset and uses them as references to predict cell types for non-anchors in the second round. However, the procedure for anchor selection in Cellcano is very different. Seurat uses Mutual Nearest Neighbors (MNN) in a low-dimensional space determined by canonical component analysis (CCA) to select anchors, which relies on the linear relationship between reference and target. The number of anchors selected is further determined by the parameter of how many neighbors are examined. Differently, Cellcano obtains predicted probabilities for cells in target data from the first-round MLP and then selects anchors based on the prediction entropies. The number of anchors in Cellcano is determined by the quantiles of entropies in each cell type.

3.3.4 Cellcano outperforms existing supervised scATAC-seq celltyping methods

We collect four human peripheral blood mononuclear cells (PBMCs) datasets and two mouse brain datasets (Appendix Table B.1) to benchmark the Cellcano. Among four human PBMCs datasets, one is cell-sorted by FACS and can be considered the "gold standard". The cell types in the other three datasets are annotated based on computational methods and prior biological knowledge, which are "silver standard" [49]. For the six datasets, we design 50 experiments, which comprehensively cover different real application scenarios (details in Appendix B Section 4).

After deciding the input data and the anchor numbers for Cellcano, we compare Cellcano with other supervised scATAC-seq celltyping methods. We benchmark Cellcano against six competing supervised celltyping methods: Seurat [101], scJoint [63], Signac [102], EpiAnno [22], ACTINN [72], and SingleR [3]. Even though Seurat and scJoint are not specifically designed for scATAC-seq celltyping using scATAC-seq data as reference, they can take gene scores as input for cell type prediction. For Signac, we follow its recently published scATAC-seq integration vignettes to first process raw scATAC-seq data into peak counts and then perform data integration along with label transfer. For EpiAnno, we use ArchR to call peaks and count reads overlapping the peak regions to generate peak-by-cell matrices as its input. ACTINN is a deep learning-based method that is very similar to the first-round prediction of Cellcano. SingleR is a correlation-based supervised scRNA-seq celltyping method. According to a recent survey study, SingleR is the second-best performer behind Seurat in scRNA-seq celltyping [45]. Even though ACTINN and SingleR are designed for scRNA-seq celltyping, they do not make any scRNA-seq-specific assumptions on the input data and thus can take the gene scores as input for scATAC-seq celltyping. We include them because we want to explore whether existing scRNA-seq supervised celltyping methods can be directly applied to scATAC-seq with gene scores as input. In addition, we also include another set of comparisons by first removing the batch effect between reference and target datasets and then using an MLP to transfer cell labels (details in the next section). We put all the results together to make direct comparisons of prediction performances. We evaluate the prediction performances from all methods by different metrics, including overall accuracy (Acc), adjusted rand index (ARI), macro F1 score (macroF1), Cohen's kappa (κ), median F1 score (medianF1), median precision, and median recall.

We first focus on the celltyping methods and compare the performances where we have one fixed gold standard target data (Figure 3.3a, Appendix Figure B.3). In total, there are seven celltyping tasks using different references. We order the boxplot according to the average performance. The results show that Cellcano achieves the highest average accuracy at 0.852 in the seven celltyping tasks, while scJoint is a close second with average accuracy of 0.837 and the third performer ACTINN has an average accuracy of 0.782. The accuracies from all other methods are significantly lower. For all other metrics (Appendix Figure B.3), Cellcano and scJoint in general have the highest performances compared to all other methods, consistent with the results in prediction accuracy. Overall, the third best performer is ACTINN which is a variation of Cellcano first-round prediction. The performance differences between Cellcano and ACTINN indicate the performance improvements by introducing our second-round prediction.

We then evaluate the performances in all other 22 human PBMCs celltyping tasks (Figure 3.3b, Appendix Figure B.4). Since the celltyping tasks involve different target

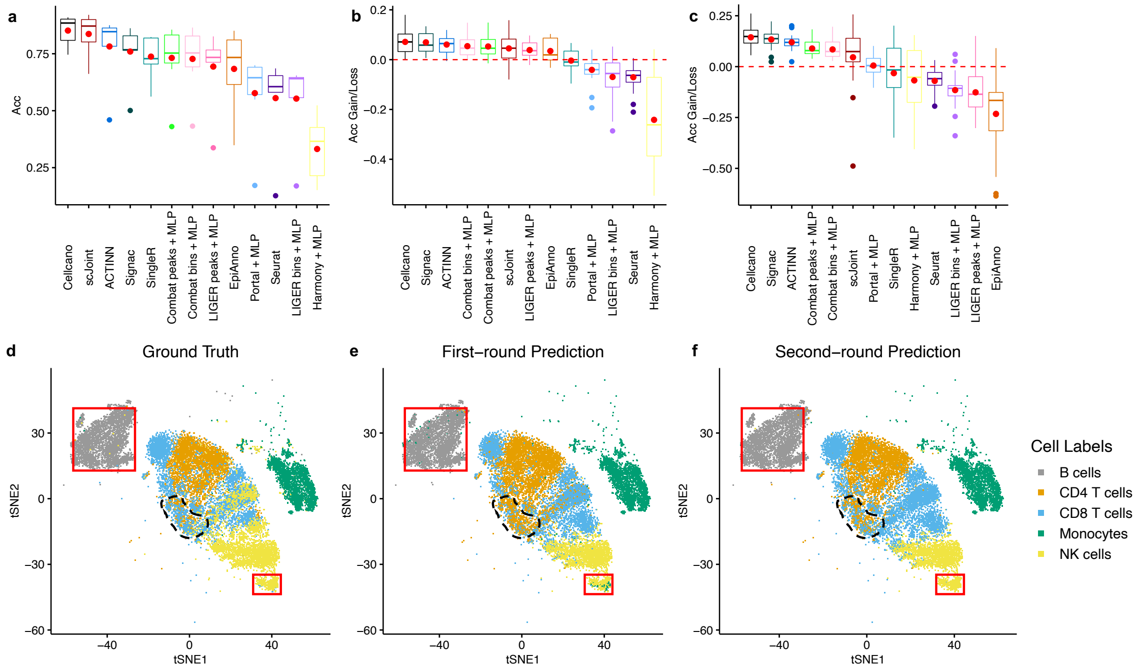


Figure 3.3: a–c Accuracy comparisons between Cellcano, Seurat, scJoint, Signac, SingleR, ACTINN, and EpiAnno along with other integration with label transfer methods on (a) $n = 7$ celltyping tasks using one human PBMCs FACS-sorted dataset as target, (b) $n = 22$ more human PBMCs celltyping tasks and (c) $n = 21$ mouse brain celltyping tasks. The boxplots are ordered to have the leftmost method with the highest average performance. d–f t-SNE plots from one of the celltyping tasks using FACS-sorted dataset as target that contains $n = 21,214$ cells. The cells are colored with (d) ground truth labels; (e) Cellcano first-round predicted labels; and (f) Cellcano second-round predicted labels. The highlighted areas illustrate Cellcano's ability to correct wrongly assigned cells predicted from the first round.

datasets, the baseline performance for each celltyping task can vary. We eliminate such baseline effects by computing the performance gains/losses for each method against the average. To be specific, we take the average of the prediction performances from all seven methods for each celltyping task, and then subtract the average from the performances for each method. From these experimental scenarios, Cellcano ranks first in average accuracy gain and average ARI gain whereas Signac ranks second. Signac slightly outperforms Cellcano in average macroF1 gain. Overall, ACTINN ranks third. Similarly, we evaluate the performances in 21 mouse brain celltyping tasks (Figure 3.3c, Appendix Figure B.5) and observe that Cellcano again outperforms all other methods with the most accuracy gain as 0.144. In the meantime, Signac acts as the second-best performer with an accuracy gain of 0.134 and ACTINN acts as the third-best performer with an accuracy gain of 0.120. Note that EpiAnno fails to generate results for two relatively larger (over 32k cells) celltyping tasks due to memory limit. Taking all 50 celltyping tasks together, we perform a paired t-test on Accuracy, ARI, and macroF1 in three comparisons: (1) Cellcano and ACTINN, (2) Cellcano and scJoint, and (3) Cellcano and Signac. The test statistics show that Cellcano performs significantly better than ACTINN (p-value: 4.857e-3), scJoint (p-value: 1.645e-3), and Signac (p-value: 0.023) in Accuracy. Results hold for all comparisons in ARI. For macroF1, Cellcano slightly outperforms ACTINN while largely outperforming scJoint and Signac. In summary, Cellcano outperforms all other methods considering all scenarios: two systems (human PBMCs and mouse brain), 50 celltyping tasks, and seven metrics.

To further demonstrate how the two-round procedure in Cellcano outperforms, we use one celltyping task (one FACS-sorted human PBMCs dataset as target, a combination of four individuals from Satpathy et al. [95] PBMCs dataset as reference) as an example to visualize the prediction results after each round by tSNE. Figure 3.3d labels the ground truth cell types provided by FACS. After the first-round prediction, some cells in B cell and natural killer (NK) are wrongly predicted as Monocytes (Figure 3.3e, red boxes). After the second round, the wrong predictions are corrected (Figure 3.3f, red boxes). Another observation is that many CD8 T cells on the boundary between CD4 T cell and CD8 T cell clusters (black dotted line area) are not correctly predicted. After the second round, most of these cells are correctly assigned back to CD8 T cells. Similarly in an example mouse brain celltyping task (Appendix Figure B.6), some inhibitory neurons are wrongly predicted as Astrocytes and Microglias are wrongly predicted as Oligodendrocytes after the first-round prediction and those are corrected after the second-round prediction (Supplementary Fig. 8, red boxes). These visualization examples demonstrate the advantage of having our second-round prediction with the KD model.

3.3.5 Cellcano works better than prediction with batch effect removed

A key advantage of the two-round approach in Cellcano is that training a model using anchors in target data alleviates the distributional shift problem between the reference and target data. The distributional shift is often caused by batch effect in high-throughput data. This leads to a question whether our two-round strategy is better than the one where we first remove batch effect and then apply a direct prediction. According to a recent benchmark study [70], LIGER [115] and ComBat [16] are the top performers when integrating scATAC-seq datasets. Although we have proven that using gene scores as input is the best choice for Cellcano, in this benchmarking study, genome-wide bins or peaks are suggested as inputs for the integration tasks. We therefore follow the suggestions and include four top-performing integration combinations into our comparison: LIGER with genome-wide bins as input, LIGER with peaks as input, ComBat with genome-wide bins as input, and ComBat with peaks as input as integration methods. We are also interested in knowing how batch-effect removed methods work with gene scores as input. Therefore, we added Harmony [53], which was demonstrated to have the best performance and shortest running time in previous batch-effect removal benchmark study in scRNA-seq data [109]. In the meantime, we also included Portal [122], a recently published integration method that has not been benchmarked and can take gene scores as input. After performing the integration between reference and target datasets, we apply MLP as the classifier to transfer cell labels according to the integrated output. We evaluate the prediction performances by Acc, ARI, and macroF1.

As mentioned earlier, we put all prediction results from celltyping and integration with label transfer into boxplots (Figure 3.3a-c, Appendix Figure B.3a-b, Appendix Figure B.4a-b, Appendix Figure B.5a-b) for a direct comparison. Since boxplots provide marginal distributions which represent the overall performances, we add heatmaps (Appendix Figure B.7, B.8) with original prediction performances to show a full-scope comparison. We categorize the heatmaps by different types of celltyping tasks and make the leftmost column have the highest average performance. When focusing on all integration with label transfer methods, ComBat with peaks as input and ComBat with genome-wide bins as input rank top, however, they do not outperform the top celltyping performers and thus are inferior to Cellcano. We generate a low-dimensional visualization before (Appendix Figure B.9a) and after the batch effect removal (Appendix Figure B.9b-e) on one example where one FACS-sorted PBMCs data is taken as target and four individuals from Satpathy et al. are combined as reference. We can observe that even when the batch effect removal methods work well on integrating reference and target datasets or integrating individuals (Appendix Figure B.9c, d using LIGER and Portal), the celltyping results are not necessarily better. In conclusion, these comparisons demonstrate that Cellcano can handle data from different individuals and batches in both reference and target data. Cellcano does not need to remove the batch effect and steadily outperforms other integration with label transfer methods. This provides the possibility of training prediction models using a large compendium of datasets.

3.3.6 Cellcano is computationally efficient and scalable

We evaluate the computational performance of Cellcano and show all celltyping methods' runtime for all celltyping tasks (Figure 3.4a-b). For fair comparisons, we combine the training time and prediction time into an overall runtime for Cellcano and Epi-Anno. This is because all other methods need both reference and target datasets as input to do prediction. Here, we do not consider the data pre-processing time (such as the time used for generating peak counts or gene scores from the raw data). We sort the celltyping tasks by the total number of cells in reference and target datasets. The results indicate that when the cell number is low, Cellcano, Seurat, and scJoint use about the same runtime. However, when the cell number starts increasing, Seurat and scJoint can be three times slower than Cellcano. Signac is 2 3 slower than Cellcano when predicting cell types for human PBMCs tasks while its running time is comparable to Cellcano in mouse brain celltyping tasks. All other methods are 5 100 times slower than Cellcano. The reason why ACTINN as a one-round prediction is slower than Cellcano is that ACTINN uses all genes for training while Cellcano selects 3000 genes as features. An additional advantage is that Cellcano is a supervised celltyping method, the pre-trained models can be re-used in future predictions, which means the runtime can be further reduced with the first-round pre-trained model as input.

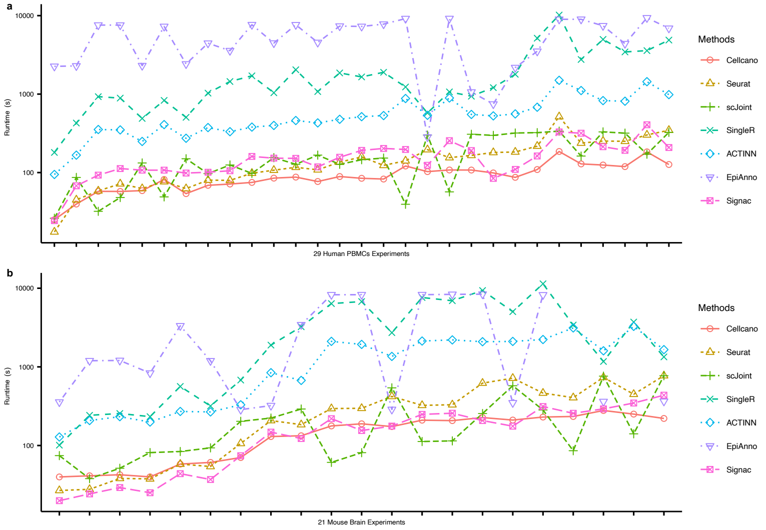


Figure 3.4: a, b Run time comparisons among Cellcano, Seurat, scJoint, Signac, SingleR, ACTINN, and EpiAnno on (a) n = 29 human PBMCs celltyping tasks and (b) n = 21 mouse brain celltyping tasks. The x-axis indicates each celltyping task and is ordered by the total number of cells in the reference and target datasets. Note that EpiAnno fails to generate results for two mouse brain celltyping tasks where the cell numbers are large.

3.4 Discussion

Computational celltyping for single cell omics data is an important problem. Such methods are under-developed for scATAC-seq data. In this work, we develop Cellcano, a two-round supervised scATAC-seq celltyping method. Due to distributional shift, the first-round prediction can be inaccurate, and the anchors can be noisy. The KD model in the second round is thus used to distill the knowledge from a noisily labeled input. We have shown in 50 celltyping tasks with data from two systems (human PBMCs and mouse brain) that Cellcano significantly outperforms other celltyping methods and integration with label transfer methods both in prediction and computational performances. Cellcano is also robust against the anchor selection procedure and batch effects in the data.

Cellcano has several advantages and methodological features. First, Cellcano uses gene scores as input, which has many advantages compared to using bin or peak counts: (1) genes have a much smaller feature space, which significantly improve the computational performance; (2) genes are shared among datasets, which provides potential to be further connected to other modalities, such as gene expression data. We show that using gene scores works as equally well or even better than using bin counts as input. Secondly, Cellcano implements strategies in selecting and using anchors. The MLP in Cellcano can better capture the non-linear relationship between the gene scores and the corresponding cell types. In addition, the KD model is robust to anchors with noisy labels. Moreover, Cellcano does not need to jointly operate on the reference and target datasets, like Seurat, Signac, and scJoint does. This allows Cellcano to be trained on a compendium of reference datasets and provide a pre-trained model.

There are some further developments for Cellcano we plan to work on. First, Cellcano can be adapted to other celltyping scenarios, for example, cross-modality predictions (using scRNA-seq as reference for scATAC-seq celltyping), celltyping in single-cell DNA methylation, etc. Another interesting question is to use multimodal reference data, for example, to jointly use scRNA- and scATAC-seq data as reference to improve celltyping results for either scRNA- and scATAC-seq data. Such an approach can potentially further improve prediction performance.

Chapter 4

Applications of cell-type-specificity

4.1 Integration of single-cell RNA-seq and ATAC-seq data

4.1.1 Introduction

Gene expression or chromatin accessibility by itself provides limited information on biological processes. Multimodal sequencing technologies have been developed to obtain a more comprehensive view of cellular processes by measuring two or more types of information in the same cell [99] [23]. However, these technologies come with increased technical complexity, costs, and data noise, resulting in limited available datasets. To address this issue, computational integration strategies have gained popularity for jointly analyzing multiple single assays to reveal more refined cell-type-specificity and provide potential insights into cell-type-specific regulatory mechanisms [5] [40]. Several methods have been developed for the data integration purpose [63] [101] [98] [122]. Seurat V3 uses canonical correlation analysis (CCA) to capture the joint reduced dimension and then transfer cell labels based on anchor cells selected from the target dataset. The scJoint method involves initial semi-supervised transfer learning to obtain a common embedding space, followed by cell label transfer using K-nearest neighbors (KNN), and then retraining the model to refine the embedding space. scGCN employs a graph convolutional network (GCN) to model the cell-to-cell relationship topology and projects them into a shared embedding space for future label transfer. Portal differs from the previously mentioned methods as it does not include the cell label transfer function. Instead, it mainly focuses on data integration and utilizes adversarial domain translation techniques. However, all these methods require all single-cell modalities as input and cannot directly leverage information from existing high-quality scRNA-seq datasets, limiting the use of existing supervised learning methods. In the meantime, they do not prioritize cell-type-specificity in their integration approach and instead, view cell type information as a byproduct of the integration results.

Our prior work utilized Cellcano to identify cell types in scATAC-seq datasets using another scATAC-seq as a reference and take gene scores as input. As genes is a common feature shared by both scRNA-seq and scATAC-seq datasets, we can leverage the Cellcano method we developed in the previous Chapter to identify cell types in the cross-modality scenario by using scRNA-seq as the reference dataset to predict cell types in scATAC-seq data. In addition, we wonder whether scRNA-seq embeddings can be reused to maximize their utility in the integration of multimodal data, therefore, we propose CellAMA, an adversarial domain translation method that aims to project scATAC-seq data onto the pre-trained scRNA-seq embeddings space while preserving cell-type-specificity, where the cell types are from Cellcano's prediction. Our real data analyses demonstrate the high accuracy of Cellcano in identifying cell types across modalities, as well as the excellent performance of CellAMA in integrating multiple single-cell modalities.

4.1.2 Methods

Adversarial learning.

In recent years, there has been a surge of interest in unsupervised domain adaptation methods, with domain alignment being the most widely used approach [65]. The goal of domain alignment is to minimize differences between the source and target domains, which can be accomplished by reducing the discrepancy either in the original data space or in the feature space. Several methods have been developed to align the feature space, including adversarial learning, statistical divergence alignment, generative domain mapping, and low-density target boundary, among others. After conducting thorough research, we have chosen to use the adversarial learning framework in combination with a statistic called maximum mean discrepancy (MMD) [36] to align a pre-trained scRNA-seq embedding with a projected scATAC-seq embedding. The MMD statistic serves as a statistical metric that enables feature alignment, accurate prediction performance, and a stable training process, all at once.

Pre-trained scRNA-seq embeddings.

Our previous work has shown that combining a multi-layer perceptron (MLP) with F-test results in the best performance for supervised celltyping in scRNA-seq data. We take a further step and investigated the representation layer of MLP (the last hidden layer), and upon performing dimension reduction, we discover that it contains significant information regarding cell-type specificity. However, when using multiple scRNA-seq datasets as input, the representation layer does not eliminate the batch effect and still retains batch-specific information. In this scenario, we use a single high-quality scRNA-seq dataset as a reference and apply CellAMA to map other datasets to it based on their respective cell label information. We subsequently use the mapped embeddings as the pre-trained scRNA-seq embeddings.

CellAMA model.

Since there is a significant domain shift between scRNA-seq and scATAC-seq datasets, using the pre-trained scRNA-seq model directly on scATAC-seq data can lead to sub-optimal performance. To address this issue, we propose CellAMA, which learns a non-linear transformation function to map scATAC-seq data onto a pre-trained scRNA-seq embedding that is shown to have high cell-type-specificity. Overall CellAMA can be divided into two steps:

1. Obtain scRNA-seq pre-trained embeddings and predict cell types in scATAC-seq. We denote the scRNA-seq reference dataset with G genes and N cells as $X_{GE} \in R^{N \times G}$, and the corresponding cell type labels as $C_{GE} \in 1, ..., k$ indicating K cell types. To select highly variable genes, we use the F-test on the known cell groups and choose the top 3,000 genes. Then, we obtain a subset of the gene expression matrix containing the selected genes and perform a standard data normalization procedure. This procedure includes cell-wise normalization (total gene counts sum to 10,000), log plus 1 transformation, and gene-wise standardization (zero-mean and unit-variance). In the first round of Cellcano, we train an MLP on the scRNA-seq reference dataset and extract the last hidden layer as the representation layer Z_{GE}. As part of this procedure, we also obtain a cell type classifier denoted as CLS_{GE}. After selecting confidently predicted cells in the target dataset as anchor cells, we proceed to the second round of Cellcano, which trains another supervised classifier to predict cell types in scATAC-seq data. Here, we summarize scATAC-seq data into gene scores with G genes and M cells, denoted as $X_{CA} \in R^{G \times M}$, the true cell labels as C_{CA} and the predicted cell labels as C'_{CA}.

2. Train an adversarial learning framework to project scATAC-seq data onto the pre-trained scRNA-seq embeddings. The adversarial training consists of an encoder E and a discriminator D. The role of E is to learn the non-linear transformation, while D is responsible for determining whether the data comes from scRNA-seq or

scATAC-seq. We aim to learn a latent space representation Z_{CA} that is well-mixed with the scRNA-seq representation Z_{GE} at the cell-type level. To achieve this, we utilize the label information from original scRNA-seq C_{GE} and predicted scATAC-seq C'_{CA} as guidance along with MMD loss with a Gaussian radial basis function (RBF) kernel (details in the later section).

The first step can be performed separately from the second step, offering the possibility of providing a pre-trained embedding space and classifier from scRNA-seq datasets for projecting other single-cell modalities. In the case of multiple scRNA-seq datasets being used as references, we select one scRNA-seq dataset as the reference and perform step 2 with another scRNA-seq dataset using the ground truth label to perform the projection.

Enhancing cell-type-specificity.

We adopt the approach of CDAN [67] and incorporate the outer product of the embedding space with one-hot encoded cell label information to enhance cell-type-specificity. Additionally, we include an MMD loss for encoder E to enhance feature alignment and stable training. Our modified Minimax loss function becomes:

$$\begin{aligned} &E_{X_{GE}}[log(D(Z_{GE} \otimes C_{GE}))] + E_{X_{CA}}[log(1 - D(Z_{CA} \otimes C'_{CA}))] + \\ &MMD(Z_{GE} \otimes C_{GE}, Z_{CA} \otimes C'_{CA}). \end{aligned} \tag{4.1}$$

Evaluation metrics.

We use Accuracy, macroF1, kohen's kappa, median precision, and median recall to measure the cell type prediction for Cellcano. As for measuring integration performance, we utilize integration metrics including Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), cell-type-level Average Silhouette Width (ASW), and cell-type-level Local Inverse Simpson's Index (LISI) that evaluate biological conservation [70]. As ASW and LISI can be calculated based on different feature spaces, we evaluate these metrics using the embedding space, the top 10 principal components of the embedding space, and t-distributed stochastic neighbor embedding (t-SNE) as inputs.

4.1.3 Results

Overview of CellAMA.

CellAMA aims to project scATAC-seq data onto a pre-trained scRNA-seq embedding with predicted cell types from Cellcano [75], as shown in Figure 4.1a. The scRNA-seq embeddings are obtained from the last hidden layer of the MLP in the first round of Cellcano. We use this pre-trained scRNA-seq embedding along with known cell labels in the scRNA-seq data directly as input for CellAMA. We follow the same procedure to obtain gene-level summaries from scATAC-seq data as we have demonstrated in Cellcano. Our previous work has shown that using gene activity scores can not only achieve superior performance compared to using other feature spaces as input but also facilitate easy integration with scRNA-seq data. We utilize the second round of Cellcano to predict the cell types for scATAC-seq data. Given that we already have the pre-trained scRNA-seq embeddings and their associated cell labels, as well as the predicted cell types for scATAC-seq data obtained from the second round of Cellcano, our objective is to learn scATAC-seq embeddings that exhibit cell-type-specificity and can be projected onto the pre-trained scRNA-seq embeddings. We approach this as a feature alignment task and utilize an adversarial learning framework that contains an encoder and a discriminator. To ensure that the cell-type-specificity is preserved during the projection process, we incorporate the outer product of the corresponding embeddings with the cell types from scRNA-seq or the predicted cell types from scATAC-seq as guidance (as shown in Figure 4.1b) and also add an MMD loss when training the encoder to further enhance feature alignment.

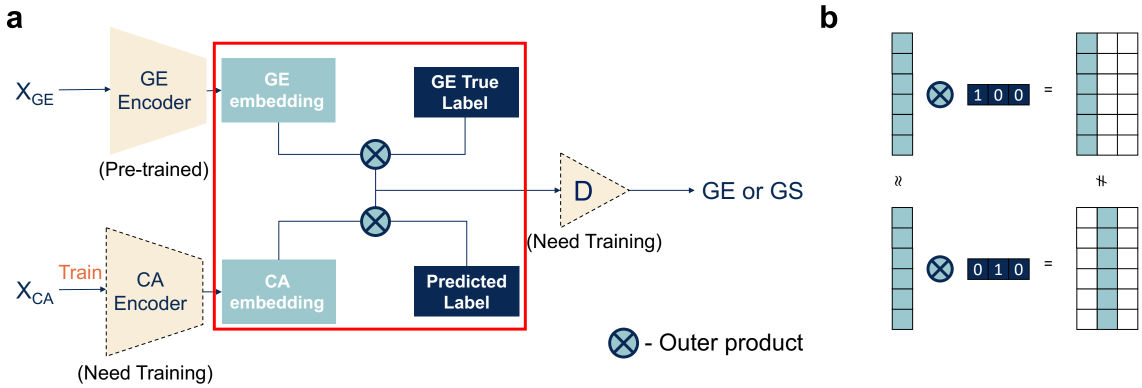


Figure 4.1: (a) CellAMA utilizes adversarial learning to project scATAC-seq data onto a pre-trained scRNA-seq embedding. The use of cell type predictions from Cellcano and outer product enable the projection to be specific to each cell type. The notation X_{GE} refers to scRNA-seq data, which captures gene expression, while X_{CA} refers to scATAC-seq data, which captures chromatin accessibility. The area marked by the dotted line indicates the models that require training procedures. The red lines delineate cell-type-specificity through the use of outer products and we employ a simple cartoon to illustrate this principle in (b).

Cellcano outperforms existing cross-modality integration methods.

To evaluate Cellcano's performance in a cross-modality scenario, we compared it with Seurat V3 and scJoint, both providing celltyping function. Seurat V3 and scJoint take scRNA-seq gene expression and scATAC-seq gene scores as input. Seurat V3 employs canonical correlation analysis (CCA) to identify canonical correlation vectors, which are then used to project both scRNA-seq and scATAC-seq datasets into a common correlated embedding space. Then, following dimension reduction, Seurat identifies the K-nearest neighbors (KNN) for each cell within its paired dataset and uses mutual nearest neighbors (MNN) to identify corresponding anchor cells for the label transfer purpose. scJoint adopts a transfer learning technique to achieve the same goal by first projecting scRNA-seq and scATAC-seq datasets into a joint embedding space and then transferring labels based on the KNN algorithm. In this section, we evaluate celltyping performances from all methods using three metrics including overall accuracy (Acc), macro F1 score (macroF1) and Cohen's kappa (κ).

We design a total of 16 celltyping prediction tasks using two scRNA-seq datasets and four scATAC-seq datasets from human PBMCs and the results are summarized in Figure 4.2. To eliminate the baseline effect caused by the use of different reference or target datasets in each prediction task, we compute the performance gains/losses for each method against the average. Cellcano achieves superior performance compared to scJoint and Seurat in all evaluation metrics. Detailed prediction results can be found in Appendix Figure C.1, where we observe that Cellcano outperforms the other two methods in most of the prediction tasks.

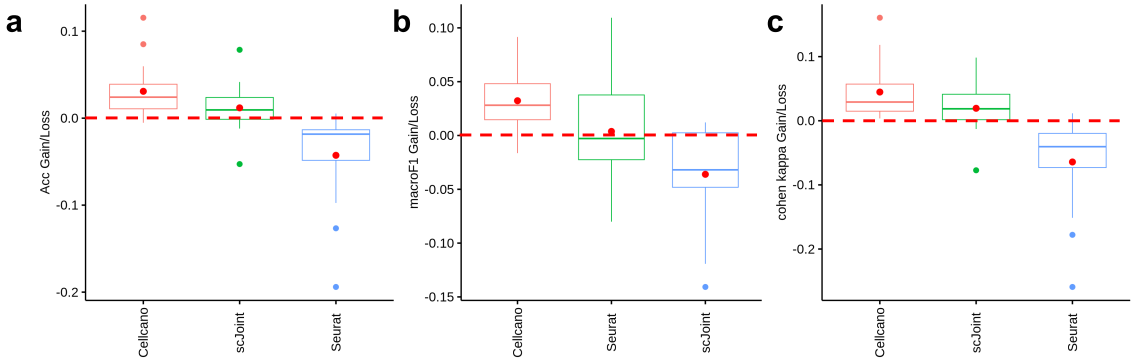


Figure 4.2: (a) Accuracy, (b) macroF1, and (c) Cohen's kappa comparisons between Cellcano, scJoint, and Seurat. Each box contains 16 human PBMCs celltyping tasks. The red dots within each box represent the mean of the performance metric, and the boxes are sorted based on their mean value.

CellAMA provides a better mixture embedding.

Following accurate cell type prediction from Cellcano, we evaluate the performance of CellAMA against Seurat V3 and scJoint, which provide co-embedding latent spaces. We conduct benchmarking using the same 16 celltyping tasks in human PBMCs and evaluate the performance of CellAMA against Seurat V3 and scJoint using NMI, ARI, cell-type-specific LISI (cLISI), and cell-type-specific ASW (cASW) metrics. NMI and ARI use both the ground truth and predicted cell type labels to evaluate clustering agreements. In contrast, cLISI measures the heterogeneity of the data points within each cluster for each cell type, and cASW measures the separation of clusters and how well each data point fits in each cluster for each cell type. While NMI and ARI evaluate the overall performance of assigning data points to clusters, cLISI and cASW focus more on cell-type-specificity.

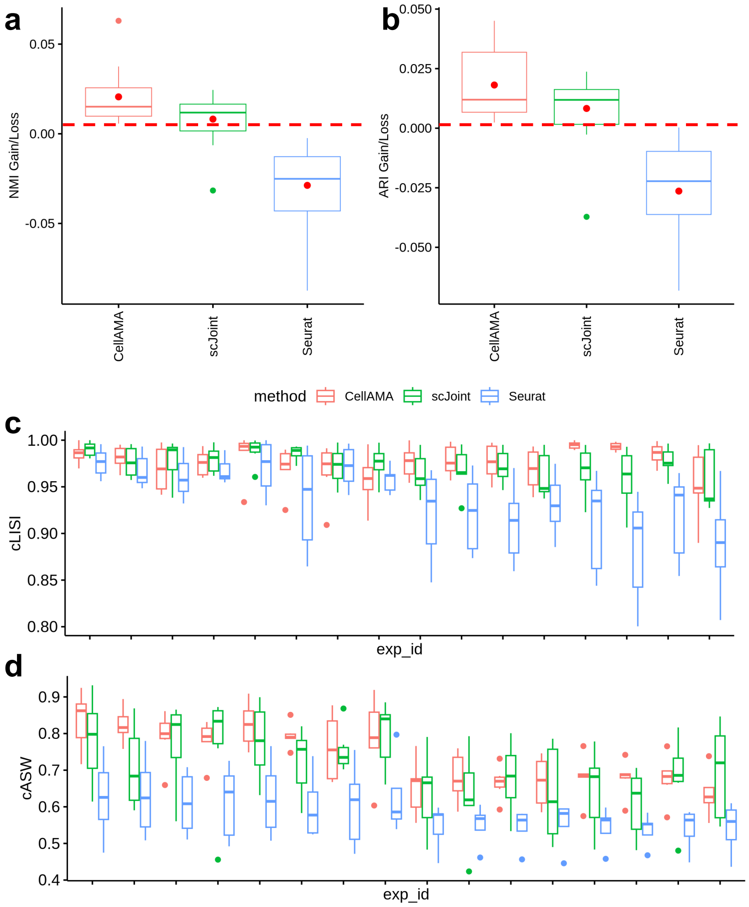


Figure 4.3: (a) NMI, (b) ARI, (c) cLISI, and (d) cASW comparisons between Cellcano, scJoint, and Seurat. cLISI and cASW take latent space as input. In (a) and (b), each box corresponds to 16 human PBMCs celltyping tasks. The red dots within each box indicate the mean of the performance metric, and the boxes are sorted based on their mean value. In (c) and (d), each box contains the number of cell types in each celltyping task, and the metrics are evaluated using the latent space as input.

According to the results, CellAMA exhibits better performance than the other two methods in both NMI and ARI (Figure 4.3a-b). Upon evaluating the cell-type-specificity using cLISI and cASW using latent space as input, CellAMA and scJoint both outperform Seurat V3. Additionally, CellAMA exhibits overall lower variations among cell types compared to scJoint and Seurat V3. When comparing CellAMA and scJoint in detail, we observe that CellAMA outperforms scJoint in 11 out of 16 prediction tasks based on median cLISI and in 10 out of 16 prediction tasks based on median cASW (Figure 4.3c-d). The similar results of cLISI and cASW using top 10 PCs and t-SNE as input are shown in Appendix Figure C.2. Additionally, we provided one prediction example using 10X human PBMCs scRNA-seq data as a reference and FACS-sorted human PBMCs scATAC-seq data as a target to visualize the data integration. The t-SNE visualization in Appendix Figure C.3 displays the integration of scRNA-seq and scATAC-seq data using CellAMA, scJoint, and Seurat V3. In the left panel, the datasets (either scRNA-seq data or scATAC-seq data) are indicated, while the middle panel shows the ground truth cell label information and the right panel shows the predicted cell types. The results show that CellAMA has a better integration in terms of batch integration and cell type integration compared to scJoint and Seurat V3 because CellAMA has a more scattered, generalized, and accurate cell-type-specific projection.

4.1.4 Discussion

The computational integration of single-cell data across different modalities is an important research question. Although several methods have been developed for this purpose, they rely on scRNA-seq data as the reference input for training new target scATAC-seq datasets. This requirement implies that the reference embedding changes every time a new target dataset is introduced, which appears unreasonable. Another current limitation of existing methods is the lack of utilization of cell-type-specific information for integration, which can lead to cell-type misalignment, particularly for small cell populations. Our work aims to tackle both these problems simultaneously. We first perform accurate celltyping with Cellcano and then utilize an adversarial learning framework to perform cell-type-specific projection towards the pre-trained reference embeddings with CellAMA. Based on our preliminary results, Cellcano outperforms two popular methods, scJoint and Seurat V3, in terms of celltyping accuracy. Moreover, the cell-type-specific integration achieved by CellAMA is superior to the other methods based on the accurate celltyping results.

As our work is still in the exploratory stage, there are some limitations that should be acknowledged. First, we have only compared our method to a limited number of existing methods. However, we plan to address this limitation by adding more state-of-the-art batch integration methods such as scGCN [98], Portal [122], GLUE [18] along with other methods into our integration and celltyping comparisons to achieve a more comprehensive evaluation. Second, the combined runtime of Cellcano for celltyping and CellAMA for integration can be slower compared to scJoint. However, one advantage of using supervised methods is that we can train a model on the reference data that can be used later without the need for retraining. Therefore, the procedure can possibly be accelerated by training the reference data with MLP in the first round only. Then with the pre-trained MLP, we can obtain the input for the second round prediction in Cellcano and pre-trained reference embeddings for CellAMA to perform the integration. The third limitation pertains to the interpretation of the trained embedding. In our integration example, we observe that Natural Killer (NK) cells are divided into two clusters. Previous research has shown that NK cells consist of two sub-states, namely CD56 bright NK cells and CD56 dim NK cells. Hence, further analysis is required to evaluate whether our proposed projection can distinguish between these sub-cell types.

Future work involves exploring the scenario of having multiple reference datasets. Existing methods only utilize one reference dataset for integration or celltyping, without investigating the possibility of using multiple references. Our previous research has shown that combining multiple reference datasets can lead to better celltyping performance [73], and similar analysis in the cross-modality scenario has also shown promising results. However, when it comes to integration, the harmonization of embeddings from different datasets is essential for the effective projection of the target dataset. While our previous research has shown that combining reference datasets can improve celltyping performance, the same approach may not necessarily work for integration. Currently, our solution is to use multiple reference datasets for Cellcano to perform celltyping to achieve more accurate results. Then, we use one of the reference datasets to obtain the pre-trained embeddings and project other reference datasets onto it with CellAMA to remove batch effects. We are currently working on this approach and plan to provide more details and results in our future paper.

4.2 LRcell: detecting the source of differential expression at the sub-cell-type level from bulk RNA-seq data

4.2.1 Introduction

Finding differentially expressed genes (DEGs) between experimental conditions is a powerful approach to understanding the molecular basis of phenotypic variation. However, most tissues consist of tens or even hundreds of diverse sub-cell types and DEGs may only occur in a small subset of these sub-cell types, which are relevant to the experimental condition. Bulk RNA-seq data alone are unable to reveal the sub-cell types that drive the DEGs.

The rapid development and proliferation of single-cell technologies resulted in massive accumulation of single-cell transcriptomics data (scRNA-seq) from diverse tissue types. These data reveal substantial variations in transcriptional regulation among different cell types and offer an unprecedented close-up view of the modifications underlying important biological processes, especially for disease pathology, including which cell types drive DEGs [84]. As an example, in a recent single-cell resolution analysis of Alzheimer's disease (AD), Mathys et al. [78] identified glial–neuronal interactions in response to AD pathology. In another single-cell study, Ruzicka et al. [93] found that neurons are the most affected cell type for schizophrenia. However, steep costs and complicated protocols prevent the widespread adoption of scRNA-seq.

Over the past 10 years, many computational cell-type deconvolution methods have been developed to infer the proportions of different sub-cell types from bulk transcriptomic data [84] [83] [33] [114] [111] [32] [60] [124]. Benchmark studies have also been conducted to compare their performance [46] [6].

In this chapter, we propose a novel computational tool named LRcell. Given the result from a bulk RNA-seq differential expression (DE) study, the goal of LRcell is to delineate which sub-cell type(s) of the tissue underwent substantial changes between the two experimental conditions. LRcell is developed under the assumption that expression change occurred at one or few sub-cell type(s) between the two experimental conditions and is the major contributor to the DEGs observed at the bulk tissue level. Cell-type deconvolution methods are not designed to infer such changes. Exploiting cell-type-specific marker genes identified from generic scRNA-seq available from publicly available data repositories, LRcell achieves the goal by surveying the enrichment of marker genes across all sub-cell types in the tissue (Figure 4.4). Thus, no scRNA-seq experiment matching the bulk RNA-seq experimental condition is needed. When applying LRcell to a diverse panel of bulk RNA-seq DE experiments, we successfully identify known sub-cell types involved in the pathogenesis of psychiatric disorders as well as produce testable new hypotheses that have the potential to produce fresh new biological insights.

4.2.2 Methods

Basic assumptions

The goal of LRcell is to identify the most affected sub-cell type(s) during the transition of experimental conditions using only bulk transcriptomic data. Based on the assumptions that cell-type-specific marker genes of key cell types tend to be over-represented among the significant DEGs in bulk transcriptomic studies, LRcell can discover which cell type(s) is involved in certain disease or condition change. In recent years, computational methods have been developed to deconvolve bulk RNA-seq data to delineate cell-type proportion changes, which could be borrowed to answer the same question. However, whenever there are more sub-cell types, the results from deconvolution methods become unreliable. In contrast, LRcell enables comparison across many more cell types which is important for complex tissues such as brain.

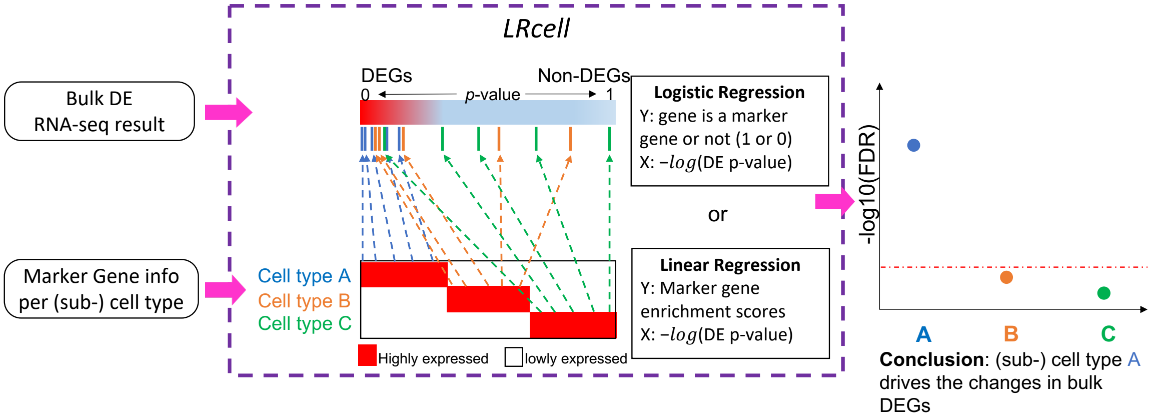


Figure 4.4: Overview of LRcell workflow. As input, LRcell takes in the result from a case-control bulk RNA-seq experiment conducted on specific tissue. For illustration purpose, assuming there are three sub-cell types within the tissue, and the marker genes derived from (unrelated) scRNA-seq experiment on the three sub-cell types are available and taken into account by LRcell. Here, we use the blue color to indicate cell type A, the yellow color to indicate cell type B and the green color to indicate cell type C. We map the marker genes to the entire gene list sorted by DE P-values from the most significant DE to non-DE. Next, for each tissue type, we apply a regression analysis. When using the binary indicator of marker gene as the response variable, we run a logistic regression (LR); when using the enrichment score of the marker gene produced by the Marques et al.'s method as the response variable, we run a linear regression (LiR). In both cases, the explanatory variable is the $-log$ transformed DE P-value. Next, the significance of the regression analysis is calculated and converted to $-log$ transformed FDR and plotted. In this illustrating example, LRcell result indicates cell type A is the most significant, which suggests that cell type A is likely to play a significant role in the case-control experiment.

The development of LRcell is inspired by LRpath [94], which is designed for linking experimental changes to biological pathways or a predefined gene set. In LRcell, we treat cell-type-specific marker genes as gene sets and calculate the enrichment of each cell type when comparing two biological conditions. We believe that the most enriched sub-cell type(s) is highly likely to play an important role in the experimental condition change.

Marker gene selection

After obtaining the log-normalized gene expression matrix along with high-quality sub-cell-type clusters, we calculate the enrichment scores for each sub-cell type using the marker gene selection method described in Marques et al. [77]. The cluster-specific gene enrichment is defined as the average gene expression levels of cells in that cluster divided by the average gene expression levels in all cells. The enrichment score is adjusted by introducing a penalty representing the fraction of cells in that cluster expressing the marker gene. Combined, this score allows the identification of genes with cluster-specific high expression values to be selected as marker genes. The description below is adapted from the original publication.

Suppose there are a total of M genes, L different clusters each with N_j cells and the total number of cells are N. Let $E = E_{ijk}$ represent the gene by cell read count matrix. Here $i = 1, ..., M, j = 1, ..., L, k = 1, ..., N_j$ and $N = \sum_{j=1}^{L} N_j$. The overall average expression of the ith gene across all cells is defined as

$$\overline{E_{i..}} = \frac{1}{N} \sum_{j=1}^{L} \sum_{k=1}^{N_j} E_{ijk}.$$

the ith gene across all cells is defined as

$$\overline{E_{i..}} = \frac{1}{N} \sum_{j=1}^{L} \sum_{k=1}^{N_j} E_{ijk}.$$

The average expression of gene i in the jth cluster is defined as

$$\overline{E_{ij.}} = \frac{1}{N_j} \sum_{k=1}^{N_j} E_{ijk}.$$

The enrichment for gene i in the jth cluster as

$$Enrichment_{i,j} = \frac{\overline{E_{ij.}}}{\overline{E_{i}.}}.$$

Next, we consider the proportion expressing the gene i in the jth cluster as

$$Prop_{i,j} = \frac{1}{N_j} \sum_{k=1}^{N_j} I(E_{ijk} > 0).$$

The $I(\bullet)$ is an indicator function.

The enrichment score for gene i in the jth cluster is computed as

$$Score_{i,j} = Enrichment_{i,j} \times (Prop_{i,j})^{power},$$

where "power" is a hyper-parameter to be tuned manually to control the penalization for the cell cluster proportion term. The power parameter is set to 1 throughout this study. After calculating the weighed gene enrichment scores in each cluster, we ranked genes based on the scores and selected the top 100 genes as the marker genes for each cluster.

LRcell analysis

LRcell is inspired by LRpath, which was designed for identifying sets of predefined gene sets that show enrichment with differentially expressed transcripts in microarray experiments. LRcell uses logistic regression (LR) or linear regression to assess whether marker genes (as defined in the Marker Gene Selection subsection above) of a specific cell type are more likely to be DEGs in a particular bulk RNA-seq study. The linear regression option is added to handle the continuous enrichment scores of marker genes. Users can choose accordingly. To facilitate our analysis, we assume that the major sub-cell types of the tissue their marker genes are known *a priori*.

We apply LRcell to each cell type independently. The required input includes a list of DEGs ranked by the level of significance and a set of marker genes for each cell type. Then, LRcell runs a LR as

$$log\frac{\theta}{1-\theta} = \alpha + \beta x$$

and

$$\theta = P(Y = 1).$$

In which $Y = 1$ denotes that gene is a marker gene and $Y = 0$ otherwise. Hence, θ represents the chance that the gene is a marker gene. We use $-log(P - value)$ as the explanatory variable x. Whether a specific cell type is involved in the experimental condition change is evaluated by testing the null hypothesis that $\beta = 0$ against the alternative that $\beta \neq 0$ using the Wald test. Typically, we run LRcell on all sub-cell types found in the tissue to see which sub-cell type(s) drives the changes.

Similar to LR, linear regression directly performs

$$Y = \alpha + \beta x,$$

where Y indicates the enrichment scores of genes. Same as LR, the P-value can be obtained from testing the null hypothesis that $\beta = 0$ against the alternative that $\beta \neq 0$ using the t-test. Once the P-values are obtained, we calculate false discover rate (FDR) using P adjust() function in R to adjust P-values with Benjamini–Hochberg method.

Input and output

LRcell requires two inputs: (i) a ranked list of genes with DE P-values in a bulk RNA-seq experiment and (ii) sets of marker genes from all sub-cell types of the bulk tissue acquired from scRNA-seq datasets *a priori* or from MSigDB C8-cell-type signature gene sets. For those cell markers derived from scRNA-seq datasets, we offer choices for users to choose between species as human or mouse and the region indicates the specific brain region or PBMCs. For MSigDB cell markers, we store the marker genes into the LRcellTypeMarkers packages which can be easily downloaded. When running LRcell() function, the LR option is set as the default, while users can also set the method option as LiR if linear regression is desired. For linear regression, an enrichment score is needed as input for each gene whereas gene sets are sufficient for LR. For MSigDB cell-type signature gene set, LR option is recommended as there is no enrichment score information available. For customized input, i.e. a scRNA-seq data, we offer a LRcell_gene_enriched_scores function which takes the read counts matrix and cell annotation as input to generate enrichment scores for genes in each cell type. For further subsetting, get_markergenes can be used for generating marker genes for more specific sub-cell types. More details on data preprocessing can be found in Appendix C Section 1-3.

The output is a list of significance P-value (or FDR), one for each sub-cell type. For visualization, LRcell produces Manhattan plot, which can be drawn through plot_manhattan_enrich function. We also provided a plot (plot_marker_dist function) indicating where certain cell-type-specific marker genes locate on the bulk DEGs. The bulk DEGs are sorted using $log10(P-value) \times sign(log2FoldChange)$ which could potentially give information on both up/downregulated directions. LRcell requires an R version beyond 4.1 and a prerequisite installation of BiocManager.

Simulation strategy

Simulated scRNA-seq data are generated using scDesign2 [105], which is capable of generating synthetic scRNA-seq data using intrinsic statistical parameters learned from real scRNA-seq datasets. We generate three synthetic scRNA-seq dataset as control samples using parameters learned from the adult mouse FC scRNA-seq data. For the three case samples, we use the same statistical model but either alter the expression level of selected genes or the proportion of one sub-cell type. We then sum up corresponding read counts to obtain the bulk RNA-seq data and use them to detect DEGs between case and control samples. For LRcell analysis, we use the marker genes computed from the original scRNA-seq dataset as input. For MuSiC analysis, we use the three control replicates as the reference scRNA-seq dataset.

For implementing the scenario where only DEGs occur, we first generate three control replicates and randomly select 1000, 2000 and 3000 out of 29,653 genes. We then either double or halve the gene expressions of those genes in the specific sub-cell type being tested. To add certain noises, we use a normal distribution with standard deviation as 0.1 to generate random fold change which fluctuates around 2 or 0.5.

As for the scenario where only proportion changes, we directly use the parameter named cell_type_prop from the function simulate_count_scDesign2() to change the simulated proportions. We decide two different proportion distributions when there are five sub-cell types: one is evenly distributed with all sub-cell types having 20% proportion and the other one is unevenly distributed with 40, 30, 10, 10 and 10%. When testing for the robustness of LRcell under more sub-cell types, we only use even distribution on cell-type proportions for illustration purpose.

More detailed information is available in the online publication [74].

4.2.3 Results

In this work, we collect and curate a compendium of marker genes from multiple published scRNA-seq datasets. We then conduct LRcell analysis on multiple bulk RNA-seq DE experiments to demonstrate its utility.

Marker gene collection and sources

Genes that show substantial expression difference between one sub-cell type and others in their native state are regarded as marker genes [49]. Similar to a collection of gene set for Gene Set Enrichment Analysis (GSEA) [104], LRcell requires a compendium of high-quality cell-type marker genes. Currently, LRcell package provides users with multiple preloaded marker gene sets from human blood, human brain and mouse brain (Figure 4.5A), computed from scRNA-seq datasets using method introduced in Marques et al.'s [77] study. Additionally, LRcell package offers external cell markers collected by Molecular Signatures Database (MSigDB) [61] with certain criteria (Appendix C Section 4). The external makers all originate from human species including midbrain, cord blood, ovary and skeletal muscle. We store all cell-type-specific marker gene sets into another R Bioconductor ExperimentHub package named LRcellTypeMarkers. Additional marker gene sets are being tested and will be added to the collection.

Simulation settings

Because the ground truth of changes in DEGs and cell-type proportion is difficult to monitor and track, we conduct simulation studies to demonstrate the effectiveness of LRcell.

In this simulation study, we consider experiments between cases and controls involving DEGs and proportion changes. We simulate both single-cell and bulk RNA-seq data. Both types of data are generated by scDesign2 [105] using the adult mouse frontal cortex (FC) scRNA-seq dataset [96] as a reference and we use the marker genes previously derived from the dataset to conduct our LRcell analysis. More details can be found in the Methods section.

For simplicity, we consider two scenarios in our simulation study: (1) the proportions for all sub-cell types remain the same during the condition change and DEGs

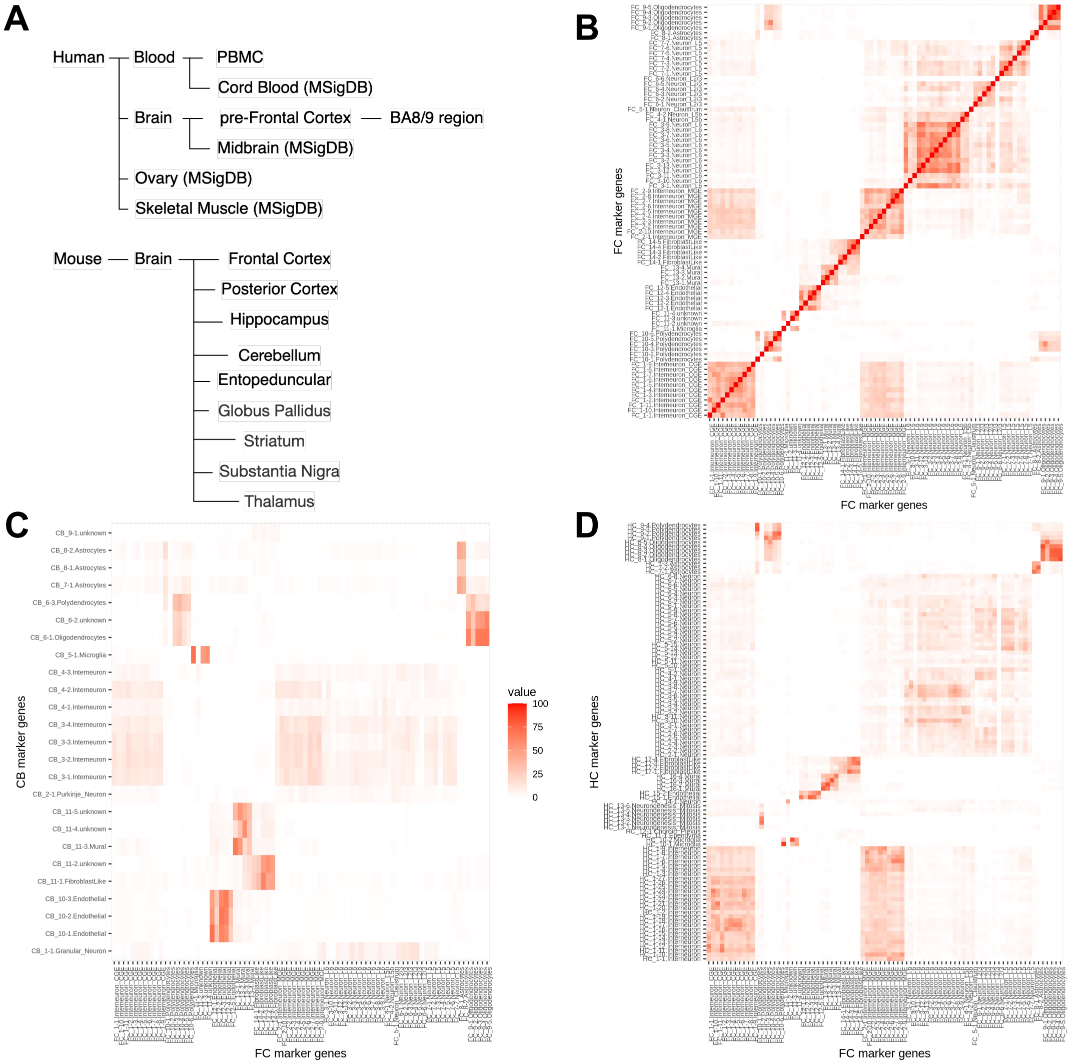


Figure 4.5: LRcell datasets and marker genes overlap between different brain regions. (A) summary of the all tissue-types in which marker genes have been pre-embedded in LRcell. In (B–D), top 100 marker genes are selected for each cell type, and thus, the maximum overlap in these figures is 100. (B) Heatmap illustrating the overlap of marker genes among cell types within the FC region derived from mouse whole brain scRNA-seq dataset. The highlighted area describes the overlap between FC_11-3.unknown, FC_11-4.unknwon and FC_11-1.Microglia as an illustration for the similarity between these three sub-cell types. (C) Heatmap illustrating the overlap of marker genes among cell types within the FC and cell types within the cerebellum CB. (D) Heatmap illustrating the overlap of marker genes among cell types within the FC and cell types within the hippocampus.

are found in one specific cell type; (2) sub-cell type proportions are different between case and control and no DEG is found in any sub-cell type. Under each scenario, we try to simulate different combinations.

Under the first scenario, we consider the following settings: (a) cell-type proportion distribution (evenly or unevenly distributed); (b) the total number of cells (1000; 5000 or 10 000 cells); (c) the number of DEGs occurred in that specific sub-cell type (1000; 2000 or 3000 DEGs out of 29,653 in the whole genome); and (d) fold change direction of DEGs (2 or 0.5 times of the original gene expression).

Under the second scenario, we consider the following combinations: (a) cell-type proportion distribution (evenly or unevenly distributed); (b) the total number of cells (1000; 5000 or 10 000 cells); and (c) proportion change in that specific sub-cell type (50; 80; 120 or 150% of the original proportion).

Additionally, to push the boundary of LRcell performance when there are many more sub-cell types, we simulate cases where there are 5, 10 and 15 sub-cell types and altering the baseline proportions which are evenly distributed in various ways.

Simulation results

For the simulation study, we take turns to alter each individual sub-cell type, then run LRcell or MuSiC [114] and track the rank of the altered sub-cell type as an indicator of the performance.

Because under the first scenario, there is no proportion change hence we do not test the performance of MuSiC. The ranking results are summarized in Appendix Figure C.4A and B, and LRcell is able to correctly identify most of the sub-cell type changes. The cases in which incorrect identification made are those with the smallest number of DEGs (in other word, where 1000 DEGs are simulated).

For the second scenario, we compare LRcell, MuSiC and GSEA (using marker genes as gene set). The results are summarized in Appendix Figure C.4C–E. We observe that MuSiC performs steadily well under all settings while LRcell produces a few errors. This is fully expected since the scenario matches the assumption of

MuSiC but not LRcell because it is not a cell-type proportion deconvolution method.

We also compare LRcell, MuSiC and GSEA under the scenario when there are more sub-cell types. The results are summarized in Appendix Figure C.4F–K. We notice that when there are 10 sub-cell types, LRcell and MuSiC work equally well and when there are 15 sub-cell types, LRcell performs slightly better than MuSiC when adding up the ranks. In particular, for the setting of 1000 cells with 20% increase of proportion, both LRcell and MuSiC detect an incorrect but similar sub-cell type. A specific showcase has been presented in Appendix Figure C.5, to show an overall performance regarding all sub-cell types. Under all settings, LRcell and MuSiC outperform GSEA.

Microglia highly enriched in neurodegenerative dementia

After the simulation study, we conduct LRcell in real data analysis. In a recent neurodegenerative dementia study, Swarup and colleagues contrasted TPR50 mice expressing tau mutant with wild type mice using bulk RNA-seq in order to identify gene networks mediating dementia [107] (‘the mouse AD study’ afterward). To identify the cell type(s) most involved in the condition, we apply LRcell to the DEG list using pre-embedded marker genes from adult mouse FC region [96]. From LRcell result, we observe that Microglia show up as highly significant (Figure 4.6A) which is concordant with previous studies [88]. Additionally, the FC_11-3.unknown and FC_11–4.unknown sub-cell types also show high level of significance. No annotation is available for these two cell clusters in the original publication. However, pairwise comparison of marker genes among all cell clusters reveal that these two unknown cell clusters have considerable overlaps with the FC_11-1, which is also a Microglia cell type (Figure 4.5B), which explains the pattern we observe.

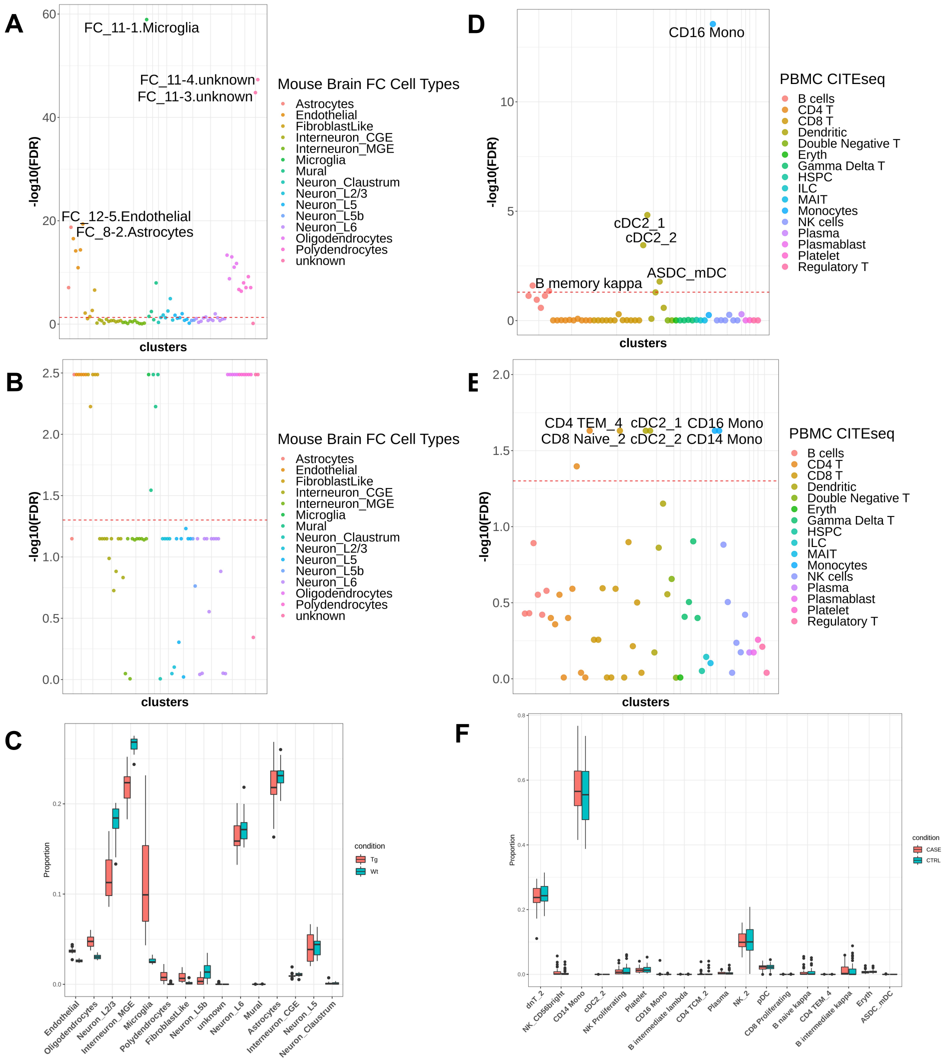


Figure 4.6: Applying LRcell to real cases. (A) LRcell result of mapping the bulk neurodegenerative dementia DEGs to the mouse brain FC region. (B) GSEA result of mapping the bulk neurodegenerative dementia DEGs using the same marker genes used in LRcell (mouse brain FC) as input. (C) Cell-type proportions for control and disease samples calculated by MuSiC. Each box contains 17 individuals. The x-axis is ordered by the t-test significance between the two conditions. (D) LRcell result of mapping bulk PTSD DEGs to human PBMC. CD16+ monocytes is shown as the most significant cell type. (E) GSEA result of mapping bulk PTSD DEGs to human PBMC using the same marker genes used in LRcell (human PBMC) as input. (F) Cell-type proportions for control and disease samples calculated by MuSiC. The x-axis is ordered by the t-test significance between two conditions.

CD16+ monocytes highly enriched in posttraumatic stress disorder

In a recent study, Breen and colleagues conducted a bulk whole-transcriptome study using peripheral blood leukocytes collected from U.S. Marines, among which some developed posttraumatic stress disorder (PTSD) postdeployment [13] ('the human PTSD study' afterward). Using this dataset, we generate a list of DEGs that show significant difference between the PTSD group and the control group at the predeployment time point.

Using human marker genes derived from a single-cell transcriptomic study on peripheral blood mononuclear cells (PBMCs) [39], LRcell analysis finds that cells annotated as CD16+ nonclassical monocytes shows up as the most significant among all cell types in PBMC (Figure 4.6D). Our finding makes biological sense because as stated in previous studies [85], heterogeneity exists in monocytes distinguished by CD16 surface proteins and nonclassical monocytes have been validated to regulate immune responses in trauma [54] [55].

Specificity, robustness and running time of LRcell

It is of interest to evaluate whether LRcell shows good specificity, i.e. low false positive rate. To do this, we simulated null scenario where there is no significant DEG in any of the sub-cell type. When apply LRcell to such null bulk RNA-seq data, we found that LRcell produce either no or much fewer and weaker significant result, illustrating good sensitivity of LRcell.

To analyze the robustness of LRcell analysis, we run experiments from two perspectives: (i) whether the number of marker genes strongly affects LRcell results and (ii) whether a different DEGs detection method affects LRcell results.

We first conduct LRcell using different marker gene number derived from PBMC scRNA-seq dataset on the human PTSD study and we get similar enrichment performances (Appendix Figure C.6). This indicates the robustness of the LRcell analysis.

In addition to DESeq2, we use Voom [57] with Limma [90] to perform DEGs analysis on the mouse AD study and the human PTSD study. Details of the usage can be found in the Appendix C Section 5. With the same marker genes set, we notice that the enrichment patterns are similar as FC_11-1. Microglia is highly enriched along with other sub-cell types (Appendix Figure C.7).

In addition, we analyze the execution time among LRcell, GSEA and MuSiC under different simulation scenarios (Appendix Figure C.8). We observe that LRcell and GSEA are steadily fast, while the execution time of MuSiC increases when the number of reference cells increases. LRcell takes about 3–4 s on average for each run on a typical laptop computer.

4.2.4 Discussion

Detecting transcriptional activity changes at the individual cell type level, especially their modifications in disease samples, is crucial for understanding the mechanisms of diseases development. In this study, we propose a novel strategy named LRcell which conducts enrichment analysis of cell-type-specific marker genes among the top (or bottom) DEGs identified by bulk transcriptome studies. Cell types that show the most enrichment are likely to play an important role in the condition alteration. When applying to real datasets, we found that LRcell can successfully identify the involvement of the Microglia and Astrocytes in the mouse AD study and rare monocytes in the human PTSD study.

Many computational methods have been developed to infer the proportions of different sub-cell types from bulk transcriptomic data. LRcell is not designed for estimating cell-type proportions. We assume that different proportion of sub-cell types in cases and control samples is not the major source of the DEGs observed at the bulk tissue level. Rather, expression changes occur at one or few sub-cell type(s) between case and control samples is the major contributor to the DEGs observed at the bulk tissue level. Recent studies showed support for our assumption. For example, Segerstolpe et al. [97] showed no significant shift of cell-type proportions in pancreatic islet between type 2 diabetes patient samples and control samples, but the amount of

DEGs vary substantially across sub-cell types. Based on this assumption, we designed LRcell to identify which sub-cell types may be involved in the experimental condition change and thus follow up experiments can be designed to explore the mechanisms of the involvement of the specific sub-cell type(s) in the experimental condition.

Although based on different assumptions, out of curiosity and also in order to put LRcell results in context, we apply MuSiC, a well-established deconvolution method to the mouse AD study data. Because some layers of neurons are predicted to have almost zero proportion (Figure 4.6C) when using all 81 sub-cell types, we merge the original sub-clusters into 15 major cell types in order to achieve a better representation. Despite this, MuSiC does not detect significant difference in Microglia or Astrocyte in terms of their proportions between the two conditions. When applied to the human PTSD study data, using the original cell cluster annotations, MuSiC shows that most of the T sub-cell types have zero proportion and the proportion of CD14+ monocytes is up to 60% (Figure 4.6F). In contrast, LRcell produces more sensible results because it is not limited by the number of cell types as it can detect the subtle differences among sub-cell types.

Interestingly, from our simulation studies, LRcell is also capable of detecting sub-cell types that undergo proportion changes, albeit with slightly lower accuracy comparing to state-of-the-art deconvolution methods.

A key advantage of LRcell lies in its ability to handle a large number of sub-cell types. This is because LRcell analyzes sub-cell types one-by-one, whereas deconvolution methods have to do the analysis jointly which leads to higher computation burden and poorer performance.

In spirit, LRcell operates similarly as GSEA, but LRcell is much more sensitive to minor differences in marker genes of sub-cell types, similar to the advantage of LRpath showed when comparing to GSEA. This indicates LRcell's potential to detect changes in sub-cell types caused by disease conditions. Simulation studies comparing

LRcell with GSEA suggest very similar patterns as observed from real data analysis. Additionally, when compared to existing bulk deconvolution methods, LRcell is more stable in its ability to handle the similarities among sub-cell types. Thus, LRcell enables researchers to glean new biological insights from the bulk transcriptomics experiments with no need of redoing the experiment using single-cell technology. We are currently applying LRcell to a diverse set of clinical studies (Sharma, personal communication) to generate more biological insights.

How to select marker genes representing sub-cell types is an important research question. Plenty of methods have been developed to optimize the selection process [77] [89] [89]. However, due to the dramatic diversity among sub-cell types and tissues, there is no consensus universal criteria on the selection criteria that can make the marker gene set representative and complete, which is also dependent on the goal of the study including cell clustering, cell-type calling and cell-type deconvolution, among others. For LRcell, our experience leads us to adopt the method introduced in Marques et al. for its simplicity and computation efficiency. We have performed empirical studies to illustrate the effectiveness of the marker genes selected by the adopted method. Alternatively, precompiled marker gene sets from emerging databases [121] cover more and more tissue types which are great resources.

To enable straightforward comparison, currently, we select a fix number of 100 marker genes from each sub-cell type. Understandably, the number of marker genes for different cell types varies; it is desirable to allow flexibility in choosing the number of marker genes based on the transcriptomic patterns across cell types. However, different numbers of marker genes post challenge for conducting enrichment analyses fairly across all cell types. This will be investigated in our future studies.

LRcell currently provides embedded marker genes from human blood, human brain and mouse brain calculated from scRNA-seq experiments along with markers from 66 cell types in four tissues (midbrain, cord blood, ovary and skeletal muscle) adopted from MSigDB. We are working to include more tissue types in the future releases of LRcell which will make it more widely applicable.

In summary, we develop LRcell, an R Bioconductor package for identifying sub-cell type(s) that drive the changes observed in bulk comparative transcriptomic studies, taking advantages of newly emerged scRNA-seq data. The rationale of LRcell is that we believe marker genes of the modifying cell types tend to be enriched toward the top (or bottom) of the DEG lists. We conduct comprehensive surveys applying LRcell across various experimental conditions and successfully identify cell types that play important roles in the mouse AD study and the human PTSD study. Hence, we believe that LRcell can provide researchers important and new biological insights in terms of the source of the biological changes at the sub-cell-type level, without the need of conducting costly and laborious scRNA-seq experiments.

Our findings from both simulated data as well as real data suggest that LRcell is complementary to cell-type deconvolution methods. Therefore, we recommend including LRcell to bulk RNA-seq analysis to gain a holistic understanding of changes occur at the sub-cell-type level inside complex tissues.

Chapter 5

Discussion

5.1 Conclusions

The rapid evolution of single-cell sequencing techniques has revealed cell-type-specificity from various perspectives and provided a wealth of high-quality single-cell genomics data. This, in turn, has stimulated the development of novel statistical and deep learning techniques to better understand biological mechanisms at a cellular level. This dissertation focuses on evaluating and developing supervised celltyping methods in Chapters 2 and 3. Chapter 2 provides a comprehensive overview of important factors involved in supervised celltyping methods for scRNA-seq data. Building upon this, Chapter 3 introduces a new method called Cellcano, which is a two-round supervised celltyping approach specifically designed for scATAC-seq data. Having successfully performed celltyping, we obtain cell-type-specific information. In Chapter 4, we investigate the feasibility of using predicted cell types as guidance to integrate single-cell genomics data. Moreover, we introduce a method called LRcell that leverages cell-type-specific markers to detect cellular activity in bulk experiments.

The primary innovation of this dissertation lies in its approach to uncovering cell-type-specificity and utilizing it to explore potential applications. Additionally, we provide valuable insights and recommendations based on our experiences. We investigate the feasibility of utilizing the simplest neural network model - MLP, as the supervised classifier and combining it with different popular feature selection strategies in this field. In contrast to more complex models, this simple network does not require a graphics processing unit (GPU) but yields comparable results and incurs similar training or prediction time. Using the cell labels of reference datasets, we employ an F-test statistical test to select variable genes among cell type groups. The combination of MLP and F-test yields improved performance, and we utilize this combination when developing our method, Cellcano. Cellcano accounts for the stronger batch effect that exists between reference and target data by employing a two-round strategy that leverages partial data from the target dataset to build a knowledge distiller model. The KD model trained on the target dataset has a better understanding of the target data distribution and therefore yields superior performance compared to other existing methods. The KD model is a simple model that minimizes system requirements and provides computational efficiency by utilizing shadow layers and only a few nodes. Furthermore, Cellcano provides clear and detailed documentation, which is user-friendly for biomedical researchers. Accurate identification of cell types enables the possibility of conducting downstream analyses with cell-type-specificity information. We provide two examples of such analyses: integration of single-cell genomics data and interpretation of signals in bulk experiments. To achieve a cell-type-specific projection, we recommend performing accurate celltyping prior to integration. Additionally, we investigate the feasibility of using cell-type-specific marker genes as indicators to identify cell types that are specifically enriched in bulk differential expression studies.

Despite the promising prediction results, several challenges still exist in supervised celltyping. The first challenge is that supervised methods usually heavily rely on labels from reference datasets. However, each dataset may contain varying levels of granularity in cell labeling, leading to inconsistencies among different datasets. As a result, current practices are based on major cell types that are either shared among datasets or aggregated from multiple sub-cell types. In Chapter 2, We tested the prediction of sub-cell types in mouse brain datasets that contained different sub-cell types in various neuronal layers. The performance was lower, at around 70% to 80%, as opposed to the 97% to 99% achieved when predicting major brain cell types. Therefore, our proposed solution would be to first classify major cell types and then in each classified cell type, we utilize unsupervised clustering with known markers to identify sub-cell types. This approach is expected to yield more accurate results compared to purely unsupervised methods. Another challenge associated with having labeled reference datasets is that certain cell types present in the target dataset may not exist in the reference dataset. While we proposed a solution in Chapter 2 to capture uncertain cell types using predicted probabilities from the prediction model, it does not provide information on the potential identity of these cell types. One potential solution could be to utilize the Human Cell Atlas or Mouse Cell Atlas as a reference since they encompass all cell types from various tissues. However, same as the first challenge, these references do not contain sub-cell types, and distinguishing the same cell types across different tissues can be difficult. The third challenge pertains to identifying cell dynamics, which involves continuous changes in cell states. Categorizing cell states into specific labels is challenging due to the dynamic nature of the process. Utilizing supervised celltyping methods can provide a general idea of which cell states a cell may belong to. However, it is anticipated that the prediction results may be noisy. Therefore, traditional unsupervised methods utilizing graph-based or tree-based trajectory generation algorithms are more appropriate for this scenario.

Moreover, there exist certain limitations that present opportunities for further improvement and research. In Cellcano, we currently set the entropy threshold at 0.4 to select confident cells. However, the optimal threshold may vary for different cell types based on the signal-to-noise ratio. Certain cell types may be more readily identifiable due to their distinct data characteristics, while others may be challenging to be distinguished due to similar profiles, for example, CD4+ T cells and CD8+ T cells in human PBMCs. To address this issue, an appropriate cutoff for the predicted probability distribution can be established for each cell type. This approach can enhance the confidence of the anchors while capturing the target data distribution. Similarly, in LRcell, we currently use an arbitrary cutoff, which includes 100 marker genes for each cell type as the marker gene set. However, cell-type-specific marker genes may vary across different cell types, necessitating the selection of a more rigorous cutoff. Additionally, LRcell only offers marker genes for a limited number of tissues. Currently, we are in the process of collecting additional scRNA-seq datasets to derive more marker genes for a broader range of tissues.

Based on our exploration of single-cell data integration, it is evident that emphasizing cell-type-specificity can improve the integration of cell types with a small population. However, accurately integrating cell types with similar profiles remains challenging and can lead to misalignments. One additional question is about the application and interpretation of the joint embeddings learned from the models. In single-cell genomics, the primary research focus is often on extracting cell-type-specific marker genes or conducting differential expression analyses to identify gene expression changes between different conditions. Although joint embeddings learned from different single-cell modalities have shown promising results in data integration, it is challenging to directly map the embedding space back to the gene expression space. Therefore, it is still unclear whether the embeddings derived from single-cell genomics data are meaningful, especially in the absence of ground truth or human-interpretable information. Further investigations are required to explore the generation and utilization of embeddings produced by these models.

A final question that remains is regarding the interpretability of the model. As famously stated by the statistician George Box, "All models are wrong, but some are useful." Deep learning models have been shown to achieve better performance and can capture non-linear transformations between input data and prediction results. However, the lack of transparency in their functioning makes them difficult to interpret as they behave like black boxes. The field of interpretable deep learning aims to address this issue by developing methods that make the prediction process more transparent. We anticipate that the development of interpretable deep learning methods will provide greater interpretability in the single-cell genomics field and offer valuable insights to address fundamental biological questions.

These unresolved questions and challenges present exciting prospects for future research. In this study, we have established a solid foundation for precise celltyping and proposed potential applications of cell-type specificity. Our aim in this dissertation is to provide valuable insights and contribute to the field of single-cell genomics research. We hope that our work will inspire and guide other researchers in this area and facilitate a deeper understanding of biological mechanisms and disease treatments at the cellular level.

5.2 Future research plan

I have several projects in mind for my future research. One of my plans is to expand my experience in deep learning and supervised celltyping to other single-cell modalities such as single-cell DNA methylation (scDNAm), single-cell three-dimensional genomic interaction measurements (scHi-C), and spatial genomics. This will allow me to accurately identify cell types for these different single-cell genomics data and gain a deeper understanding of cellular characteristics. By diving deeper into the intricacies of cell-type-specificity, I aim to explore the potential of leveraging it for achieving more accurate integrative analysis in single-cell genomics. This could involve developing novel methods and algorithms. For instance, while our current supervised integration method CellAMA is effective in projecting other modalities to the reference embedding, it is limited by summarizing all modalities into a shared common space. By leveraging single-cell multi-omics techniques, we could potentially explore the use of mosaic integration to achieve even more accurate celltyping and better integration performances.

Another research question that I am interested in is understanding the sequencing data generated by single-cell techniques. As we explored the best input for scATAC-seq celltyping, we observed the presence of discordant peaks called from different scATAC-seq datasets. This makes me wonder if the discordance in peak calling results is influenced by specific characteristics of single-cell data, such as the presence of excessive zeros and lower counts, or whether it is indicative of true dataset-specific information. Building on this idea, I am also interested in exploring the potential application of stable diffusion to perform quality control and data imputation in spatial transcriptomics data. I choose the diffusion model over other generative models because it can preserve the original dimensionality of the data, rather than projecting it into a lower-dimensional embedding space, which makes the downstream analysis convenient to be performed. By examining the results of downstream analysis at each step, I can determine whether the current dataset requires data imputation or has good quality.

In addition to method development for deciphering cell and genomics characteristics, I also plan to address biological questions in my future research. One of the most intriguing biological questions I aim to investigate is the cell-type-specific gene regulatory mechanism using the publicly available single-cell genomics data. While biological experiments have been conducted to analyze one or several specific genes and link them to biological processes or disease progression, exploring the entire gene regulatory network of specific cell types could reveal new insights into the underlying molecular mechanisms of cell function and disease. Currently, our approach using Cellcano for celltyping in the cross-modality scenario yields reasonable results, suggesting a potential cell-type-specific correlation between gene expression and summarized gene scores. However, we acknowledge that our summarization approach fails to capture distal regions from the gene body that may contain crucial regulatory information. Therefore, my plan is to use the pseudo-labels generated by Cellcano as guidance, and then model the bin counts from scATAC-seq and gene expressions from scRNA-seq to reveal the connections between gene expressions and regulatory elements beyond the gene body region. Another research perspective that I am interested in is leveraging the co-expression from scRNA-seq data, co-accessibility from scATAC-seq data, and co-localization information from spatial transcriptomics data to predict other interaction information such as co-methylation and three-dimensional genomic interactions. As co-interaction data is structured as a network or graph and contains multiple modalities, I plan to use multi-view learning in conjunction with graph neural network techniques to achieve my goal.

Appendix A

Appendix for Chapter 2

A.1 Performance Grain / Loss Calculation

For each metric in each experiment, we have 9 classifiers and 6 feature selection methods resulting in a 9x6 two-way table. We first calculate the mean of each two-way table as the experiment's baseline performance. Next, we subtract the mean from the two-way table for each experiment and aggregate all experiments together by taking the average. This results in one 9x6 data matrix for each metric. Finally, we sort the table in a descending way by rows and columns, in which the top left corner combination has the most gain while bottom right combination has the most loss. We present the gains/losses in a heatmap for each metric.

When merely focusing on the gains/losses from feature selection strategies, we subtract the mean of each column (representing each feature selection method) to average out effects from classifiers. Then, we combine all results together to draw the boxplot. Similar procedure has been done for evaluating the gains/losses from different classifiers.

A.2 Data Pre-processing

All scRNA-seq datasets have been pre-processed by filtering out low-quality cells expressing in less than 10 genes and genes expressing in less than 10 cells. We then use normalize each cell to have 10,000 reads and do log-transformation. Next, we scale the dataset to zero mean and unit variance and truncate absolute values with maximum of 6. Finally, the data is fed to corresponding classifiers.

A.2.1 Analysis details for comparing condition effects

For comparing condition effects, we include 7 mice from "Mouse brain FC" and 6 mice from "Mouse brain HC" using Drop-seq, 6 mice with saline treatment from "Mouse brain pFC" using 10X Chromium, and 2 cortex samples named cortex1 and cortex2 from "Mouse brain cortex" using DroNc-seq. DroNc-seq and Drop-seq are proved to have similar performance. To summarize the differences for the two datasets effect, the "Mouse brain pFC" contains protocol difference and the "Mouse brain cortex" includes certain region difference as it profiles the whole cortex region.

We then curate each dataset to contain only major cell types, such as integrating multi-layers of neurons in "Mouse brain FC" together as neurons, removing pericytes from "Mouse brain cortex", etc. During cell type curation, we find there exists both newly formed oligodendrocytes (NF Oligo) and oligodendrocytes (Oligo) in "Mouse brain pFC". We first categorize NF Oligo into oligodendrocytes, but we find NF Oligo is annotated as polydendrocytes in "Mouse brain FC" when visualizing subjects from "Mouse brain FC" and "Mouse brain pFC" together (Appendix Figure A.7). Therefore, we decide to categorize NF Oligo as polydendrocytes. In summary, there are 7 major cell types: neuron, interneuron, astrocytes, oligodendrocyte, polydendrocyte, endothelial and microglia. We use P60FCCx3cr1Rep1 from "Mouse brain FC" as the target individual to validate the condition effect.

A.2.2 Analysis details for comparing pooling effect

Under intra-dataset setting, we conduct three experiments (1) 8 lupus patients in batch1 from “Human PBMC lupus”, (2) 7 mice from “Mouse brain FC” using 14 major cell types, and (3) the same subjects using 81 sub-cell types. We fix one individual (ID: 1085) in (1) and one mouse subject (P60FCCx3cr1Rep1) in (2)(3) as target and then perform the “pooling” strategy. For “pooling”, we combine all other individuals or subjects together to predict the fixed one. Then, we down-sample the combined reference to the average number of the dataset (total number of cells divided by number of individuals or subjects) for 30 times. As for inter-dataset, we use 7 mice in “Mouse brain FC” to predict 6 mice with saline treatment in “Mouse brain pFC”. Curation procedure has been done first. Then, we use each mouse in “Mouse brain pFC” as target to perform individual effect and then the “pooling” strategy. When using each mouse as target, the down-sampling is performed 10 times.

A.2.3 Datasets for pooling saturation analysis

When analyzing the performance saturation of using larger reference data, we conduct three experiments using mouse brain datasets because they have more individuals and cells compared to other datasets. For predicting major and sub-cell types within the dataset, we use 6 individuals in “Mouse brain FC” to predict the rest individual (P60FCCx3cr1Rep1). For across datasets prediction, we use all 6 mice from “Mouse brain pFC” and 3 mice from “Mouse brain Allen” to predict major cell types in one mouse from “Mouse brain FC” (P60FCCx3cr1Rep1) to mimic the real scenario.

A.2.4 Analysis details for purifications

We conduct four experiments for testing cell purifications. The first three experiments come from “Human PBMC lupus” and the last experiment is conducted on “Mouse brain FC". We (1) use one lupus patient (ID: 1154) to predict another patient (ID: 1085) in batch1; (2) use 8 lupus samples from batch1 to predict 8 lupus samples from batch2; (3) use 8 lupus samples from batch2 to predict 8 IFN-β treated samples from the same batch; (4) use one mouse subject (ID: P60FCRep1) to predict another subject (ID: P60FCCx3cr1Rep1) from the same brain region on sub-cell types. For distance-based purification, we first compute each cluster's centroid by averaging the processed read count matrix (scale and log-normalized) of cells belonging to this cluster. Next, we compute the Euclidean distance between each cell and the centroid and remove 10% cells with the largest distance. For probability-based purification, we first fit an SVM with RBF kernel model on the reference dataset and generate a probability matrix denoting how possible a cell belongs to a cluster. Then, for each cluster, we remove 10% cells with the lowest probability. After purifying the cluster, we predict again on the same target dataset.

A.3 Analyses details on pooling saturation

For creating reference data by combining individuals, we first randomly shuffle the orders of the individuals, and then sequentially add them to the reference dataset. For each reference, we perform F-test to select the top 1000 features selection and predict cell types in the target dataset. To remove the potential variations brought by the order of the individual, we repeat the above procedure for 50 times and average the results. For creating reference data by subsampling from all cells, we first pool all cells from all individuals together and randomly shuffle the order. We add 3,000 cells each time to create a reference dataset. Again, we repeat this process 50 times in order to reduce the variations in sampling.

A.4 Number of features has an impact on performance

We also inspect how the number of features might affect the prediction. We pick two experiments as illustrations. One experiment is using 8 samples from “Human PBMC lupus” batch2 under control status as a reference to predict 8 IFN-β stimulated samples from the same batch. The other experiment is using one mouse in “Mouse brain FC” to predict another mouse “Mouse brain FC”. We set the feature number from 100 to 5,000 with 100 as the step size. For the first experiment, the performance reaches a peak of around 500-600 features and decreases when the feature number increases (Appendix Figure A.8A). For the second experiment, the performance first increases and plateaus after 500 features (Appendix Figure A.9A). The pattern can be fully explained by the tSNE dimension reduction plot (Appendix Figure A.8B, A.9B) for both experiments. When the feature number increases, clusters first become tighter and then gradually over-clustered. For major cell types, over-clustering will not affect prediction, but for similar subtypes, it introduces biases. However, feature selection itself is a very interesting research topic in the single-cell research area. In our study, we choose 1,000 as the number of features for further analysis because most experiments perform well.

Table A.1: Mouse brain datasets used in evaluation study

	Dataset Description	Protocol	No. cells	No. major cell types (subtypes)
Mouse brain FC	GSE116470, Frontal cortex brain region, 7 male adult mice subjects	Drop-seq	71,639	14 (81)
Mouse brain HC	GSE116470, Hippocampus cortex brain region, 6 male adult mice subjects	Drop-seq	53,204	12 (103)
Mouse brain pFC	GSE124952, 6 saline-treated adult mice (2 in each 3 timepoints: control, 48h after cocaine withdrawal (CW), 15 days after CW)	10X Chromium	11,886	8 (9)
Mouse brain cortex	SCP425, cortex1 and cortex2 samples from one-month old mice	DroNc-seq	1,452 (cortex1) 892 (cortex2)	8
Mouse brain Allen	NeMO: dat-jb2f34y, 3 male adult mice with frontal cortex extracted	10X Chromium	65,944	8 (47)

Note: We remove pericytes from Mouse brain cortex dataset. For Mouse brain Allen dataset, we extract out cells within ACA and PL;ILA;ORB brain regions and consider them as frontal cortex.

Table A.2: Human PBMC Datasets used in evaluation study

	Dataset Description	Protocol	No. cells	No. major cell types (subtypes)
Human PBMC lupus	GSE96583, batch1, 8 SLE patients	10X Chromium	12,544	6 (8)
Human PBMC lupus	GSE96583, batch2, 8 SLE patients untreated for 6 hours	10X Chromium	12,138	6 (8)
Human PBMC lupus	GSE96583, batch2, 8 SLE patients activated by IFN-β for 6 hours	10X Chromium	12,167	6 (8)
Human PBMC protocols	SCP424, pooled frozen 25million pbmc1 and within 4-hour fresh blood pbmc2	Smart-seq2/ CEL-Seq2/ 10X Chromium (v2)	6,814 (pbmc1) 223 (pbmc2)	6 (9)
Human PBMC FACS	10X Genomics Datasets, fresh healthy Donor A with 10 bead-enriched subpopulations	FACS	94,655	5 (10)

Note: For Human PBMC 7 protocols dataset, we extract pbmc1 data with Smart-seq2, CEL-Seq2 and 10X Chromium protocols and pbmc2 data with Smart-seq2 data only.

Table A.3: Human Pancreas datasets used in evaluation study

	Dataset Description	Protocol	No. cells	No. major cell types (subtypes)
Human Pancreas	GSE85241, 4 dead donors (1 female, 3 males; variation in Age and BMI), 8 libraries	CEL-Seq2	2,018	6
Human Pancreas	E-MTAB-5061, 6 healthy and 4 T2D individuals (variation in healthy gender and age, BMI)	Smart-Seq2	2,038	6
Human Pancreas	GSE81608, 12 Healthy and 6 T2D donors (balanced gender, varied age, BMI, weight)	SMARTer	1,492	6

Note: We curate the human pancreas datasets only containing the 6 major cell types including alpha, beta, gamma, delta, acinar and ductal cells.

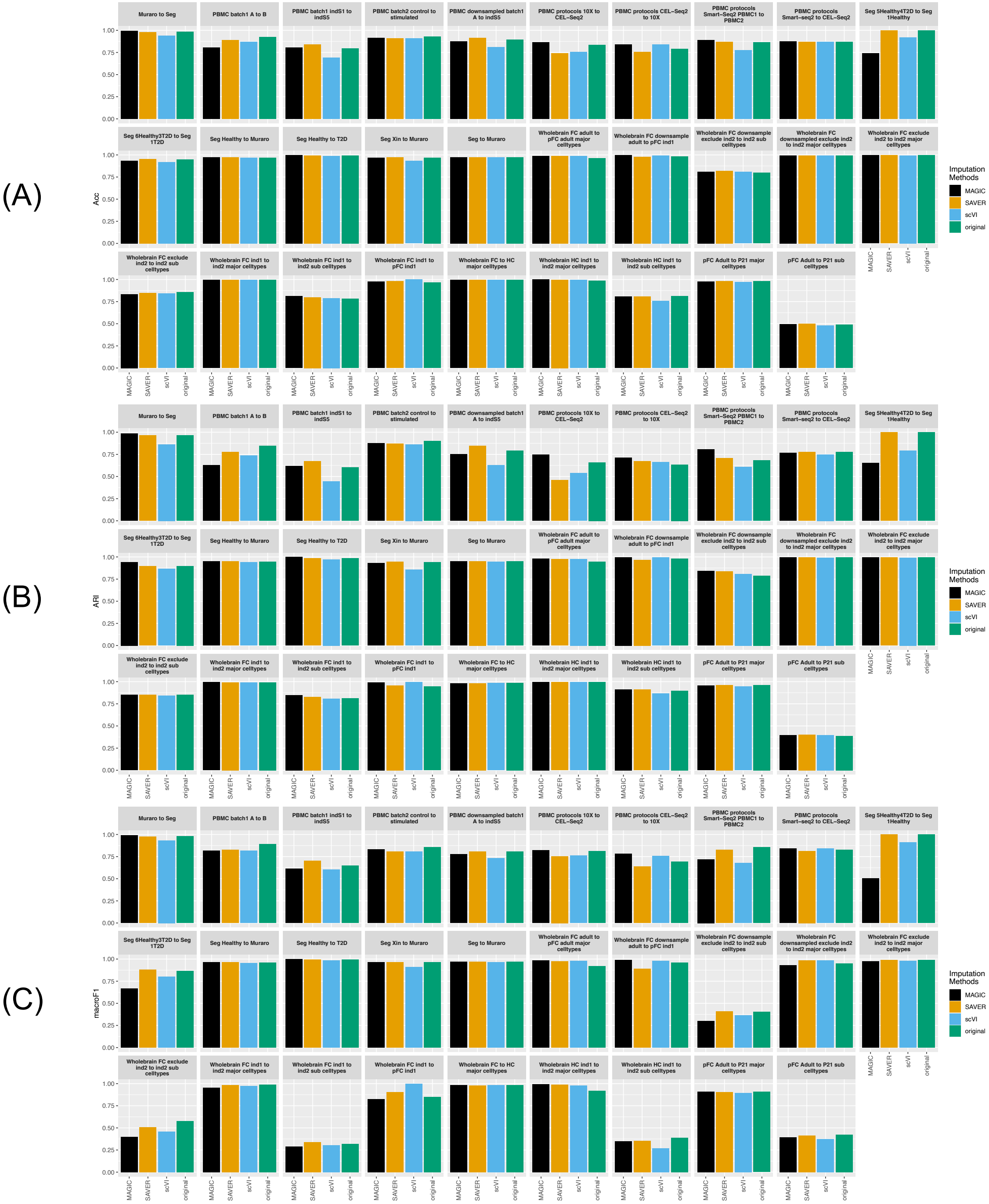


Figure A.1: Prediction performance comparisons before and after imputation. (A) Accuracy; (B) ARI; (C) Macro F1. The imputation methods are performed on both reference and target datasets. The black, orange, blue, and green bars stand for MAGIC, SAVER, scVI and the one without imputation respectively.

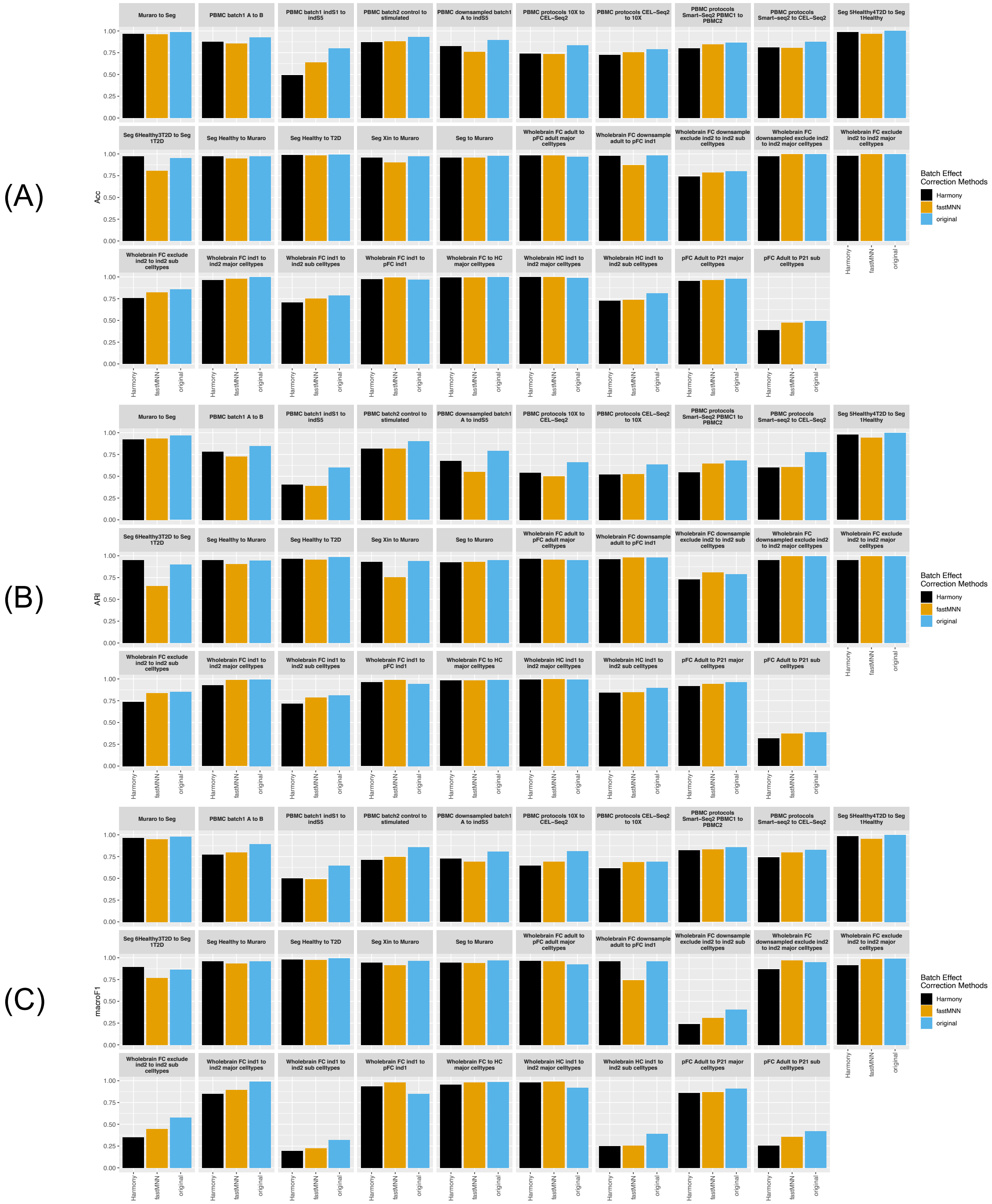


Figure A.2: Prediction performance comparisons before and after batch effect correction. (A) Accuracy; (B) ARI; (C) Macro F1. The batch effect correction is performed between reference and target datasets. The black, orange, and blue bars stand for Harmony, fastMNN and the one without batch effect correction respectively.

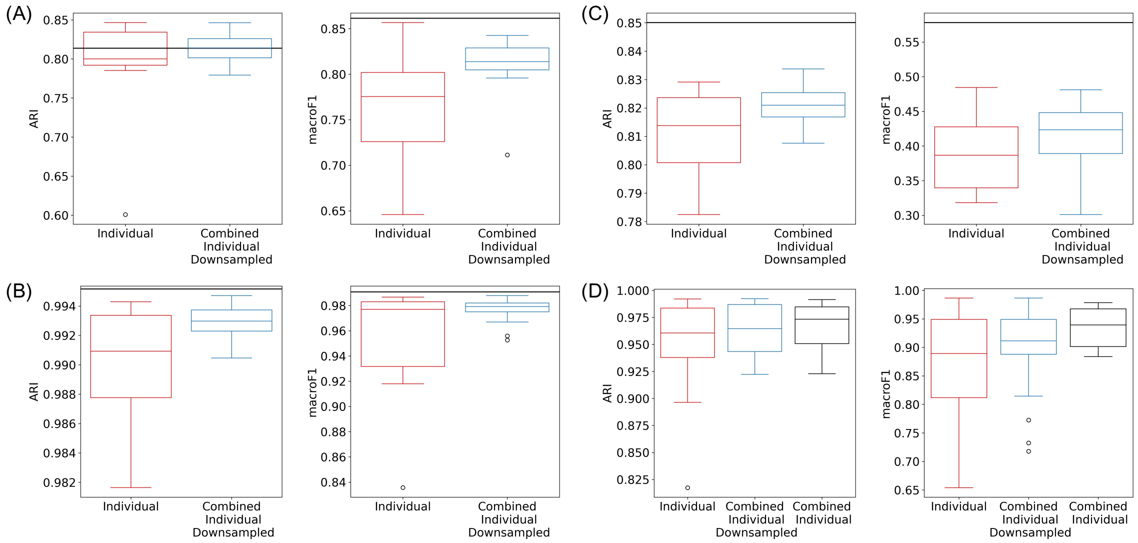


Figure A.3: Impact of "pooling" on individual effect under intra-dataset and inter-dataset scenarios on ARI and Macro F1. (A)(B)(C) are under intra-dataset setting (black line indicates "pooling" all individuals) and (D) is under inter-dataset setting (black box indicates "pooling" all individuals).

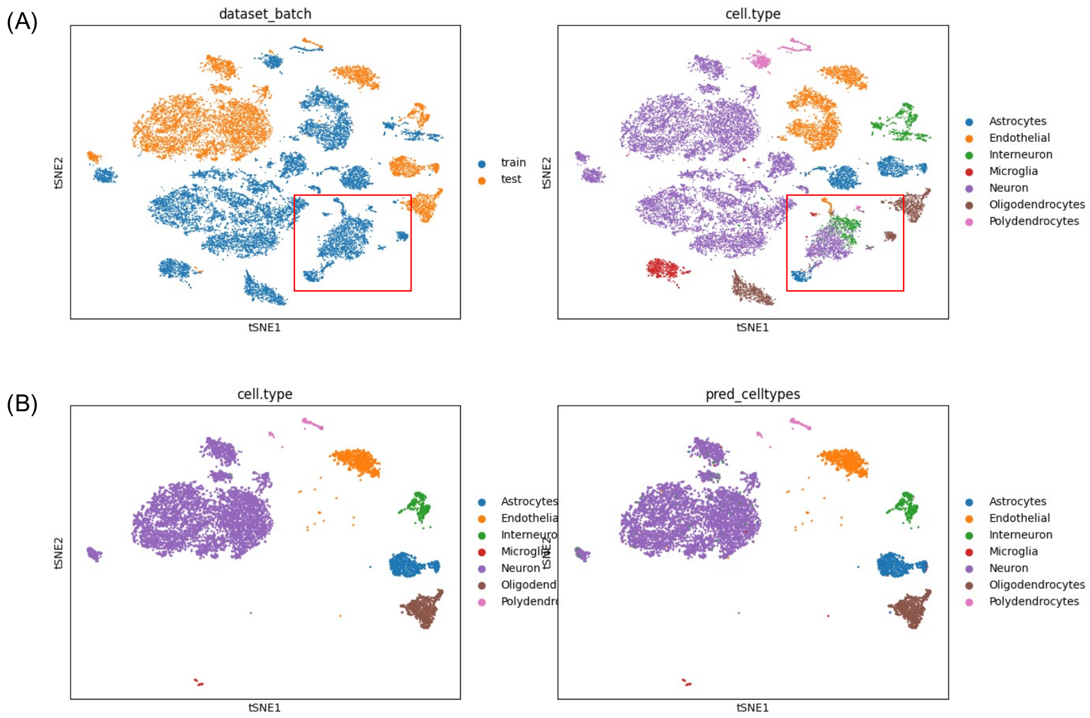


Figure A.4: Cell type annotations when combining “Mouse brain pFC” and “Mouse brain cortex” to predict the target from the “Mouse brain FC” dataset. (A) The blue dots in red box are cells from “Mouse brain cortex” and all other blue dots come from “Mouse brain pFC”. The orange dots are cells from “Mouse brain FC”. As shown in (B), corresponding cells in the red box contain a mixture of several cell types. (C) Ground truth cell types for target dataset. (D) Predicted cell types for target dataset. (D) shows some interneurons (green dots) misclassified as neurons (purple dots).

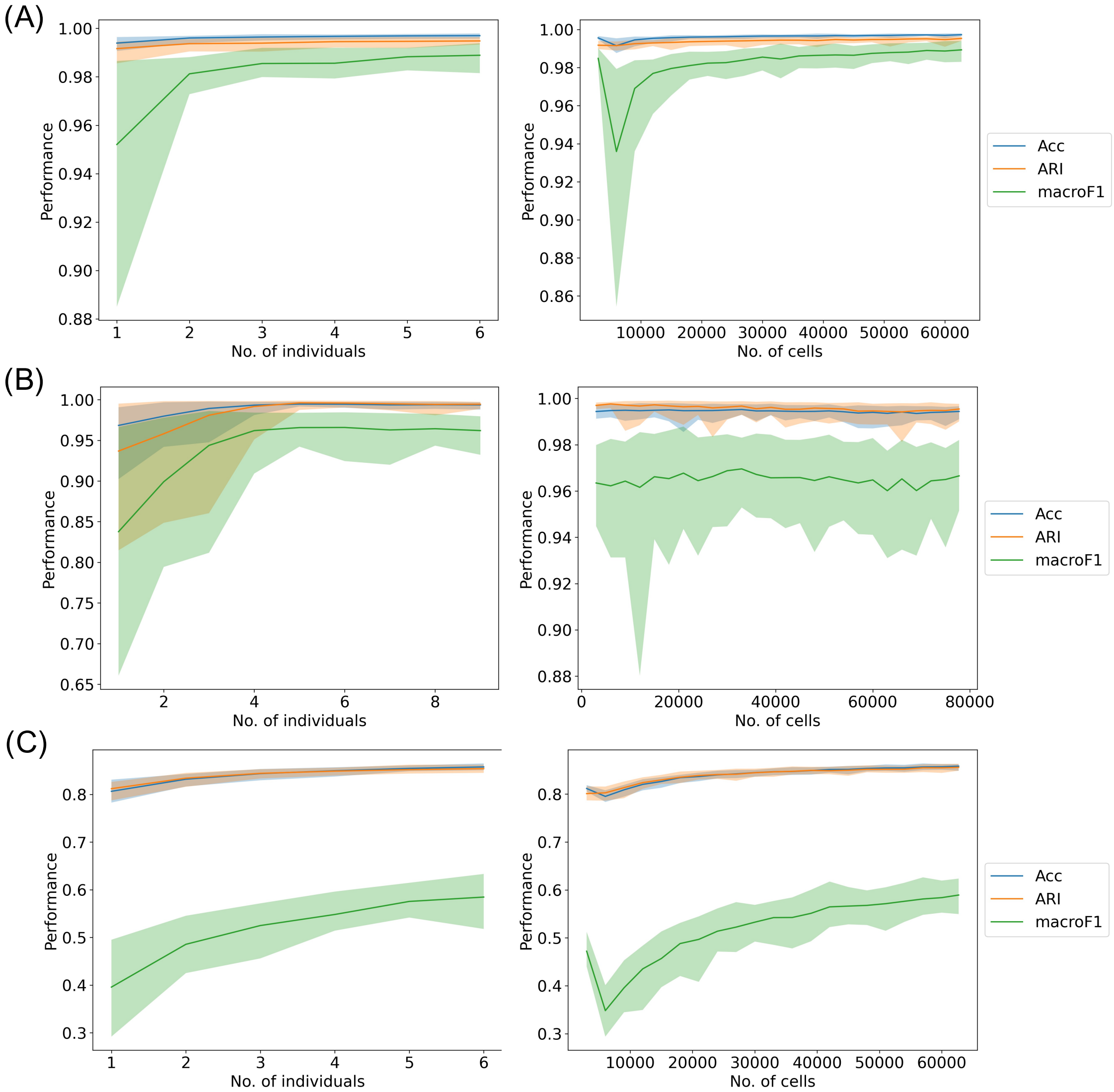


Figure A.5: The blue line, orange line and green line indicate Accuracy, ARI and macroF1 changes respectively. Results are based on 50 random shuffles when adding individuals (left panel) and cells (right panel). The shaded area is the 0.025 quantiles and 0.975 quantiles of the 50 results. (A) "Mouse brain FC" within dataset prediction using major cell types; (B) combines the "Mouse brain pFC" and "Mouse brain Allen" to predict one individual in "Mouse brain FC"; and (C) "Mouse brain FC" sub-cell types prediction. With more individuals being added, the performance increases and saturates in (A) and (B) but increases without saturation in (C).

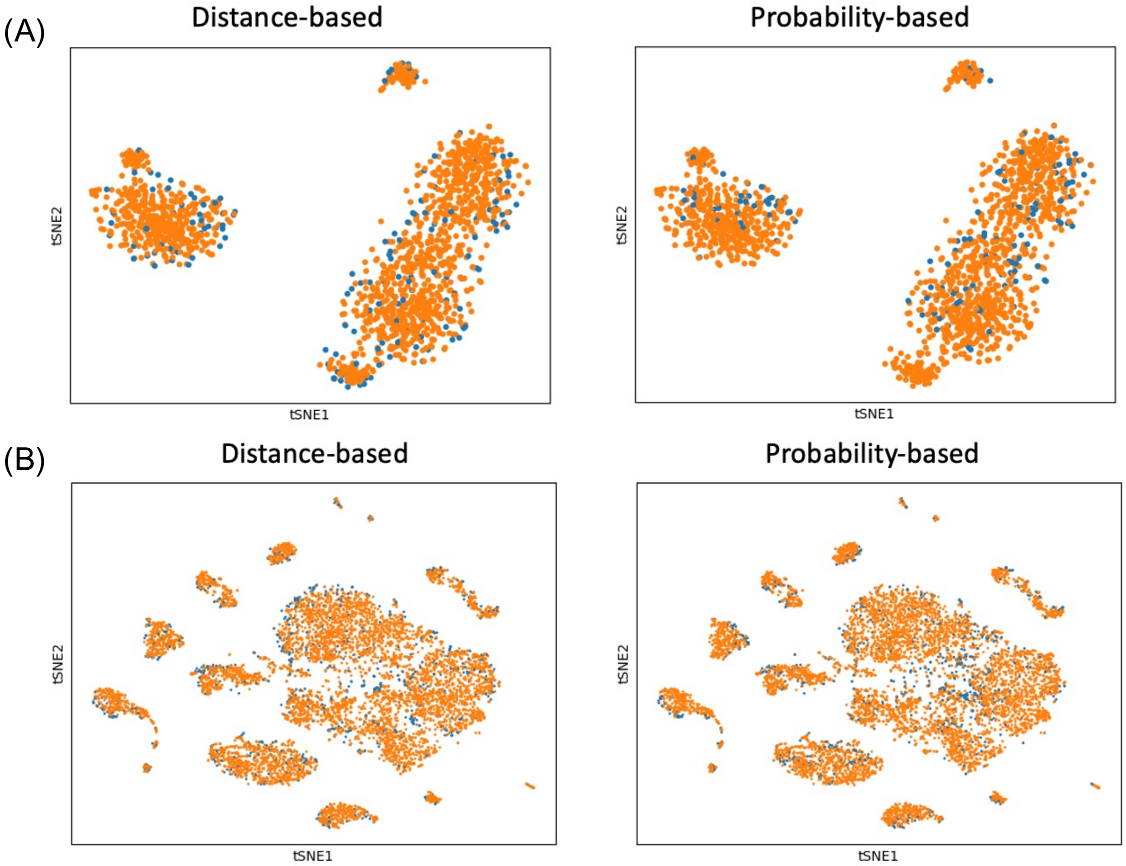


Figure A.6: Orange dots are remained as reference dataset and blue dots are removed by different purification strategies. (A) "Human PBMC lupus": use one lupus sample from batch1 to predict another sample from the same batch under the same condition. (B) "Mouse brain FC": use one mouse subject to predict another mouse from the same dataset under the same condition on sub-cell types. As shown in the right panels, cells on boundaries of clusters are removed in probability-based purification.

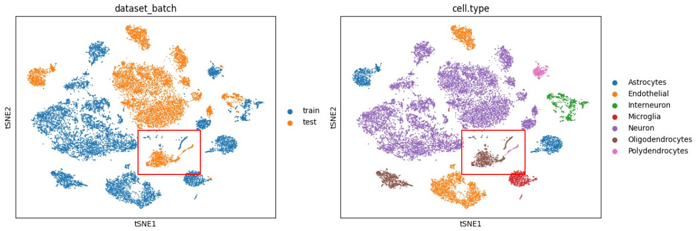


Figure A.7: Different cell type annotations between mouse brain datasets. The red box contains two lineages. The blue lineage is annotated as newly formed oligodendrocytes (NF Oligo) while the orange lineage is annotated as polydendrocytes.

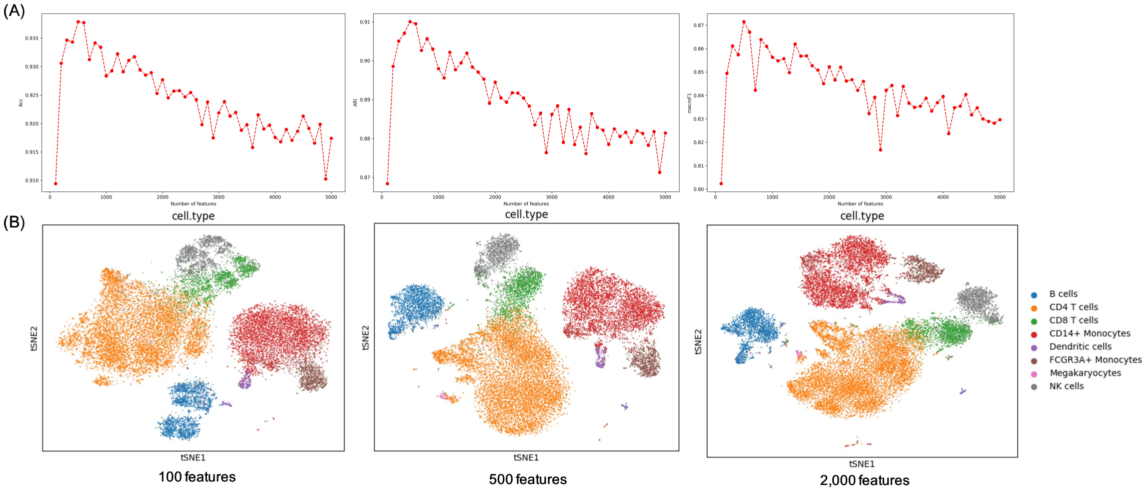


Figure A.8: Impact of feature number when using 8 samples from "Human PBMC lupus" batch2 to predict 8 IFN-β treated samples from the same batch. (A) shows Accuracy, ARI and Macro F1 performance changes when selecting 100 to 5,000 features and (B) shows tSNE visualizations when selecting 100, 500 and 2,000 features.

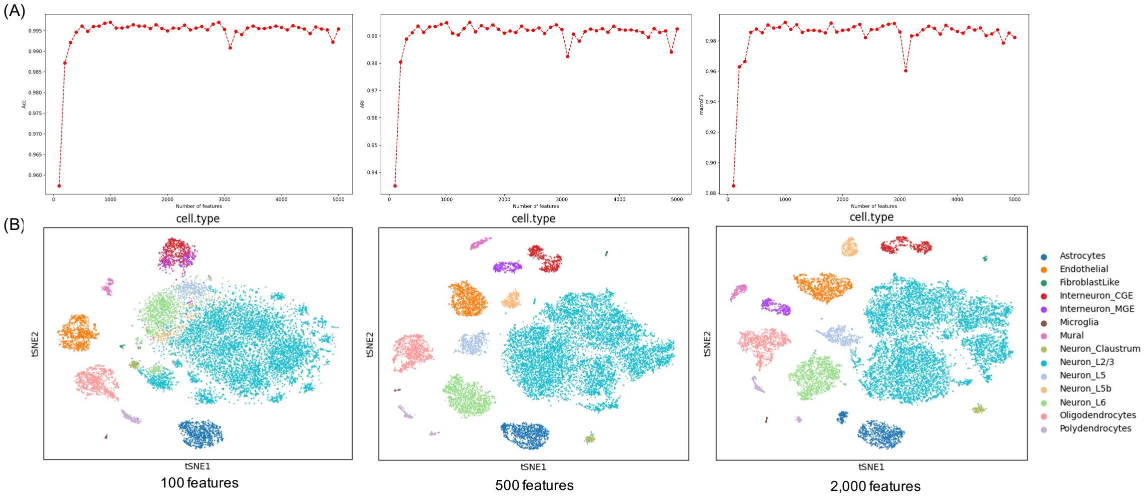


Figure A.9: Impact of feature number when using one mouse subject from "Mouse brain FC" to predict another mouse subject from the same dataset under the same condition. (A) shows Accuracy, ARI and Macro F1 performance changes when selecting 100 to 5,000 features and (B) shows tSNE visualizations when selecting 100, 500 and 2,000 features.

Appendix B

Appendix for Chapter 3

B.1 Data preprocessing by ArchR

All raw scATAC-seq data (fragment or bam files) are processed by ArchR. We set genome hg19 for human PBMCs datasets and mm10 for mouse brain datasets. Then, we load the downloaded fragment files or bam files as input for ArchR to generate the ArrowFiles with $createArrowFiles()$ function. In the function, two parameters serve with quality control purpose: minTSS and minFrags. We adjust the thresholds according to original papers to obtain high-quality cells.

The gene score matrices and genome-wide fixed-size bin counts are generated using the default setting in ArchR. The gene score matrix is generated with ArchR recommended gene score model (as illustrated in the next subsection). The bin counts are generated with 500-bp bins genome-wide. This results in around 6 million bins in hg19 and 5 million bins in mm10. To accelerate the data loading time, we filter out the bins with non-zero counts in less than 1% cells to reduce the feature space. The peak-by-cell matrices generation needs additional peak calling steps in ArchR. To reuse the ArrowFiles generated earlier, we put ArrowFiles from all human PBMCs datasets together and call peaks. ArchR first clusters cells and then creates pseudo-bulk replicates to assure the reproducibility of peak calling. Once the peaks are obtained, reads are counted on the peak regions to generate the peak count matrices. The same procedure has been performed in mouse brain datasets.

B.2 An introduction to different ArchR gene score models

The script to generate gene score models is provided by ArchR (`https://github.com/GreenleafLab/ArchR_2020`). In total, there are eight categories of gene score models including:

- (1) Model – Promoter: This class of models count the reads located on the promoter region with different window sizes.
- (2) Model – GeneBody: This class of models count the reads located on the whole gene body with certain extension in up- or down-stream.
- (3) GeneModel – Constant: This class of models count reads from 1K bps upstream transcription start site (TSS) and different bps downstream TSS. The constant gene model considers each read having the same weight as 1.
- (4) GeneModel – TSS – Exponential: This class of models extract reads from 1K bps upstream and 100K bps downstream TSS. Gene boundaries are set so that reads from one gene body will not overlap with other gene bodies. Then, an exponential decay function is used to weight the reads from each windowed tile based on the distance to TSS. The exponential decay function is demonstrated as $exp(\frac{-abs(distance)}{window} + exp(-1))$ with different window parameters.
- (5) GeneModel – TSS – NoBoundary – Exponential: Same as (4) except no gene boundaries are set.

- (6) GeneModel – GB – Exponential: Same as (4) except the distance in the exponential decay function is calculated based on the distance to gene bodies instead of TSS. Gene boundaries are set in this class of models.

- (7) GeneModel – GB – Exponential – Extend: Same as (6) except the gene bodies are extended. The distance in the exponential decay function is calculated based on the extended gene bodies.

- (8) GeneModel – GB – NoBoundary – Exponential: Same as (6) except there are no gene boundaries limitations.

The gene score model recommended by ArchR lies in category (7). It integrates the signals from the gene body with TSS extended 5kb in the upstream direction. Then, it weights the reads outside the gene body region and use the window parameter as 10,000.

B.3 Majority voting strategy

When evaluating the choice of gene score model, we apply the majority voting strategy to 54 ArchR gene score models. We use one gene score matrix from the reference data to train the Cellcano two-round model and predict the corresponding gene score matrix from the target data. This results in 54 predictions for each cell. We then select the one with the highest vote as the final cell type. In total, we select four inter-dataset human PBMCs experiments as examples which are: (1) use PBMC_D10T1 from Granja et al. PBMCs dataset as reference to predict PBMC_Rep1 from Satpathy et al. PBMCs dataset; (2) use PBMC_D10T1 from Granja et al. PBMCs dataset as reference to predict PBMC_Rep2 from Satpathy et al. PBMCs dataset; (3) use PBMC_Rep1 from Stapathy et al. PBMCs dataset as reference to predict PBMC_D10T1 from Granja et al. PBMCs dataset; and (4) use PBMC_Rep1 from

Stapathy et al. PBMCs dataset as reference to predict PBMC_D11T1 from Granja et al. PBMCs dataset.

B.4 Details on datasets processing

We download either fragment or bam files for all datasets. We collect datasets for human PBMCs, and mouse brains listed in Appendix Table B.1.

Datasets in human PBMCs include:

- The Satpathy et al. PBMC dataset [95] is downloaded from GEO with the accession number GSE129785. It contains 4 healthy individuals labeled as PBMC_Rep1, PBMC_Rep2, PBMC_Rep3, and PBMC_Rep4. We download the fragment files for them. The cell types are annotated based on unsupervised clustering with prior biological knowledge.

- The Granja et al. PBMC dataset [34] is from a mixed-phenotype acute leukemias study (MPAL). We download the fragment files from GEO with the accession number GSE139369. We focus on the 5 replicates which contain 3 healthy donors: PBMC_D10T1, PBMC_D11T1, PBMC_D12T1, PBMC_D12T2, and PBMC_D12T3. The cell types are annotated based on Seurat SNN clustering results as well as the manually curated marker gene lists.

- The 10X PBMC dataset is downloaded from the 10X Single Cell Multiome ATAC + Gene Expression with granulocytes removed through cell sorting. We use the data with 10k cells. The dataset contains one healthy donor and the cell type annotations are obtained from MOFA pipeline [4].

- The FACS PBMC dataset [56] is available on GEO with accession number as GSE123578. Five human PBMCs cell types are sorted: CD4 T cells, CD8 T cells, B cells, Monocytes, and NK cells.

All human PBMCs datasets are mapped to human genome build hg19, except for 10X PBMC dataset, which is based on hg38. We use liftOver to map that dataset to hg19 so that all four datasets are consistent. All cell types are curated into 6 major cell types: B cells, CD4 T cells, CD8 T cells, NK cells, Monocytes and Dendritic cells.

The mouse brain datasets include:

- The Lareau et al. dataset [56] is downloaded from GEO with accession number as GSE123581. There are 2 mice in this dataset. Cell types are labeled based on the projection of another scRNA-seq mouse brain dataset. The projection is done by calculating the correlation between the promoter-region chromatin accessibility scores and gene expression on marker genes.

- The Cusanovich et al. dataset [25] is obtained from The Mouse sci-ATAC-seq Atlas (`https://atlas.gs.washington.edu/mouse-atac/data/`). We extract WholeBrainA_62216, WholeBrainA_62816, PreFrontalCortex_62216 and Cerebellum_62216 as our mouse brain samples. Cells are annotated based on unsupervised clustering and cluster-specific marker gene lists.

All mouse brain datasets are mapped to mouse genome build mm10, except for the dscATAC-seq Mouse Brain dataset, which is based on mm9. We use liftOver to lift the genome to mm10. We curate all cells into 7 major cell types including: Excitatory neurons, Inhibitory neurons, Microglia, Endothelial, Astrocyte, Oligodendrocyte and Polydendrocyte.

Table B.1: Datasets used in Cellcano study

Datasets	Organisms	Tissue	Protocol	No. cells	No. individuals (replicates)	No. cell types
Satpathy et al.	Human	PBMCs	10X Chromium	21,126	4	6
Granja et al.	Human	PBMCs	10X Chromium	8,302	3 (5)	6
10X PBMCs	Human	PBMCs	10X Single Cell Multiome ATAC + Gene Expression	11,909	1	6
FACS PBMCs	Human	PBMCs	Flow Cytometry	21,214	1	5
Lareau et al.	Mouse	Brain	dscATAC-seq	61,558	2	7
Cusanovich et al.	Mouse	Brain	sci-ATAC-seq	18,632	2 (4)	7

Note: In FACS PBMCs dataset, each cell type is extracted from different donors. Here, we consider them as one individual.

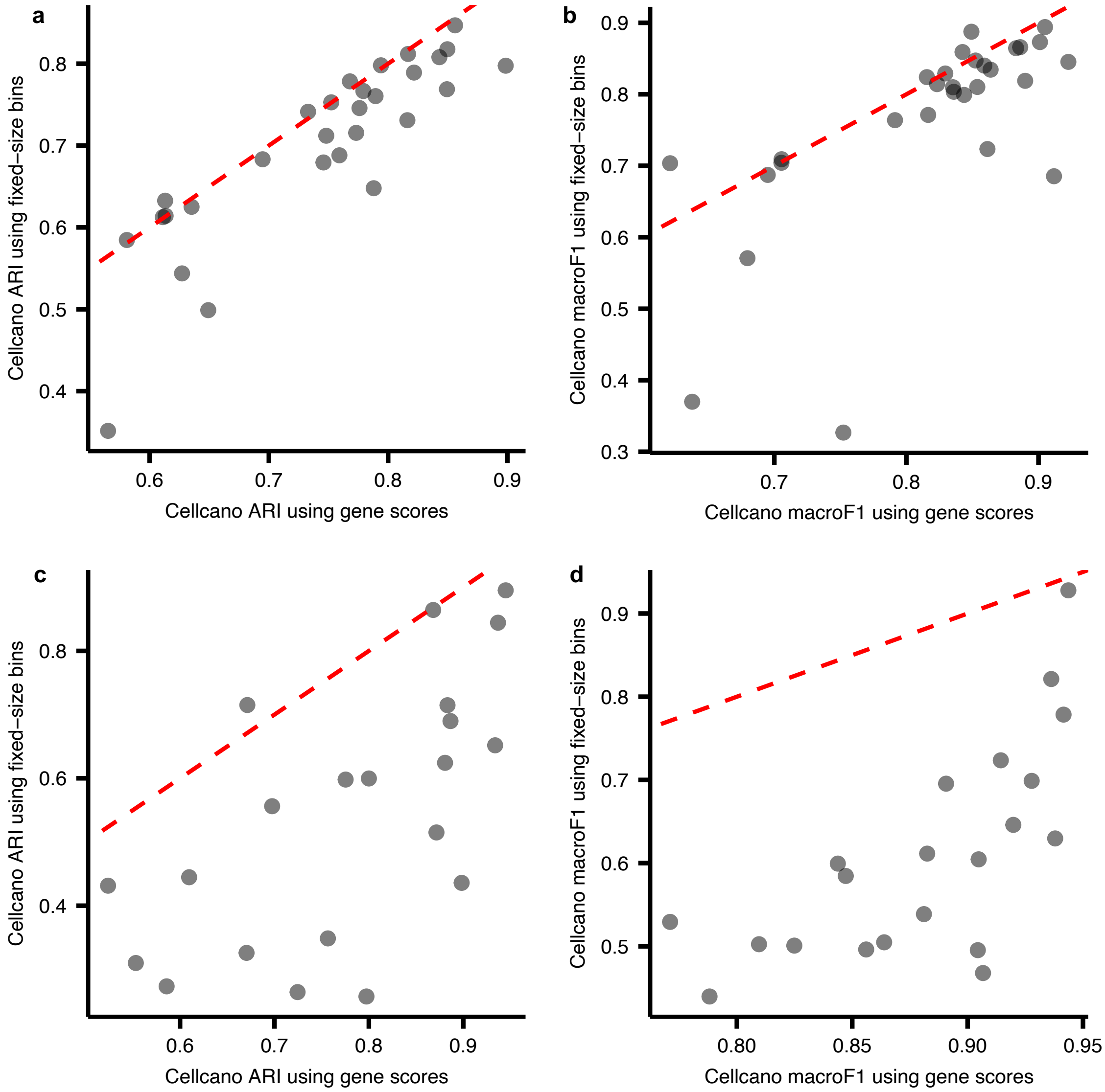


Figure B.1: (a) ARI and (b) macroF1 comparisons on n = 29 human PBMCs celltyping tasks between Cellcano with genome-wide fixed-size bins as input and Cellcano with gene scores as input. (c) ARI and (d) macroF1 comparisons on n = 21 mouse brain celltyping tasks. The dotted red lines are identity lines.

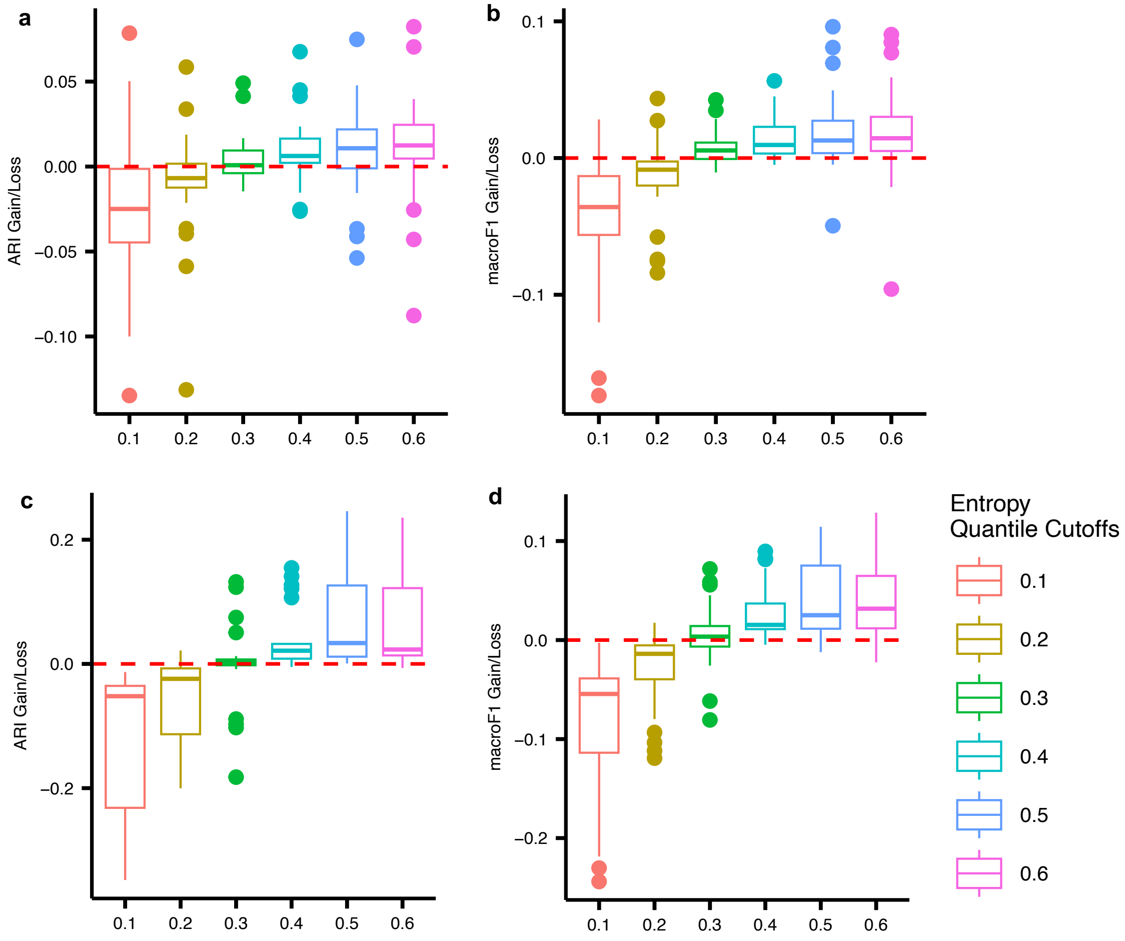


Figure B.2: (a) ARI and (b) macroF1 gains/losses using different entropy cutoffs in n = 29 human PBMCs celltyping tasks. Each box contains n = 29 prediction results. (c) ARI and (d) macroF1 gains/losses using different entropy cutoffs in n = 21 mouse brain celltyping tasks. Each box contains n = 21 prediction results.

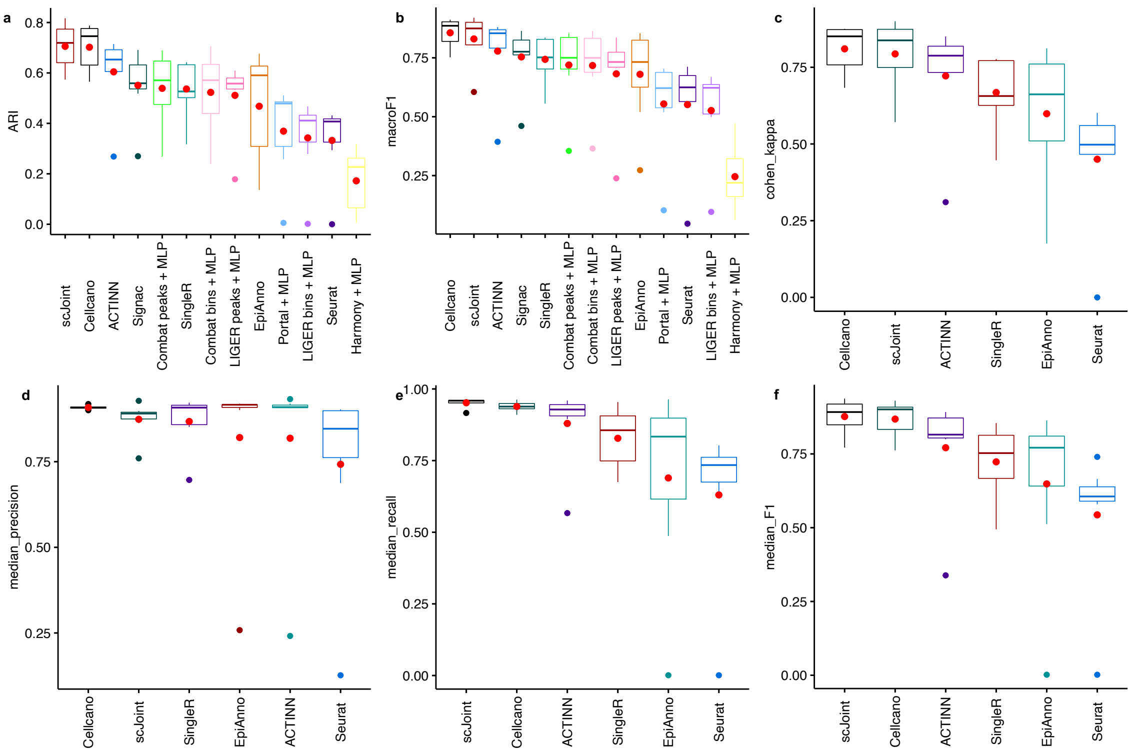


Figure B.3: (a) ARI, (b) macroF1, (c) Cohen's kappa, (d) median precision, (e) median recall and (f) median F1 comparisons on n = 7 celltyping tasks using one human PBMCs FACS-sorted dataset as the target. Each box contains n = 7 prediction results. (a)-(b) include prediction performances both from celltyping methods and integration with label transfer methods. All boxplots are ordered to have the leftmost method with the highest average performance. Note that we use red dots to indicate the mean of the data.

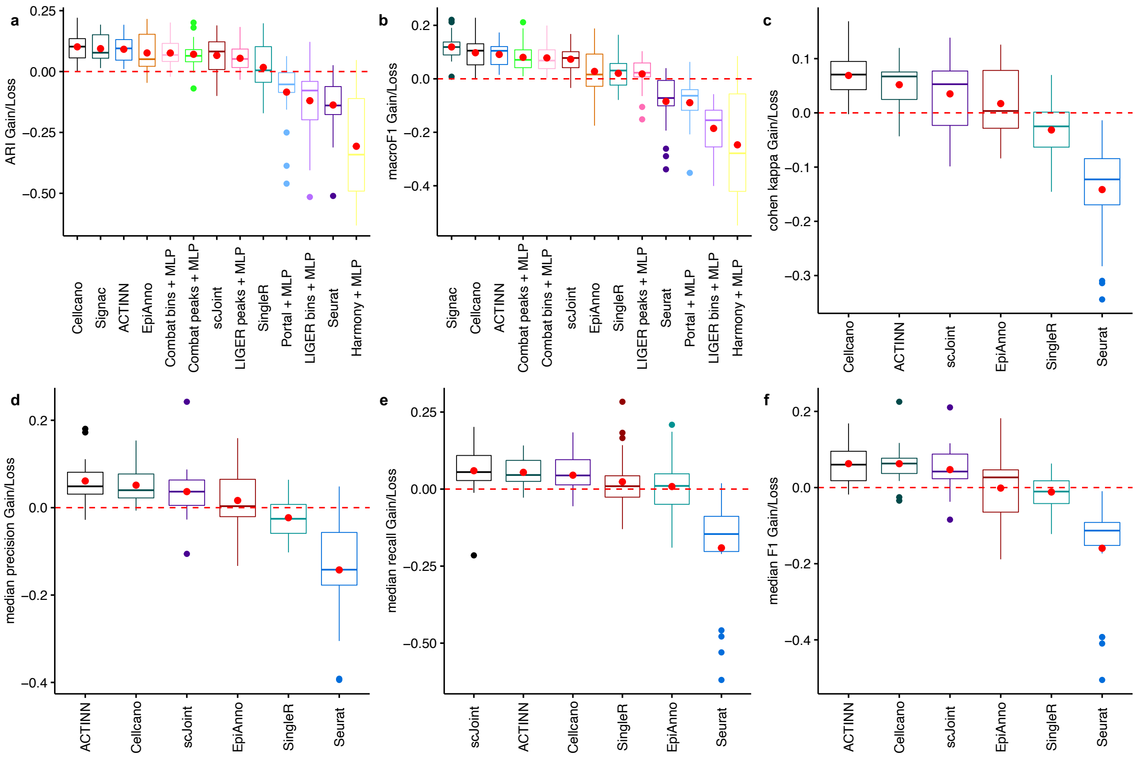


Figure B.4: (a) ARI, (b) macroF1, (c) Cohen's kappa, (d) median precision, (e) median recall, and (f) median F1 comparisons on n = 22 more human PBMCs celltyping tasks. Each box contains n = 22 prediction results. (a)-(b) include prediction performances both from celltyping methods and integration with label transfer methods. All boxplots are ordered to have the leftmost method with the highest average performance. Note that we use red dots to indicate the mean of the data.

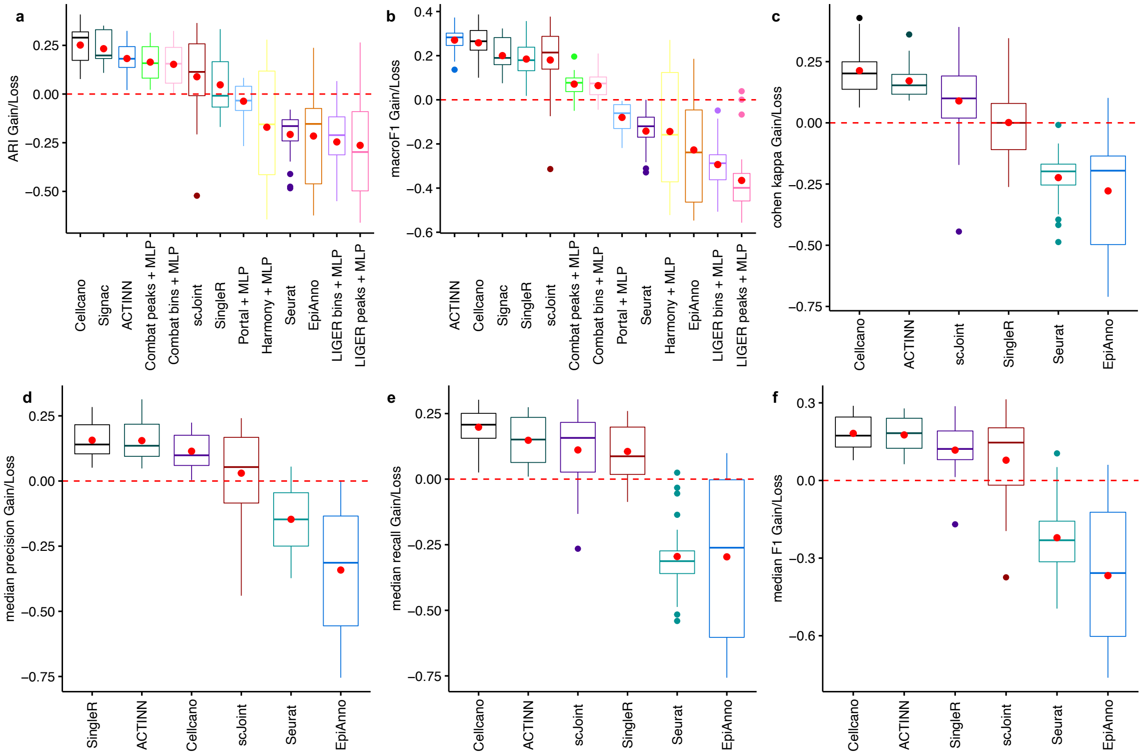


Figure B.5: (a) ARI, (b) macroF1, (c) Cohen's kappa, (d) median precision, (e) median recall, and (f) median F1 comparisons on n = 21 mouse brain celltyping tasks. Each box contains n = 21 prediction results. (a)-(b) include prediction performances both from celltyping methods and integration with label transfer methods. All boxplots are ordered to have the leftmost method with highest average performance. Note that we use red dots to indicate the mean of the data.

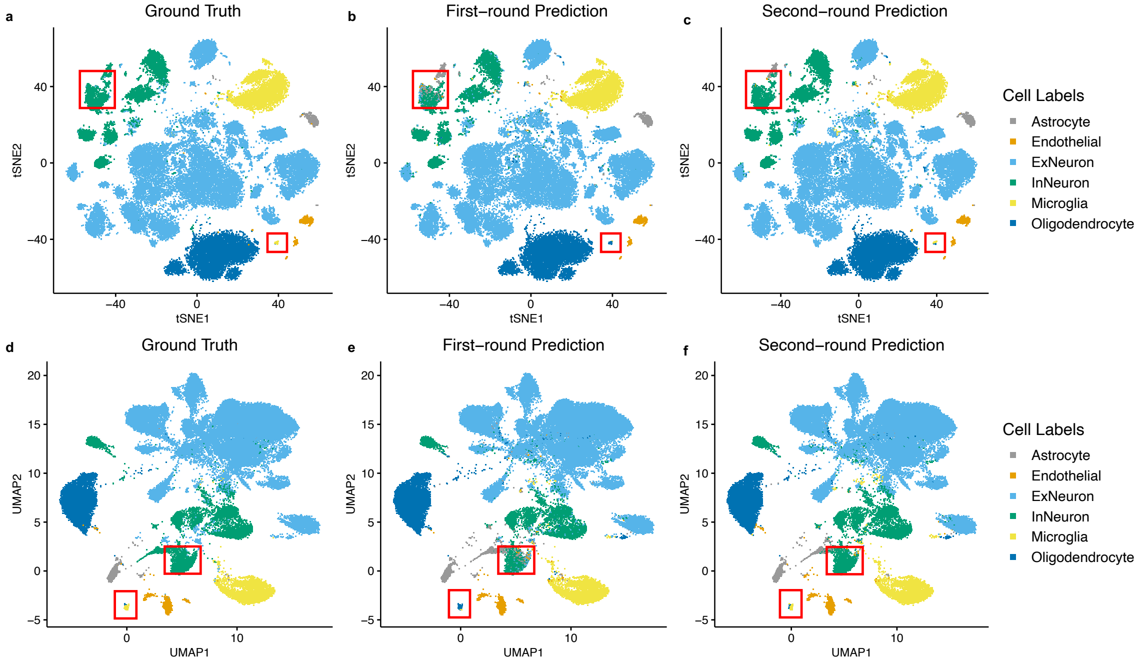


Figure B.6: (a)-(c) are tSNE visualizations and (d)-(f) are UMAP visualizations. The cells are colored with (a)(d) ground truth labels; (b)(e) Cellcano first-round predicted labels; and (c)(f) Cellcano second-round predicted labels. The red boxes indicate Cellcano's ability to correct wrongly assigned cells predicted from the first round.

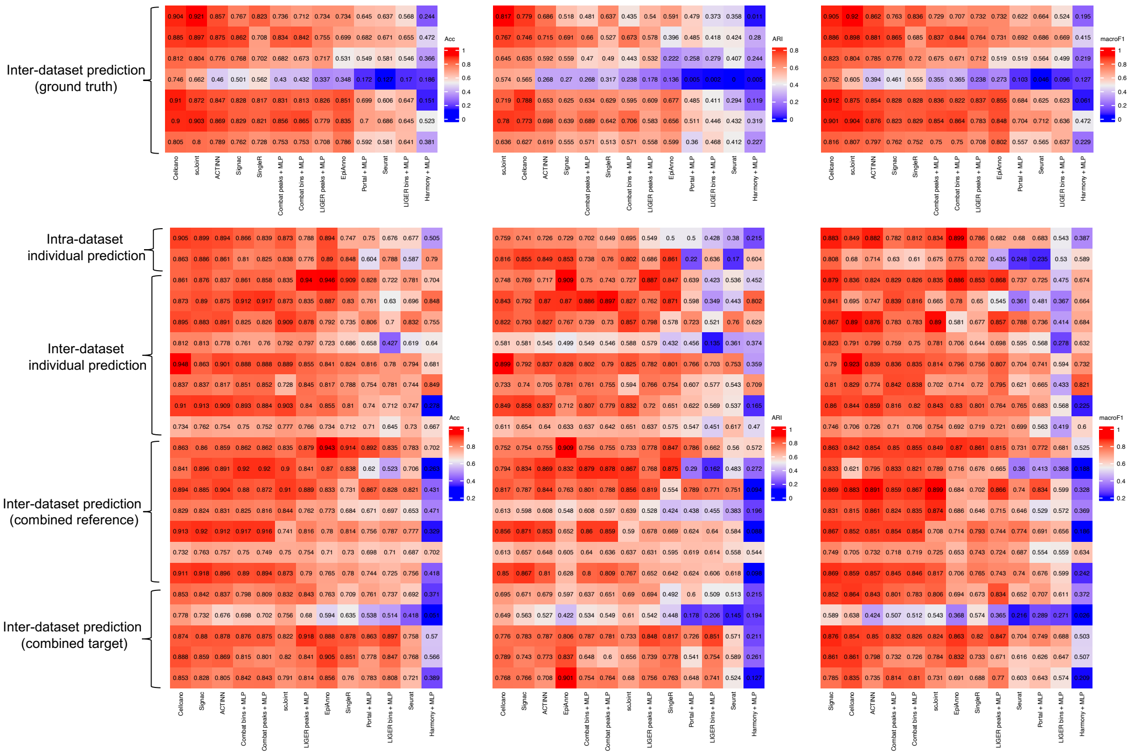


Figure B.7: Heatmap showing all prediction performances in 29 human PBMCs celltyping tasks. The celltyping tasks are labeled with corresponding categories. The heatmap is sorted to have the left most column with the highest average performance.

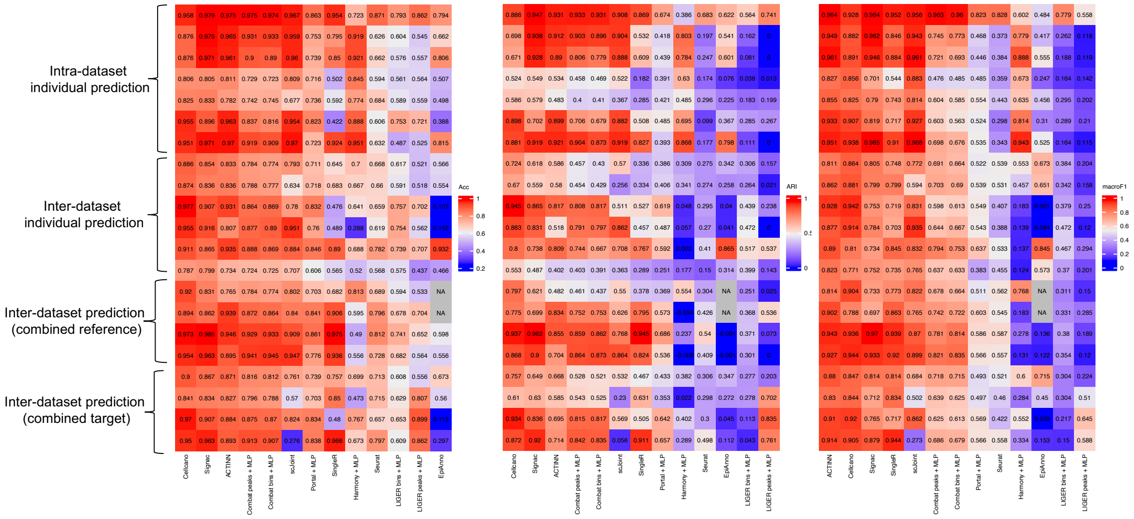


Figure B.8: Heatmap showing all prediction performances in 21 mouse brain celltyping tasks. The celltyping tasks are labeled with corresponding categories. The heatmap is sorted to have the left most column with the highest average performance. Note that EpiAnno fails to generate results for two larger celltyping tasks (denoted as NA in the figure) due to memory limit.

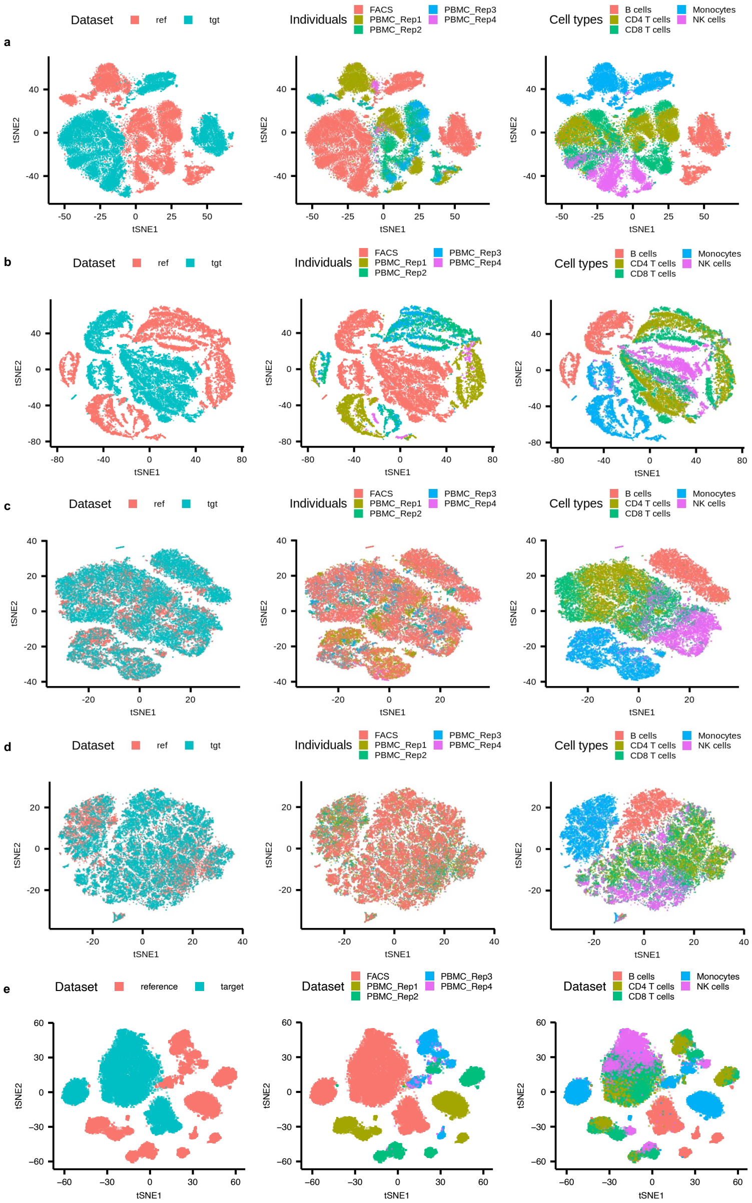


Figure B.9: Visualization on batch effect removal showing one of the celltyping tasks using one FACS-sorted dataset as target and a combination of four individuals from Satpathy et al. PBMCs dataset as reference. The left panels show the integrated datasets labeled by data source which is either from reference or target. The middle panels show the individual information, and the right panels show the cells colored by cell types. (a) shows the visualization before batch effect removal along with visualizations after batch effect removal conducted with (b) ComBat using peaks as input, (c) LIGER using peaks as input, (d) Portal using gene scores as input, and (e) Harmony using gene scores as input.

Appendix C

Appendix for Chapter 4

C.1 Bulk RNA-seq data preprocessing

The raw count of mouse bulk RNA-seq study on neurodegenerative dementia is downloaded from Gene Expression Omnibus (GEO) (Accession number: GSE90693). DE analysis is performed using DESeq2 [69] to obtain DEGs in each brain region.

The raw count of bulk RNA-seq study on PTSD is downloaded from Recount2 [24]. We extract out the experiment contrasting PTSD cases and healthy controls with time point of preemployment and perform DESeq2 to obtain DEGs.

C.2 DEGs detection using DEseq2

We first use DESeqDataSetFromMatrix() function to create an object for running DESeq2 and feed in the design matrix. We do not filter out any genes and perform DESeq() function to estimate the parameters. Then results() function is used to extract out pvalues or DEGs.

C.3 scRNA-seq data preprocessing

In this study, we include marker genes from mouse whole brain, human prefrontal cortex (pFC) and human PBMC, along with 66 cell-types' markers from four tissues (midbrain, cord blood, ovary and skeletal muscle) adopted from MSigDB. For each scRNA-seq dataset, we first retrieve raw read count matrix. Next, we filter out low-quality cells and genes and apply column-wise normalization and log transformation on the data.

The mouse whole brain scRNA-seq dataset [96] produced using the Drop-seq technology [76] contains nine brain regions from adult mice. The data provided has already been prefiltered by the authors. For cell types other than neurons, we directly utilize the information provided on the study website (`http://dropviz.org/`). For neurons and interneurons, we curate the sub-cell types following the original study.

The human pFC scRNA-seq dataset [82], produced by 10X Genomics Chromium, is derived from the pFC region (specifically BA9). The dataset contains two conditions: healthy controls and major depressive disorder. We split the data matrix into two parts and filter out cells expressing less than 10 genes and genes expressed in less than 10 cells, respectively. We also filter out mitochondrial, ribosomal genes and genes from annotation clusters (Astros_1, Mix_1, Mix_2, Mix_3, Mix_4, Mix_5 and Inhib_4_SST).

The PBMC dataset [39], generated by CITE-seq technology [99], is derived from an HIV vaccine trial study which involves eight volunteers at three time points: immediately before, three days and seven days after the vaccine. The study contains 161 764 cells in total. To accelerate the marker gene selection, we separate the count matrix according to the time label and filter out low-quality cells and genes (mitochondrial, ribosomal genes and those expressed in less than 1000 cells). The cluster annotated as 'Doublet' is filtered out.

C.4 MSigDB marker genes

We download cell marker gene sets from MSigDB category C8-cell type signature gene sets. Since not all tissue types are suitable for LRcell, we apply the following criteria to select tissues: (i) nonfetal tissues; (ii) have more than eight sub-cell types; (iii) minimum number of marker gene greater than 50 and (iv) median number of marker genes greater than 80. In the end, four tissue types remain: the midbrain, cord blood, ovary and skeletal muscle.

C.5 Differential expressed genes (DEGs) detection using Limma-Voom

We first use the calcNormFactors() function to calculate normalization factors. For comparison purpose, since we do not filter out genes during DESeq2 analysis, we do not filter out any genes when conducting Limma-Voom analysis. We then design the design matrix and use voom() function to calculate the mean-variance trend. Next, a linear model is fit using lmFit() function and group contrast is calculated using makeContrasts() function. Then contrast for each gene is estimated by contrasts.fit() function along with smoothing of standard errors with eBayes(). Finally, topTable() function is used to extract out p-values of DEGs.

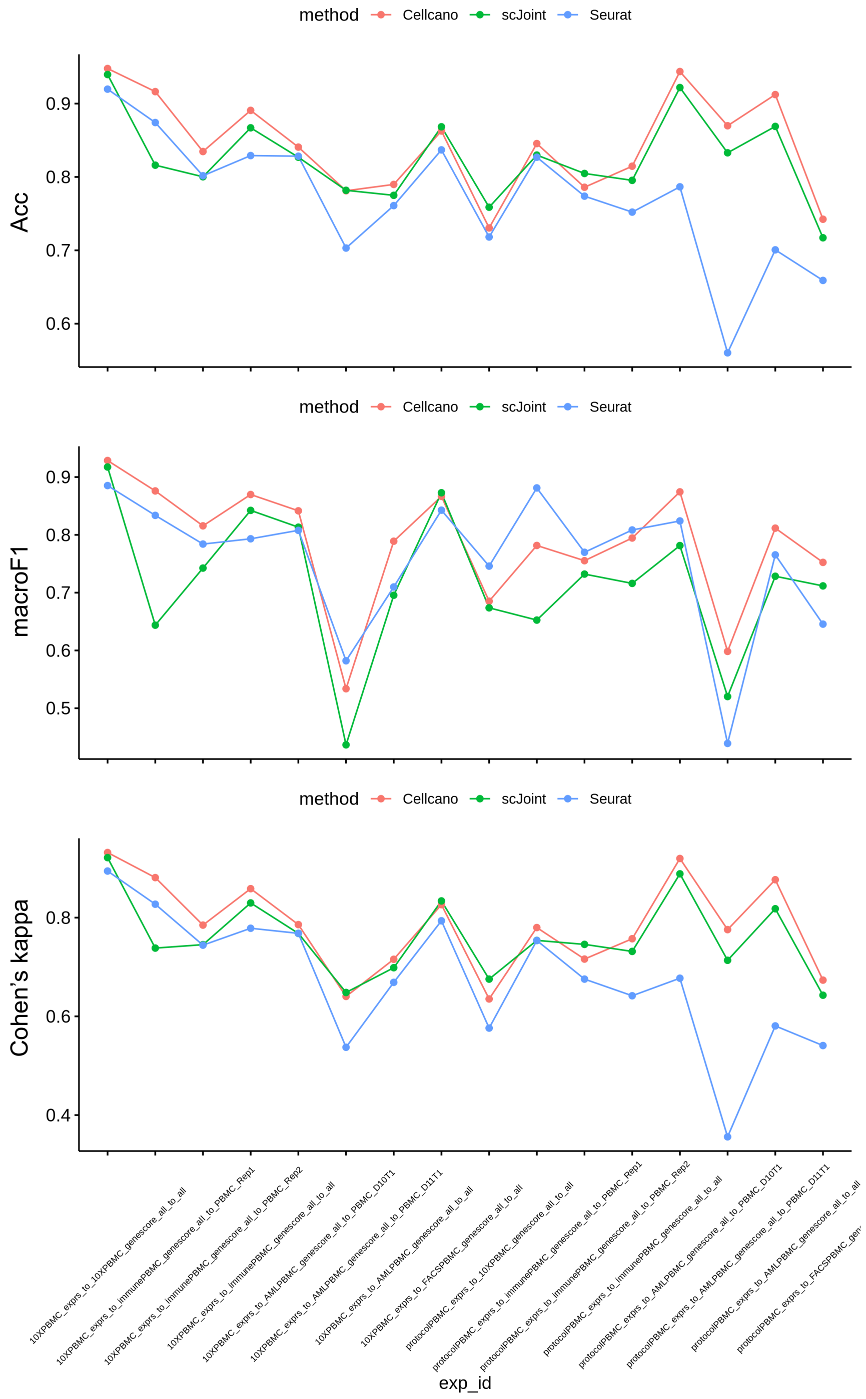


Figure C.1: Line plots show the comparisons of CellAMA, scJoint, and Seurat in Accuracy, macroF1, and Cohen's kappa for each celltyping prediction task.

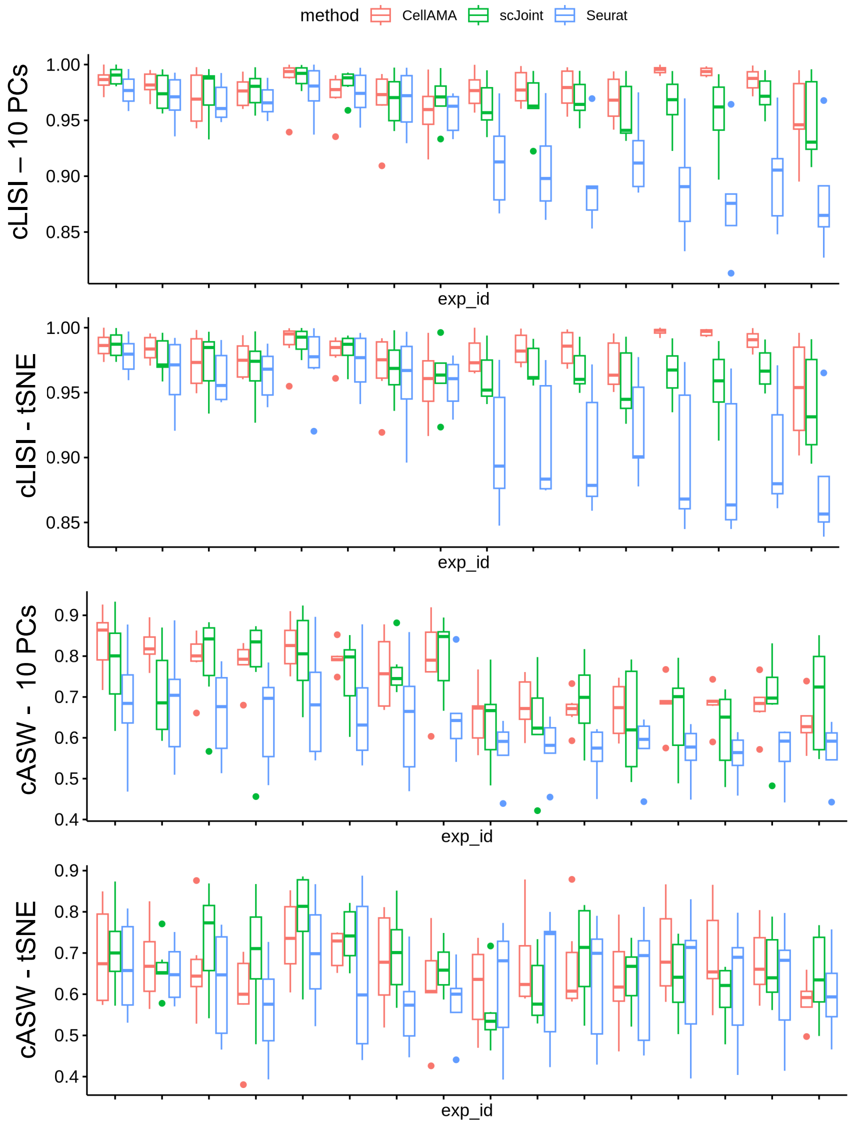


Figure C.2: Each box contains the number of cell types in each celltyping task, and the metrics are evaluated using the top 10 PCs and tSNE as input as indicated in the y-axis.

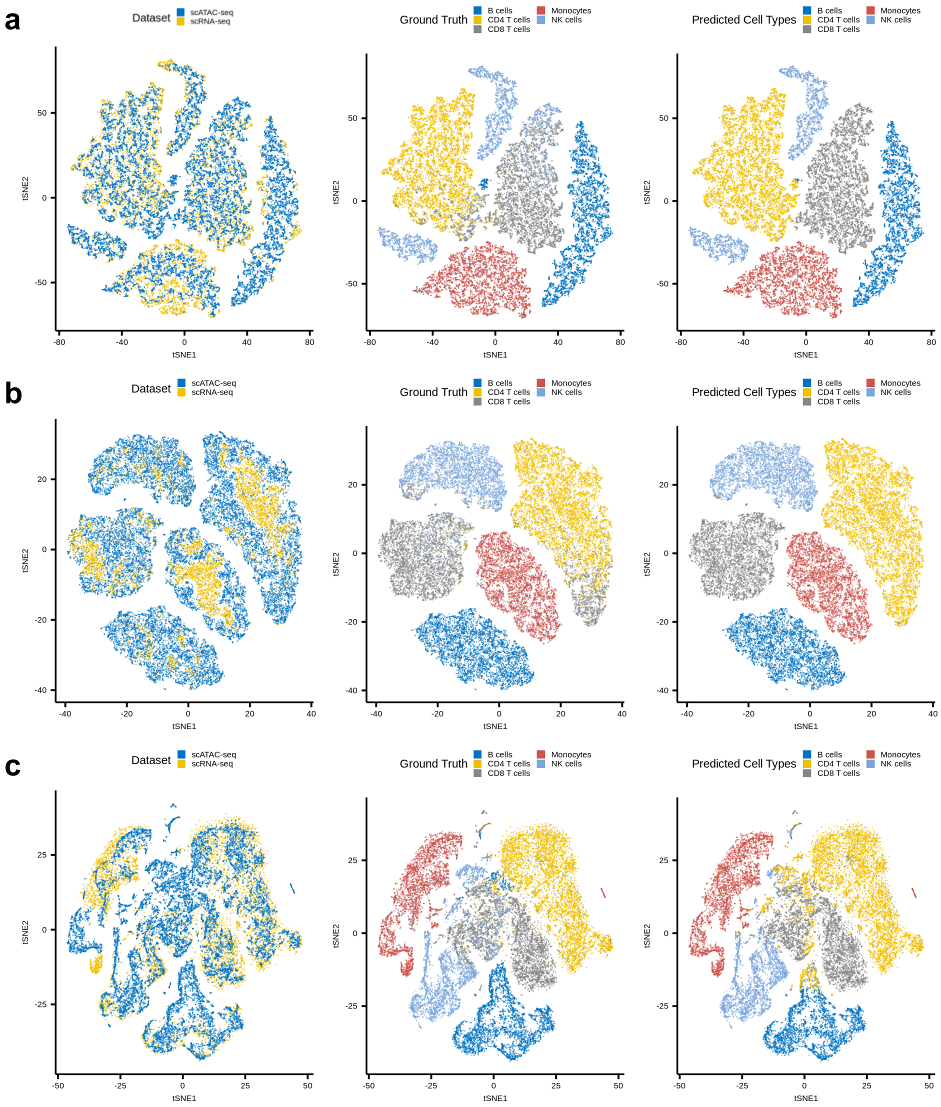


Figure C.3: tSNE visualization of (a) CellAMA, (b) scJoint, and (c) Seurat V3. The left panels indicate the modality of the datasets, either scRNA-seq data or scATAC-seq data. The middle panels show the ground truth cell label information and the right panels show the predicted cell types.

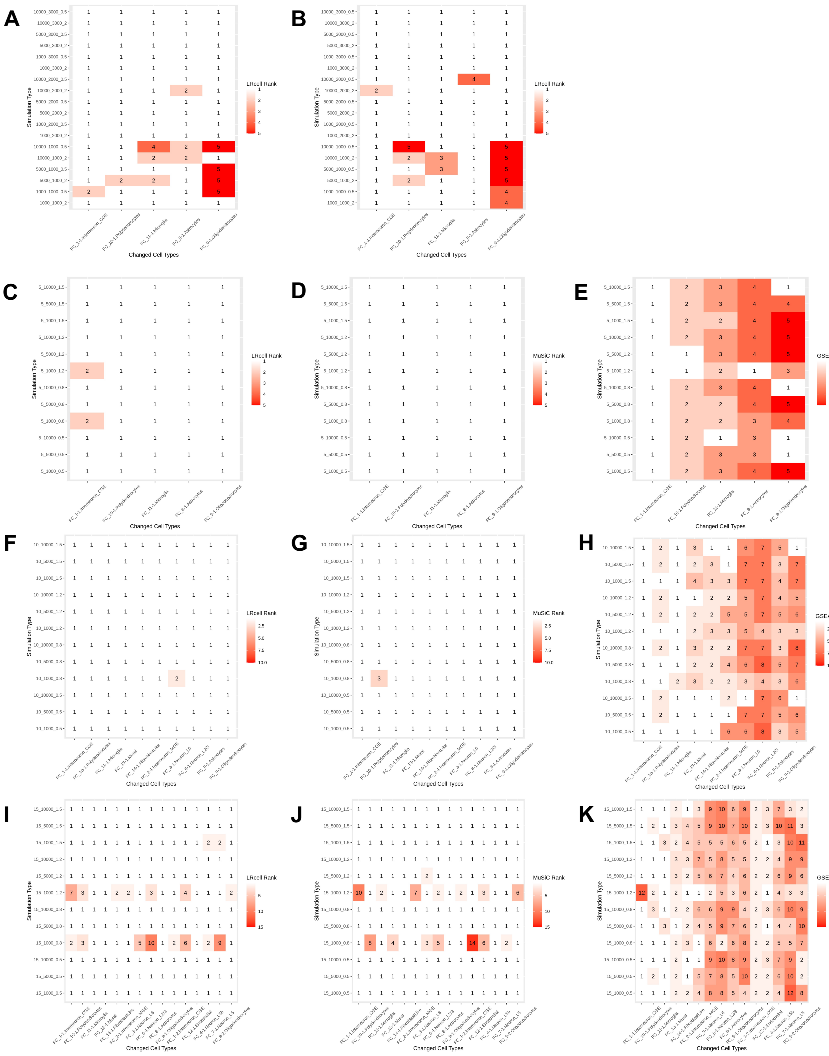


Figure C.4: LRcell analysis results obtained from simulated data. (A)-(B): LRcell analysis result from data simulated under the scenario that there are DEGs in selected sub-cell type but the proportions of all sub-cell types remain the same. (A) The proportions of all sub-cell types are evenly distributed; (B) The proportion of all sub-cell types are not evenly distributed. (C)-(E): Comparison of results obtained using different methods on data simulated under the scenario that only proportions of the five sub-cell types have changed. (C) LRcell result. (D) MuSiC result. (E) GSEA result. (F)-(H): Comparison of results obtained using different methods on data simulated under the scenario that only proportions of the 10 sub-cell types have changed. (F) LRcell result. (G) MuSiC result. (H) GSEA result. (I)-(K): Comparison of results obtained using different methods on data simulated under the scenario that only proportions of the 15 sub-cell types have changed. (I) LRcell result. (J) MuSiC result. (K) GSEA result. Here, rank 1 indicates the changes are correctly identified where the actual change in the specific sub-cell type among the 5, 10 or 15 sub-cell types is correctly captured.

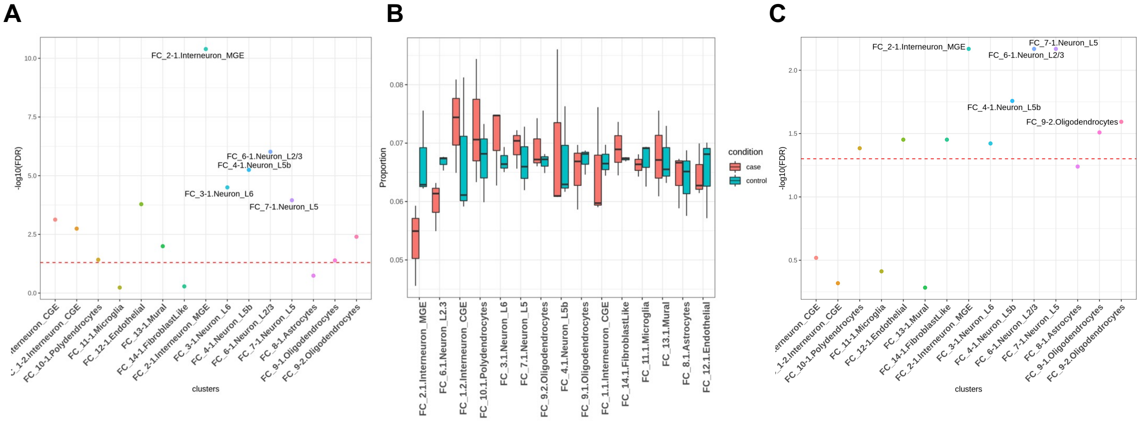


Figure C.5: Comparison of results obtained using different methods on data simulated under a specific simulation experiment scenario. There are 15 sub-cell types were involved and 1,000 cells were simulated. In the simulation experiment, only FC_1-1.Interneuron_CGE's proportion was increased by 20% from the original proportion. (A) LRcell result. (B) MuSiC result. (C) GSEA result.

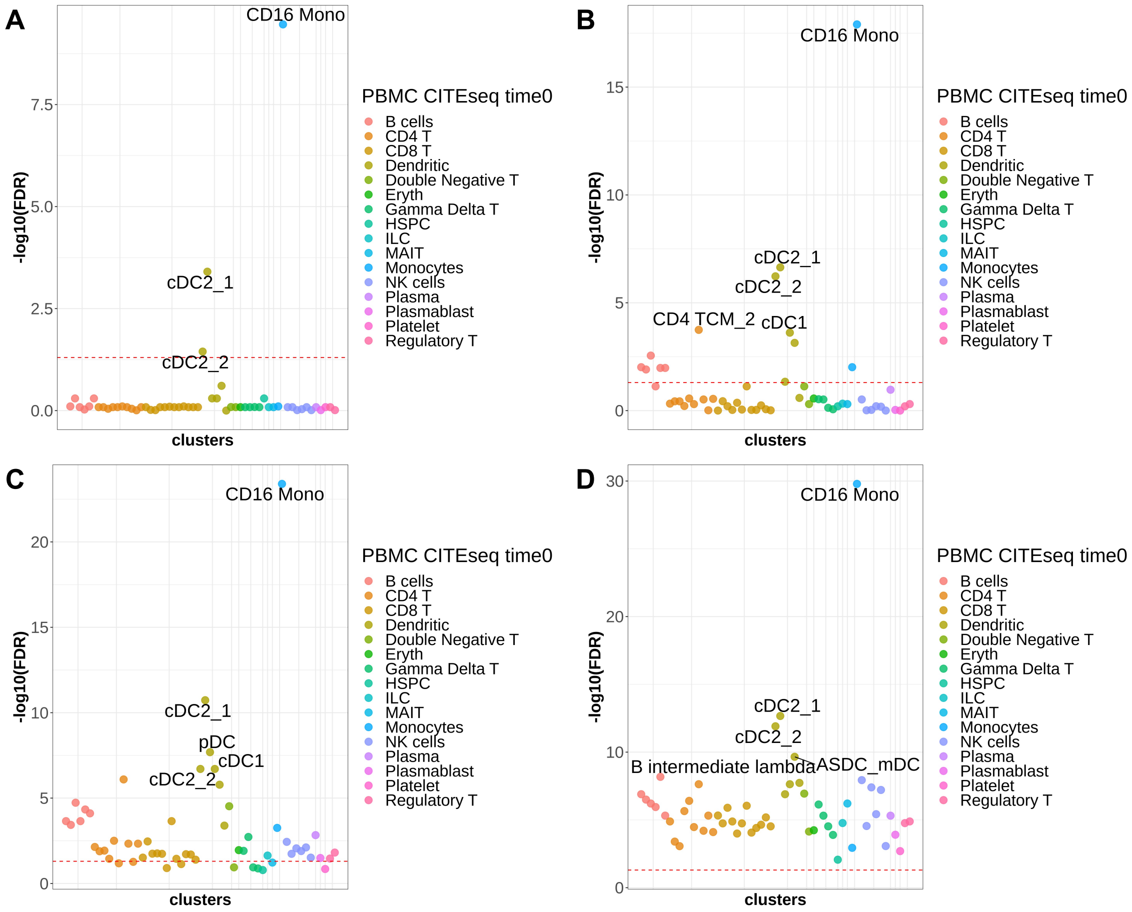


Figure C.6: Robustness of LRcell results with different number of marker genes on the PTSD study. LRcell results using (A) top 50, (B) top 300, (C) top 600 and (D) top 1,000 marker genes derived from the PBMC dataset.

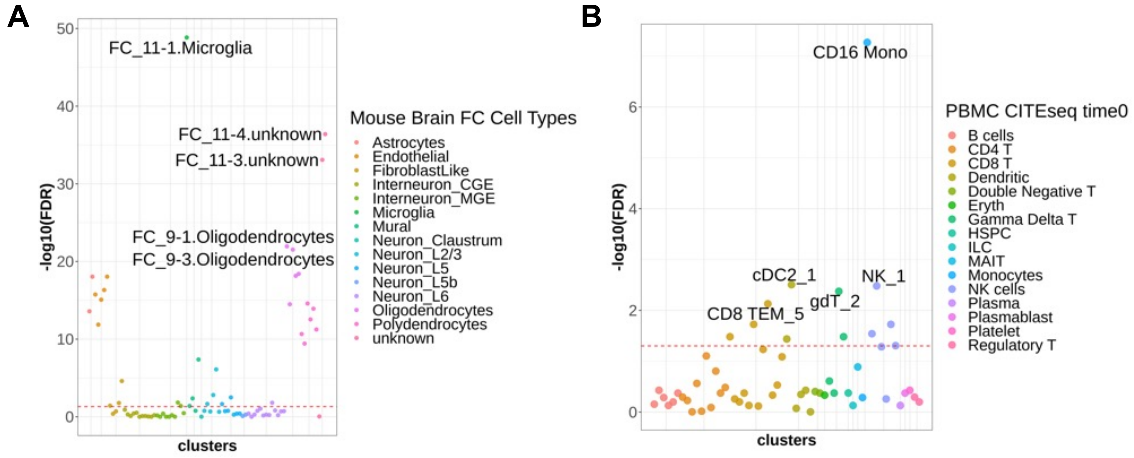


Figure C.7: LRcell result obtained on DEGs detected by the Limma-Voom method. (A) LRcell result on the AD study using DEG p-values calculated by Limma-Voom. (B) LRcell result on the PTSD study using DEG p-values calculated by Limma-Voom. We did not filter out any gene and default parameters of Limma-Voom were used.

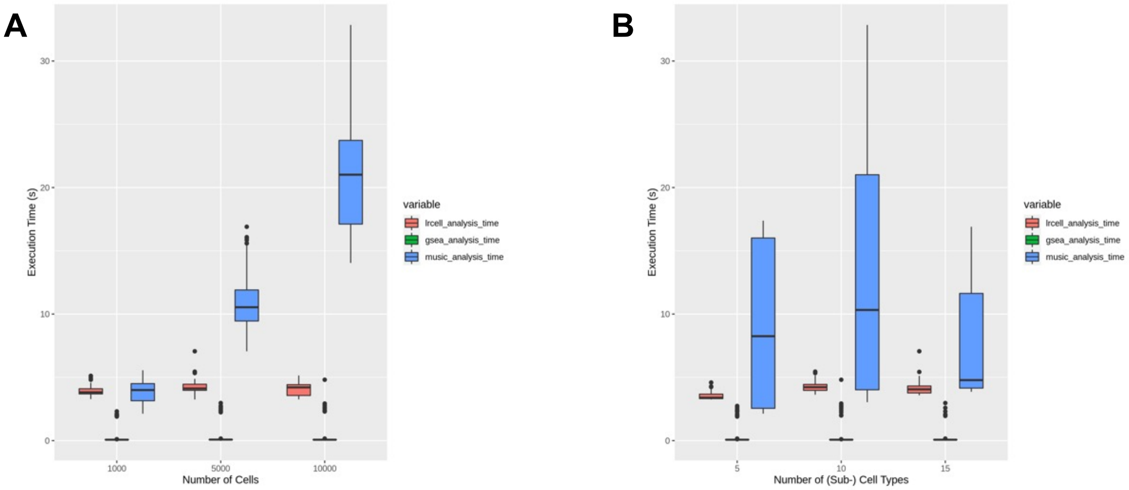


Figure C.8: Comparison of computing time of LRcell, GSEA and MuSiC when analyzing simulated data. (A) Different number of cells in the simulated dataset. (B) Different number of sub-cell types in the simulated dataset.

Bibliography

[1] Tamim Abdelaal, Lieke Michielsen, Davy Cats, Dylan Hoogduin, Hailiang Mei, Marcel JT Reinders, and Ahmed Mahfouz. A comparison of automatic cell identification methods for single-cell rna sequencing data. *Genome biology*, 20 (1):1–19, 2019.

[2] Jose Alquicira-Hernandez, Anuja Sathe, Hanlee P Ji, Quan Nguyen, and Joseph E Powell. scpred: accurate supervised method for cell-type classification from single-cell rna-seq data. *Genome biology*, 20(1):1–17, 2019.

[3] Dvir Aran, Agnieszka P Looney, Leqian Liu, Esther Wu, Valerie Fong, Austin Hsu, Suzanna Chak, Ram P Naikawadi, Paul J Wolters, Adam R Abate, et al. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nature immunology*, 20(2):163–172, 2019.

[4] Ricard Argelaguet, Britta Velten, Damien Arnol, Sascha Dietrich, Thorsten Zenz, John C Marioni, Florian Buettner, Wolfgang Huber, and Oliver Stegle. Multi-omics factor analysis—a framework for unsupervised integration of multi-omics data sets. *Molecular systems biology*, 14(6):e8124, 2018.

[5] Ricard Argelaguet, Anna SE Cuomo, Oliver Stegle, and John C Marioni. Computational principles and challenges in single-cell data integration. *Nature biotechnology*, 39(10):1202–1215, 2021.

[6] Francisco Avila Cobos, José Alquicira-Hernandez, Joseph E Powell, Pieter Mestdagh, and Katleen De Preter. Benchmarking of cell type deconvolution pipelines for transcriptomics data. *Nature communications*, 11(1):1–14, 2020.

[7] Seungbyn Baek and Insuk Lee. Single-cell atac sequencing analysis: From data preprocessing to hypothesis generation. *Computational and structural biotechnology journal*, 18:1429–1439, 2020.

[8] Syed Murtuza Baker, Connor Rogerson, Andrew Hayes, Andrew D Sharrocks, and Magnus Rattray. Classifying cells with scasat, a single-cell atac-seq analysis tool. *Nucleic acids research*, 47(2):e10–e10, 2019.

[9] Marek Bartosovic, Mukund Kabbe, and Gonçalo Castelo-Branco. Single-cell cut&tag profiles histone modifications and transcription factors in complex tissues. *Nature biotechnology*, 39(7):825–835, 2021.

[10] Aritra Bhattacherjee, Mohamed Nadhir Djekidel, Renchao Chen, Wenqiang Chen, Luis M Tuesta, and Yi Zhang. Cell type-specific transcriptional programs in mouse prefrontal cortex during adolescence and addiction. *Nature communications*, 10(1):1–18, 2019.

[11] Carmen Bravo González-Blas, Liesbeth Minnoye, Dafni Papasokrati, Sara Aibar, Gert Hulselmans, Valerie Christiaens, Kristofer Davie, Jasper Wouters, and Stein Aerts. cistopic: cis-regulatory topic modeling on single-cell atac-seq data. *Nature methods*, 16(5):397–400, 2019.

[12] Maria Brbić, Marinka Zitnik, Sheng Wang, Angela O Pisco, Russ B Altman, Spyros Darmanis, and Jure Leskovec. Mars: discovering novel cell types across heterogeneous single-cell experiments. *Nature methods*, 17(12):1200–1206, 2020.

[13] Michael S Breen, Adam X Maihofer, Stephen J Glatt, Daniel S Tylee, Sharon D Chandler, Ming T Tsuang, Victoria B Risbrough, Dewleen G Baker, Daniel T

O'Connor, Caroline M Nievergelt, et al. Gene networks specific for innate immunity define post-traumatic stress disorder. *Molecular psychiatry*, 20(12): 1538–1545, 2015.

[14] Jason D Buenrostro, Beijing Wu, Ulrike M Litzenburger, Dave Ruff, Michael L Gonzales, Michael P Snyder, Howard Y Chang, and William J Greenleaf. Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature*, 523(7561):486–490, 2015.

[15] Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*, 36(5):411–420, 2018.

[16] Maren Büttner, Zhichao Miao, F Alexander Wolf, Sarah A Teichmann, and Fabian J Theis. A test metric for assessing single-cell rna-seq batch correction. *Nature methods*, 16(1):43–49, 2019.

[17] Junyue Cao, Malte Spielmann, Xiaojie Qiu, Xingfan Huang, Daniel M Ibrahim, Andrew J Hill, Fan Zhang, Stefan Mundlos, Lena Christiansen, Frank J Steemers, et al. The single-cell transcriptional landscape of mammalian organogenesis. *Nature*, 566(7745):496–502, 2019.

[18] Zhi-Jie Cao and Ge Gao. Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. *Nature Biotechnology*, 40(10):1458–1466, 2022.

[19] Huidong Chen, Caleb Lareau, Tommaso Andreani, Michael E Vinyard, Sara P Garcia, Kendell Clement, Miguel A Andrade-Navarro, Jason D Buenrostro, and Luca Pinello. Assessment of computational methods for the analysis of single-cell atac-seq data. *Genome biology*, 20(1):1–25, 2019.

[20] Liang Chen, Qiuyan He, Yuyao Zhai, and Minghua Deng. Single-cell rna-seq data semi-supervised clustering and annotation via structural regularized domain adaptation. *Bioinformatics*, 37(6):775–784, 2021.

[21] Song Chen, Blue B Lake, and Kun Zhang. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nature biotechnology*, 37(12):1452–1457, 2019.

[22] Xiaoyang Chen, Shengquan Chen, Shuang Song, Zijing Gao, Lin Hou, Xuegong Zhang, Hairong Lv, and Rui Jiang. Cell type annotation of single-cell chromatin accessibility data via supervised bayesian embedding. *Nature Machine Intelligence*, 4(2):116–126, 2022.

[23] Stephen J Clark, Ricard Argelaguet, Chantriolnt-Andreas Kapourani, Thomas M Stubbs, Heather J Lee, Celia Alda-Catalinas, Felix Krueger, Guido Sanguinetti, Gavin Kelsey, John C Marioni, et al. scnmt-seq enables joint profiling of chromatin accessibility dna methylation and transcription in single cells. *Nature communications*, 9(1):781, 2018.

[24] Leonardo Collado-Torres, Abhinav Nellore, Kai Kammers, Shannon E Ellis, Margaret A Taub, Kasper D Hansen, Andrew E Jaffe, Ben Langmead, and Jeffrey T Leek. Reproducible rna-seq analysis using recount2. *Nature biotechnology*, 35(4):319–321, 2017.

[25] Darren A Cusanovich, Andrew J Hill, Delasa Aghamirzaie, Riza M Daza, Hannah A Pliner, Joel B Berletch, Galina N Filippova, Xingfan Huang, Lena Christiansen, William S DeWitt, et al. A single-cell atlas of in vivo mammalian chromatin accessibility. *Cell*, 174(5):1309–1324, 2018.

[26] Hazel M Davey and Douglas B Kell. Flow cytometry and cell sorting of hetero-

geneous microbial populations: the importance of single-cell analyses. *Microbiological reviews*, 60(4):641–696, 1996.

[27] Jurrian K De Kanter, Philip Lijnzaad, Tito Candelli, Thanasis Margaritis, and Frank CP Holstege. Chetah: a selective, hierarchical cell type identification method for single-cell rna sequencing. *Nucleic acids research*, 47(16):e95–e95, 2019.

[28] Jiarui Ding, Xian Adiconis, Sean K Simmons, Monika S Kowalczyk, Cynthia C Hession, Nemanja D Marjanovic, Travis K Hughes, Marc H Wadsworth, Tyler Burks, Lan T Nguyen, et al. Systematic comparison of single-cell and single-nucleus rna-sequencing methods. *Nature biotechnology*, 38(6):737–746, 2020.

[29] N Editorial. Method of the year 2013. *Nat. Methods*, 11(1):1, 2014.

[30] Rongxin Fang, Sebastian Preissl, Yang Li, Xiaomeng Hou, Jacinta Lucero, Xinxin Wang, Amir Motamedi, Andrew K Shiau, Xinzhu Zhou, Fangming Xie, et al. Comprehensive analysis of single cell atac-seq data with snapatac. *Nature communications*, 12(1):1–15, 2021.

[31] Laiyi Fu, Lihua Zhang, Emmanuel Dollinger, Qinke Peng, Qing Nie, and Xiaohui Xie. Predicting transcription factor binding in single cells through deep learning. *Science advances*, 6(51):eaba9031, 2020.

[32] Renaud Gaujoux and Cathal Seoighe. Semi-supervised nonnegative matrix factorization for gene expression deconvolution: a case study. *Infection, Genetics and Evolution*, 12(5):913–921, 2012.

[33] Ting Gong and Joseph D Szustakowski. Deconrnaseq: a statistical framework for deconvolution of heterogeneous tissue samples based on mrna-seq data. *Bioinformatics*, 29(8):1083–1085, 2013.

[34] Jeffrey M Granja, Sandy Klemm, Lisa M McGinnis, Arwa S Kathiria, Anja Mezger, M Ryan Corces, Benjamin Parks, Eric Gars, Michaela Liedtke, Grace XY Zheng, et al. Single-cell multiomic analysis identifies regulatory programs in mixed-phenotype acute leukemia. *Nature biotechnology*, 37(12): 1458–1465, 2019.

[35] Jeffrey M Granja, M Ryan Corces, Sarah E Pierce, S Tansu Bagdatli, Hani Choudhry, Howard Y Chang, and William J Greenleaf. Archr is a scalable software package for integrative single-cell chromatin accessibility analysis. *Nature genetics*, 53(3):403–411, 2021.

[36] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006.

[37] Hongyu Guo and Jun Li. scsorter: assigning cells to known cell types according to marker genes. *Genome biology*, 22(1):1–18, 2021.

[38] Laleh Haghverdi, Aaron TL Lun, Michael D Morgan, and John C Marioni. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology*, 36(5):421–427, 2018.

[39] Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M Mauck III, Shiwei Zheng, Andrew Butler, Maddie J Lee, Aaron J Wilk, Charlotte Darby, Michael Zager, et al. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587, 2021.

[40] Lukas Heumos, Anna C Schaar, Christopher Lance, Anastasia Litinetskaya, Felix Drost, Luke Zappia, Malte D Lücken, Daniel C Strobl, Juan Henao, Fabiola Curion, et al. Best practices for single-cell analysis across modalities. *Nature Reviews Genetics*, pages 1–23, 2023.

[41] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[42] Wenpin Hou, Zhicheng Ji, Hongkai Ji, and Stephanie C Hicks. A systematic evaluation of single-cell rna-sequencing imputation methods. *Genome biology*, 21(1):1–30, 2020.

[43] Jian Hu, Xiangjie Li, Gang Hu, Yafei Lyu, Katalin Susztak, and Mingyao Li. Iterative transfer learning with neural network for clustering and cell type classification in single-cell rna-seq analysis. *Nature machine intelligence*, 2(10): 607–618, 2020.

[44] Mo Huang, Jingshu Wang, Eduardo Torre, Hannah Dueck, Sydney Shaffer, Roberto Bonasio, John I Murray, Arjun Raj, Mingyao Li, and Nancy R Zhang. Saver: gene expression recovery for single-cell rna sequencing. *Nature methods*, 15(7):539–542, 2018.

[45] Qianhui Huang, Yu Liu, Yuheng Du, and Lana X Garmire. Evaluation of cell type annotation r packages on single-cell rna-seq data. *Genomics, proteomics & bioinformatics*, 19(2):267–281, 2021.

[46] Haijing Jin and Zhandong Liu. A benchmark for rna-seq deconvolution analysis under dynamic testing environments. *Genome biology*, 22(1):1–23, 2021.

[47] Hyun Min Kang, Meena Subramaniam, Sasha Targ, Michelle Nguyen, Lenka Maliskova, Elizabeth McCarthy, Eunice Wan, Simon Wong, Lauren Byrnes, Cristina M Lanata, et al. Multiplexed droplet single-cell rna-sequencing using natural genetic variation. *Nature biotechnology*, 36(1):89–94, 2018.

[48] Jacob C Kimmel and David R Kelley. Semisupervised adversarial neural networks for single-cell classification. *Genome research*, 31(10):1781–1793, 2021.

[49] Vladimir Yu Kiselev, Kristina Kirschner, Michael T Schaub, Tallulah Andrews, Andrew Yiu, Tamir Chandra, Kedar N Natarajan, Wolf Reik, Mauricio Barahona, Anthony R Green, et al. Sc3: consensus clustering of single-cell rna-seq data. *Nature methods*, 14(5):483–486, 2017.

[50] Vladimir Yu Kiselev, Andrew Yiu, and Martin Hemberg. scmap: projection of single-cell rna-seq data across data sets. *Nature methods*, 15(5):359–362, 2018.

[51] Vladimir Yu Kiselev, Tallulah S Andrews, and Martin Hemberg. Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 20(5):273–282, 2019.

[52] Yunchuan Kong and Tianwei Yu. A graph-embedded deep feedforward network for disease outcome classification and feature selection using gene expression data. *Bioinformatics*, 34(21):3727–3737, 2018.

[53] Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature methods*, 16(12):1289–1296, 2019.

[54] Rachel M Kratofil, Paul Kubes, and Justin F Deniset. Monocyte conversion during inflammation and injury. *Arteriosclerosis, thrombosis, and vascular biology*, 37(1):35–42, 2017.

[55] Pei-Fen Kuan, Xiaohua Yang, Sean Clouston, Xu Ren, Roman Kotov, Monika Waszczuk, Prashant K Singh, Sean T Glenn, Eduardo Cortes Gomez, Jianmin Wang, et al. Cell type-specific gene expression patterns associated with posttraumatic stress disorder in world trade center responders. *Translational psychiatry*, 9(1):1–11, 2019.

[56] Caleb A Lareau, Fabiana M Duarte, Jennifer G Chew, Vinay K Kartha, Zach D Burkett, Andrew S Kohlway, Dmitry Pokholok, Martin J Aryee, Frank J Steemers, Ronald Lebofsky, et al. Droplet-based combinatorial indexing for massive-scale single-cell chromatin accessibility. *Nature Biotechnology*, 37(8): 916–924, 2019.

[57] Charity W Law, Yunshun Chen, Wei Shi, and Gordon K Smyth. voom: Precision weights unlock linear model analysis tools for rna-seq read counts. *Genome biology*, 15(2):1–17, 2014.

[58] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[59] Bo Li, Eric Severson, Jean-Christophe Pignon, Haoquan Zhao, Taiwen Li, Jesse Novak, Peng Jiang, Hui Shen, Jon C Aster, Scott Rodig, et al. Comprehensive analyses of tumor immunity: implications for cancer immunotherapy. *Genome biology*, 17(1):1–16, 2016.

[60] Ziyi Li and Hao Wu. Toast: improving reference-free cell composition estimation by cross-cell type differential analysis. *Genome biology*, 20(1):1–17, 2019.

[61] Arthur Liberzon, Chet Birger, Helga Thorvaldsdóttir, Mahmoud Ghandi, Jill P Mesirov, and Pablo Tamayo. The molecular signatures database hallmark gene set collection. *Cell systems*, 1(6):417–425, 2015.

[62] Wei-Chao Lin, Chih-Fong Tsai, Ya-Han Hu, and Jing-Shang Jhang. Clustering-based undersampling in class-imbalanced data. *Information Sciences*, 409:17–26, 2017.

[63] Yingxin Lin, Tung-Yu Wu, Sheng Wan, Jean YH Yang, Wing H Wong, and

YX Rachel Wang. scjoint integrates atlas-scale single-cell rna-seq and atac-seq data with transfer learning. *Nature biotechnology*, 40(5):703–710, 2022.

[64] Sten Linnarsson and Sarah A Teichmann. Single-cell genomics: coming of age, 2016.

[65] Xiaofeng Liu, Chaehwa Yoo, Fangxu Xing, Hyejin Oh, Georges El Fakhri, Je-Won Kang, Jonghye Woo, et al. Deep unsupervised domain adaptation: A review of recent advances and perspectives. *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.

[66] Yang Liu, Sheng Shen, and Mirella Lapata. Noisy self-knowledge distillation for text summarization. *arXiv preprint arXiv:2009.07032*, 2020.

[67] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

[68] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15 (12):1053–1058, 2018.

[69] MI Love, W Huber, and S Anders. Moderated estimation of fold changes and dispersion for rna-seq data with deseq2. vol. 15. *Genome Biol*, 2014.

[70] Malte D Luecken, Maren Büttner, Kridsadakorn Chaichoompu, Anna Danese, Marta Interlandi, Michaela F Müller, Daniel C Strobl, Luke Zappia, Martin Dugas, Maria Colomé-Tatché, et al. Benchmarking atlas-level data integration in single-cell genomics. *Nature methods*, 19(1):41–50, 2022.

[71] Chongyuan Luo, Christopher L Keown, Laurie Kurihara, Jingtian Zhou, Yupeng He, Junhao Li, Rosa Castanon, Jacinta Lucero, Joseph R Nery, Justin P

Sandoval, et al. Single-cell methylomes identify neuronal subtypes and regulatory elements in mammalian cortex. *Science*, 357(6351):600–604, 2017.

[72] Feiyang Ma and Matteo Pellegrini. Actinn: automated identification of cell types in single cell rna sequencing. *Bioinformatics*, 36(2):533–538, 2020.

[73] Wenjing Ma, Kenong Su, and Hao Wu. Evaluation of some aspects in supervised cell type identification for single-cell rna-seq: classifier, feature selection, and reference construction. *Genome biology*, 22(1):1–23, 2021.

[74] Wenjing Ma, Sumeet Sharma, Peng Jin, Shannon L Gourley, and Zhaohui S Qin. Lrcell: detecting the source of differential expression at the sub–cell-type level from bulk rna-seq data. *Briefings in Bioinformatics*, 23(3):bbac063, 2022.

[75] Wenjing Ma, Jiaying Lu, and Hao Wu. Cellcano: supervised cell type identification for single cell atac-seq data. *Nature Communications*, 14(1):1864, 2023.

[76] Evan Z Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R Bialas, Nolan Kamitaki, Emily M Martersteck, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, 2015.

[77] Sueli Marques, Amit Zeisel, Simone Codeluppi, David Van Bruggen, Ana Mendanha Falcão, Lin Xiao, Huiliang Li, Martin Häring, Hannah Hochgerner, Roman A Romanov, et al. Oligodendrocyte heterogeneity in the mouse juvenile and adult central nervous system. *Science*, 352(6291):1326–1329, 2016.

[78] Hansruedi Mathys, Jose Davila-Velderrain, Zhuyu Peng, Fan Gao, Shahin Mohammadi, Jennie Z Young, Madhvi Menon, Liang He, Fatema Abdurrob, Xueqiao Jiang, et al. Single-cell transcriptomic analysis of alzheimer's disease. *Nature*, 570(7761):332–337, 2019.

[79] Nicolas Merienne, Cécile Meunier, Anne Schneider, Jonathan Seguin, Satish S Nair, Anne B Rocher, Stéphanie Le Gras, Celine Keime, Richard Faull, Luc Pellerin, et al. Cell-type-specific gene expression profiling in adult mouse brain reveals normal and disease-state signatures. *Cell reports*, 26(9):2477–2493, 2019.

[80] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

[81] Mauro J Muraro, Gitanjali Dharmadhikari, Dominic Grün, Nathalie Groen, Tim Dielen, Erik Jansen, Leon Van Gurp, Marten A Engelse, Francoise Carlotti, Eelco Jp De Koning, et al. A single-cell transcriptome atlas of the human pancreas. *Cell systems*, 3(4):385–394, 2016.

[82] Corina Nagy, Malosree Maitra, Arnaud Tanti, Matthew Suderman, Jean-Francois Théroux, Maria Antonietta Davoli, Kelly Perlman, Volodymyr Yerko, Yu Chang Wang, Shreejoy J Tripathy, et al. Single-nucleus transcriptomics of the prefrontal cortex in major depressive disorder implicates oligodendrocyte precursor cells and excitatory neurons. *Nature neuroscience*, 23(6):771–781, 2020.

[83] Aaron M Newman, Chih Long Liu, Michael R Green, Andrew J Gentles, Weiguo Feng, Yue Xu, Chuong D Hoang, Maximilian Diehn, and Ash A Alizadeh. Robust enumeration of cell subsets from tissue expression profiles. *Nature methods*, 12(5):453–457, 2015.

[84] Aaron M Newman, Chloé B Steen, Chih Long Liu, Andrew J Gentles, Aadel A Chaudhuri, Florian Scherer, Michael S Khodadoust, Mohammad S Esfahani, Bogdan A Luca, David Steiner, et al. Determining cell type abundance and expression from bulk tissues with digital cytometry. *Nature biotechnology*, 37 (7):773–782, 2019.

[85] Siew-Min Ong, Karen Teng, Evan Newell, Hao Chen, Jinmiao Chen, Thomas Loy, Tsin-Wen Yeo, Katja Fink, and Siew-Cheng Wong. A novel, five-marker alternative to cd16–cd14 gating to identify the three human monocyte subsets. *Frontiers in immunology*, 10:1761, 2019.

[86] Giovanni Pasquini, Jesus Eduardo Rojo Arias, Patrick Schäfer, and Volker Busskamp. Automated methods for cell type annotation on scrna-seq data. *Computational and Structural Biotechnology Journal*, 19:961–969, 2021.

[87] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[88] V Hugh Perry, James AR Nicoll, and Clive Holmes. Microglia in neurodegenerative disease. *Nature Reviews Neurology*, 6(4):193–201, 2010.

[89] Yixuan Qiu, Jiebiao Wang, Jing Lei, and Kathryn Roeder. Identification of cell-type-specific marker genes from co-expression patterns in tissue samples. *Bioinformatics*, 37(19):3228–3234, 2021.

[90] Matthew E Ritchie, Belinda Phipson, DI Wu, Yifang Hu, Charity W Law, Wei Shi, and Gordon K Smyth. limma powers differential expression analyses for rna-sequencing and microarray studies. *Nucleic acids research*, 43(7):e47–e47, 2015.

[91] Devin Rocks, Ivana Jaric, Lydia Tesfa, John M Greally, Masako Suzuki, and Marija Kundakovic. Cell type-specific chromatin accessibility analysis in the mouse and human brain. *Epigenetics*, 17(2):202–219, 2022.

[92] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[93] W Brad Ruzicka, Shahin Mohammadi, Jose Davila-Velderrain, Sivan Subburaju, Daniel Reed Tso, Makayla Hourihan, and Manolis Kellis. Single-cell dissection of schizophrenia reveals neurodevelopmental-synaptic axis and transcriptional resilience. *MedRxiv*, 2020.

[94] Maureen A Sartor, George D Leikauf, and Mario Medvedovic. Lrpath: a logistic regression approach for identifying enriched biological groups in gene expression data. *Bioinformatics*, 25(2):211–217, 2009.

[95] Ansuman T Satpathy, Jeffrey M Granja, Kathryn E Yost, Yanyan Qi, Francesca Meschi, Geoffrey P McDermott, Brett N Olsen, Maxwell R Mumbach, Sarah E Pierce, M Ryan Corces, et al. Massively parallel single-cell chromatin landscapes of human immune cell development and intratumoral t cell exhaustion. *Nature biotechnology*, 37(8):925–936, 2019.

[96] Arpiar Saunders, Evan Z Macosko, Alec Wysoker, Melissa Goldman, Fenna M Krienen, Heather de Rivera, Elizabeth Bien, Matthew Baum, Laura Bortolin, Shuyu Wang, et al. Molecular diversity and specializations among the cells of the adult mouse brain. *Cell*, 174(4):1015–1030, 2018.

[97] Åsa Segerstolpe, Athanasia Palasantza, Pernilla Eliasson, Eva-Marie Andersson, Anne-Christine Andréasson, Xiaoyan Sun, Simone Picelli, Alan Sabirsh, Maryam Clausen, Magnus K Bjursell, et al. Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. *Cell metabolism*, 24 (4):593–607, 2016.

[98] Qianqian Song, Jing Su, and Wei Zhang. scgcn is a graph convolutional networks algorithm for knowledge transfer in single cell omics. *Nature communications*, 12(1):3826, 2021.

[99] Marlon Stoeckius, Christoph Hafemeister, William Stephenson, Brian Houck-

Loomis, Pratip K Chattopadhyay, Harold Swerdlow, Rahul Satija, and Peter Smibert. Simultaneous epitope and transcriptome measurement in single cells. *Nature methods*, 14(9):865–868, 2017.

[100] Kelly Street, Davide Risso, Russell B Fletcher, Diya Das, John Ngai, Nir Yosef, Elizabeth Purdom, and Sandrine Dudoit. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC genomics*, 19(1):1–16, 2018.

[101] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *Cell*, 177(7): 1888–1902, 2019.

[102] Tim Stuart, Avi Srivastava, Shaista Madad, Caleb A Lareau, and Rahul Satija. Single-cell chromatin state analysis with signac. *Nature methods*, 18(11):1333–1341, 2021.

[103] Kenong Su, Tianwei Yu, and Hao Wu. Accurate feature selection improves single-cell rna-seq cell clustering. *Briefings in bioinformatics*, 22(5):bbab034, 2021.

[104] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550, 2005.

[105] Tianyi Sun, Dongyuan Song, Wei Vivian Li, and Jingyi Jessica Li. scdesign2: a transparent simulator that generates high-fidelity single-cell gene expression count data with gene correlations captured. *Genome biology*, 22(1):1–37, 2021.

[106] Xiaobo Sun, Xiaochu Lin, Ziyi Li, and Hao Wu. A comprehensive comparison of supervised and unsupervised methods for cell type identification in single-cell rna-seq. *Briefings in bioinformatics*, 23(2):bbab567, 2022.

[107] Vivek Swarup, Flora I Hinz, Jessica E Rexach, Ken-ichi Noguchi, Hiroyoshi Toyoshiba, Akira Oda, Keisuke Hirai, Arjun Sarkar, Nicholas T Seyfried, Chialin Cheng, et al. Identification of evolutionarily conserved gene networks mediating neurodegenerative dementia. *Nature medicine*, 25(1):152–164, 2019.

[108] Yuqi Tan and Patrick Cahan. Singlecellnet: a computational tool to classify single cell rna-seq data across platforms and across species. *Cell systems*, 9(2): 207–213, 2019.

[109] Hoa Thi Nhu Tran, Kok Siong Ang, Marion Chevrier, Xiaomeng Zhang, Nicole Yee Shin Lee, Michelle Goh, and Jinmiao Chen. A benchmark of batch-effect correction methods for single-cell rna sequencing data. *Genome biology*, 21(1): 1–32, 2020.

[110] Maria Tsompana and Michael J Buck. Chromatin accessibility: a window into the genome. *Epigenetics & chromatin*, 7(1):1–16, 2014.

[111] Daphne Tsoucas, Rui Dong, Haide Chen, Qian Zhu, Guoji Guo, and Guo-Cheng Yuan. Accurate estimation of cell-type composition from gene expression data. *Nature communications*, 10(1):1–9, 2019.

[112] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[113] David Van Dijk, Roshan Sharma, Juozas Nainys, Kristina Yim, Pooja Kathail, Ambrose J Carr, Cassandra Burdziak, Kevin R Moon, Christine L Chaffer, Diwakar Pattabiraman, et al. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729, 2018.

[114] Xuran Wang, Jihwan Park, Katalin Susztak, Nancy R Zhang, and Mingyao Li. Bulk tissue cell type deconvolution with multi-subject single-cell expression reference. *Nature communications*, 10(1):1–9, 2019.

[115] Joshua D Welch, Velina Kozareva, Ashley Ferreira, Charles Vanderburg, Carly Martin, and Evan Z Macosko. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*, 177(7):1873–1887, 2019.

[116] Yurong Xin, Jinrang Kim, Haruka Okamoto, Min Ni, Yi Wei, Christina Adler, Andrew J Murphy, George D Yancopoulos, Calvin Lin, and Jesper Gromada. Rna sequencing of single human islet cells reveals type 2 diabetes genes. *Cell metabolism*, 24(4):608–615, 2016.

[117] Zizhen Yao, Cindy TJ van Velthoven, Thuc Nghi Nguyen, Jeff Goldy, Adriana E Sedeno-Cortes, Fahimeh Baftizadeh, Darren Bertagnolli, Tamara Casper, Megan Chiang, Kirsten Crichton, et al. A taxonomy of transcriptomic cell types across the isocortex and hippocampal formation. *Cell*, 184(12):3222–3241, 2021.

[118] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3911, 2020.

[119] Mahdi Zamanighomi, Zhixiang Lin, Timothy Daley, Xi Chen, Zhana Duren, Alicia Schep, William J Greenleaf, and Wing Hung Wong. Unsupervised clustering and epigenetic classification of single cells. *Nature communications*, 9(1): 1–8, 2018.

[120] Allen W Zhang, Ciara O'Flanagan, Elizabeth A Chavez, Jamie LP Lim, Nicholas Ceglia, Andrew McPherson, Matt Wiens, Pascale Walters, Tim Chan,

Brittany Hewitson, et al. Probabilistic cell-type assignment of single-cell rna-seq for tumor microenvironment profiling. *Nature methods*, 16(10):1007–1015, 2019.

[121] Xinxin Zhang, Yujia Lan, Jinyuan Xu, Fei Quan, Erjie Zhao, Chunyu Deng, Tao Luo, Liwen Xu, Gaoming Liao, Min Yan, et al. Cellmarker: a manually curated resource of cell markers in human and mouse. *Nucleic acids research*, 47(D1):D721–D728, 2019.

[122] Jia Zhao, Gefei Wang, Jingsi Ming, Zhixiang Lin, Yang Wang, Angela Ruohao Wu, and Can Yang. Adversarial domain translation networks for integrating large-scale atlas-level single-cell datasets. *Nature Computational Science*, 2(5): 317–330, 2022.

[123] Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1):1–12, 2017.

[124] Yi Zhong, Ying-Wooi Wan, Kaifang Pang, Lionel ML Chow, and Zhandong Liu. Digital sorting of complex tissues for cell type-specific gene expression profiles. *BMC bioinformatics*, 14(1):1–10, 2013.