UNIVERSITY OF CALIFORNIA,
IRVINE

Deep Representation Learning for Single-cell Sequencing Data Analysis

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Mathematical, Computational and Systems Biology

by

Yingxin Cao

Dissertation Committee:
Professor Xiaohui Xie, Chair
Professor Qing Nie
Associate Professor Trina Norden-Krichmar

2023

DEDICATION

To my parents, Yang & Yueju,
To my better half, Yiqun.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisors, Professor Xiaohui Xie and Professor Qing Nie, for their support, guidance, and encouragement throughout the course of my research. Your profound insights, patience, and dedication have immensely contributed to the development and completion of this dissertation.

I would also like to extend my heartfelt thanks to my collaborators and advancement committee members, Professor Jing Zhang, Professor Trina Norden-Krichmar, and Professor Ken Cho, for their invaluable insights, innovative ideas on various projects. It was an honor and pleasure working with you.

A special mention goes to all the previous and current members of Xie Lab. Sharing the lab with you has been both an educational and enjoyable experience. Your support, and constructive feedback have been essential in shaping my journey. In particular, I wish to thank Dr. Laiyi Fu and Muhammed Hasan Çelik, for numerous enriching discussions that have been pivotal to the progress of my work.

In addition to the academic and professional support I have received, I wish to express gratitude to previous and current MCSB administrators, Karen Martin, Cely Dean, Naomi Carreon, Tina Rimal, Austin Berryman, Emi Embler, and CMCF administrator Clare Cheng. Thanks for being patient and responsive to all the questions and requests during my entire Ph.D training.

Lastly, to anyone who contributed, either directly or indirectly, to this thesis and has not been mentioned, please know that your assistance and influence have not gone unnoticed. Thank you for being a part of this journey.

I thank Oxford Academic for letting me use the materials in Chapter 2 and 3 of my dissertation, which was originally published in *Bioinformatics* and *Nucleic Acids Research*. Financial support was provided by National Science Foundation IIS-1715017, DMS-1763272, and Simons Foundation 594598.

VITA

Yingxin Cao

EDUCATION

Doctor of Philosophy in
Mathematical, Computational and Systems Biology **2023**
University of California, Irvine *Irvine, California*

Master of Science in Bioengineering **2018**
University of Washington *Seattle, Washington*

Bachelor of Engineering in Biological Engineering **2016**
Xiamen University *Xiamen, Fujian, China*

PUBLICATIONS

Y. Cao, L. Fu, J. Wu, Q. Peng, Q. Nie, J. Zhang, and X. Xie. ”Integrated analysis of multimodal single-cell data with structural similarity.” *Nucleic Acids Research*, (2022).

L. Fu, Y. Cao, J. Wu, Q. Peng, Q. Nie, and X. Xie. ”Ufold: fast and accurate rna secondary structure prediction with deep learning.” *Nucleic acids research*, (2022).

Y. Cao, L. Fu, J. Wu, Q. Peng, Q. Nie, J. Zhang, and X. Xie. ”Sailer: scalable and accurate invariant representation learning for single-cell atac-seq processing and integration.” *Bioinformatics*, (2021).

Y. S. Vang, Y. Cao, P. D. Chang, D. S. Chow, A. U. Brandt, F. Paul, M. Scheel, and X. Xie. ”Synergynet: a fusion framework for multiple sclerosis brain mri segmentation with local refinement.” In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, (2020).

Y. S. Vang, Y. Cao, and X. Xie. ”A combined deep learning-gradient boosting machine framework for fluid intelligence prediction.” In *Adolescent Brain Cognitive Development Neurocognitive Prediction: First Challenge, ABCD-NP 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 1*, (2019).

ABSTRACT OF THE DISSERTATION

Deep Representation Learning for Single-cell Sequencing Data Analysis

By

Yingxin Cao

Doctor of Philosophy in Mathematical, Computational and Systems Biology

University of California, Irvine, 2023

Professor Xiaohui Xie, Chair

Single-cell sequencing assays nowadays provide comprehensive genomics readouts at single cell resolution. These measurements provide unprecedented opportunities for researchers to study cell heterogeneity and elucidate transcriptional regulatory mechanisms. However, computational modeling of single-cell sequencing data is challenging due to its high dimension, extreme sparsity, complex dependencies and high sensitivity to noises from various sources.

In this thesis, we present our works of designing representation learning frameworks to deal with various noises and effectively learn meaningful representations of cells and genes from large-scale single-cell sequencing datasets. In the first part, we present our design using deep generative models to learn confounding-free representations of cells through invariant representation learning on scATAC-seq data. By eliminating the variations of confounding factors in the latent space through mutual information minimization, our method produces biologically more meaningful representations of cells, which brings in significant benefits in downstream analyses. As a follow-up work, we present our strategy to extend this framework to a multi-modal setting. Instead of performing hard alignment by projecting both modalities to a shared latent space, our method encourages the local structures of two modalities measured by pairwise similarities to be similar. This strategy is more robust against overfitting of noises, and facilitates various downstream analysis such as clustering, imputation, and marker gene detection. In the second line of work, we present our design of foundation models to learn meaningful semantic representation of genes from broad scRNA-seq datasets. We show that pretraining foundation models on large-scale single cell datasets enable the models to learn meaningful features of genes that are transferable to many other downstream tasks. The pretrained model can also be adapted for imputation tasks with great performance.

Chapter 1

Introduction

1.1 Background

Single-cell Sequencing

The human body is made up of 37 trillion cells, each with their own structure and function. The structural and functional characteristics of cells are determined by the proteins they contain. No two cells in the body contain the exact same amount of each protein. The functions served by these cells are broad and diverse, ranging from red blood cells delivering oxygen, to immune cells defending our body and neurons generating perception and cognition. The central dogma of molecular biology states that the instructions for producing all the cells are written in the DNA. One of the greatest puzzles in biology is how the one-dimensional string of just four letters gives rise to the complexity and diversity epitomized by the 37 trillion cells in our body. To solve this puzzle, we need efficient experimental data to characterize the structures and functions of cells systemically and comprehensively, as well as innovative computational methods to figure out the mappings between genomics elements that eventually generate cell functions.

Recent advances in single cell sequencing technologies offer genome-wide measurements of genetic information from individual cells and have produced a number of large-scale reference data to characterize the complexity and diversity of human cells [84, 89, 14, 51]. Specifically, single-cell RNA sequencing (scRNA-seq) provides quantitative measures of the RNA expressions of all genes in single cells, up to millions of cells in one experiment. Single-cell ATAC-seq (sc-ATAC-seq) provides a comprehensive measurements of the chromatin accessibility of the entire genome in individual cells, describing cell functions and states in terms of the epigenetic landscape of the chromatin. In addition to RNAs and chromatin accessibility, single cell technologies for other genomic modalities such as proteins and DNA methylations, have also been developed. These developments have lead to the generation of hundreds of large-scale single cell datasets, providing unprecedented views on the complexity and diversity of cells across multiple genomic modalities, revealing new cell types, and new definition of cell states and conditions. However, these large scale measurements also come with limited coverage, shallow read depth per cell, and batch effect, which makes data analysis challenging, especially for integrative analysis across multiple datasets [60]. To effectively utilize the information from these datasets, we need computational methods that could be efficiently scaled to large-scale high dimensional dataset, and able to denoise the data to produce biologically meaningful outputs.

Representation Learning

Representation learning aims to learn a suitable transformation of raw data to a space that is easier for downstream machine learning algorithms to understand and process. The transformed data is often called an embedding of original data. A good embedding captures the inherent structure or meaningful attributes of the data, and is less noisy and usually more interpretable compared with original measurements.

Many methods for single cell sequencing data analysis are developed based on the manifold assumption [75, 60], which assumes that even though measurements of single cell data is originally from extremely high dimension, since expression of different genes are correlated according to certain rules, intrinsic cell distributions is from a lower-dimensional manifold. Standard single cell data analysis pipelines usually perform clustering on this manifold to assign a label to each single cell in an unsupervised mannar [68]. The clustering is usually the start point of all the other downstream analysis, thus, many efforts are made to perform better dimensional reduction on raw single cell data, to transform the raw measurements of gene expressions to a interpretable representation of cells that researchers could used to determine cell states, compute cell trajectories, study differentially expressed genes, and identify targets or key elements that lead to scientific discoveries.

Deep Generative Models

Variational Autoencoder (VAE) [55] is a class of deep generative models that estimate the likelihood of data through Bayesian variational inference. The Bayesian approach to describe data distribution is usually through a latent random variable z as intermediate, and then data could be generate by sampling from likelihood distribution $p_\theta(x|z)$, with θ as parameters of the model. $p(z)$ is the prior distribution of the data. VAE assumes a standard normal prior, and the posterior distribution $q_\phi(z|x)$ is approximated by a encoder model.

This specification is suitable for representation learning, as one can specify the latent variable with lower dimension to represent the states of the cell, and jointly optimize the inference model $q_\phi(z|x)$ parameterized by encoder network and the generative model $p_\theta(x|z)$ parameterized by decoder network on single cell datasets. With GPU acceleration, and batch-based gradient decent optimization, these models can be computed very efficiently and scaled to large sample sizes. Also, this specification allow us to impose additional desired properties on the latent variable z through auxiliary objective functions, which makes it a flexible framework with probabilistic interpretations.

Foundation Models

A foundation model is a large deep neural net model trained on a vast quantity of data at scale that can be adapted to a wide range of downstream applications [7]. Foundation models are behind recent transformations in how AI systems are built. Examples include large-scale language models such as BERT [27], GPT [10], and PaLM [23], vision models such as ResNet [44], Inception [99], and MAE [43], and multimodal models such as CLIP [82] and DALL-E [83]. The rise of foundation models has significantly lowered the barrier of building AI models for downstream applications, with users focusing on adapting and transferring knowledge from these models to new applications, instead of building new models from scratch.

Most of the recent foundation models are built on top of the transformer architecture [109]. Inputs to the transformer model are a sequence of word tokens from a differentiable embedding module (NLP), or ordered patch tokens generated from a projection neural network (CV). The tokens are then modeled through projections and multi-head attentions to generate high-level features used for predictions or other downstream tasks. The attention mechanism allows transformer to model interactions between tokens at any locations of the sequence with constant time complexity regarding to the order of the sequence. This allows transformer models to be efficiently applied to large-scale training on broad datasets, learning complex and meaningful features that are transferable for multiple tasks.

Pretraining of foundation models requires a pretext that could be conducted in an self-supervised manner. Masked prediction is a common pretext for pretrain foundation models both in NLP [27] and CV [43]. By masking out a significant amount of input data, the model needs to learn fundamental rules in order to reconstruct the original data. In this way, these pretexts allow the model to learn meaningful features without any labeled data. And by increase the scale of the data as well as the parameter size of the model, researchers observed continued performance increase on downstream tasks [10, 82, 83]. After pretraining, the embedding module which contains tokens corresponding to individual words are optimized to represent knowledge of the large scale datasets. Simple finetuning of these embeddings could achieve great results on many downstream tasks. By representing genomics elements as differentiable tokens [1], researchers have built foundation models for representation learning on many types of largely available public datasets [87, 4, 49]. This provides a novel way for representation learning, that could significantly reduce the computational resources needed to use state-of-the-art AI models and benefit studies with only limited sample sizes.

1.2 Overview of the Dissertation

The main context of this dissertation contains three chapters. Chapter 2 and 3 focus on learning representation of cells with variants of deep generative model frameworks. Chapter 4 focuses on learning representation of genes.

In chapter 2, we discuss our work on the SAILER project [16]. The aim of the project is to use invariant representation learning to disentangle known confounding factors from the latent representation of the cells. We achieve this through an auxiliary objective on mutual information and achieved better results on clustering and imputation.

Chapter 3 discusses SAILERX [17], which is a follow-up on the SAILER work. In this project, we focuses on fusing information from multiple modalities in a synergistic way, which would eventually generate a better embedding than working with single modality. We found that aligning different modalities through a similarity metric offers better results than performing hard alignment. And we also extend this framework to integrate multi-modal datasets with single-modal ones.

In chapter 4, we discuss our work on learning representation of genes through pretraining foundation models on large scale datasets. We explain details of our framework that designed to perform pretraining on scRNA-seq data, and demonstrate the utility of pretrained model on multiple downstream tasks.

Chapter 2

Disentangle Known Confounding Factors from Latent Representation of Cells through Invariant Representation Learning

Single-cell sequencing assay for transposase-accessible chromatin (scATAC-seq) provides new opportunities to dissect epigenomic heterogeneity and elucidate transcriptional regulatory mechanisms. However, computational modelling of scATAC-seq data is challenging due to its high dimension, extreme sparsity, complex dependencies, and high sensitivity to confounding factors from various sources.

Here we propose a new deep generative model framework, named SAILER, for analysing scATAC-seq data. SAILER aims to learn a low-dimensional nonlinear latent representation of each cell that defines its intrinsic chromatin state, invariant to extrinsic confounding factors like read depth and batch effects. SAILER adopts the conventional encoder-decoder framework to learn the latent representation but imposes additional constraints to ensure the independence of the learned representations from the confounding factors. Experimental results on both simulated and real scATAC-seq datasets demonstrate that SAILER learns better and biologically more meaningful representations of cells than other methods. Its noise-free cell embeddings bring in significant benefits in downstream analyses: Clustering and imputation based on SAILER result in 6.9% and 18.5% improvements over existing methods, respectively. Moreover, because no matrix factorization is involved, SAILER can easily scale to process millions of cells. We implemented SAILER into a software package, freely available to all for large-scale scATAC-seq data analysis.

2.1 Introduction

Accessible chromatin regions host a network of complex interplays among numerous cis-regulatory elements (CREs, such as enhancers and promoters), transcription factors (TFs), cofactors, and chromatin remodelers in the three-dimensional genome for precise spatiotemporal gene expression control [56, 106, 8]. Assay for transposase-accessible chromatin using sequencing (ATAC-seq) is an efficient method to probe accessible DNA regions in the genome, by tagging them with sequencing adapters using the Tn5 transposase [11]. More recently, researchers have developed single-cell ATAC-seq (scATAC-seq) technology to massively probe accessible chromatin regions in individual cells [12, 25, 21, 91]. These methods make it possible to comprehensively dissect the epigenetic heterogeneity across diverse cell states at an unprecedented resolution. Due to its easy protocols and high-throughput capacities, many labs and big consortia (e.g., the Human Cell Atlas, Human BioMolecular Atlas Program) have employed scATAC-seq for single-cell epigenetic profiling [84, 62]. Furthermore, the scientific community and funding agencies have initiated essential data-sharing policies for expedited translational research. Thus, there is an urgent and essential need to develop robust, accurate, and scalable computational methods for scATAC-seq data analysis and integration at a large scale.

Unfortunately, computational modeling of scATAC-seq data has faced several challenges. First, scATAC-seq data tends to have very low coverage, usually with a few thousand distinct reads representing hundreds of thousands to even millions of accessible regions. Second, scATAC-seq contains a high degree of dependencies because numerous cell-type-specific CREs in accessible chromatin regions work in concert to jointly decide cell fate. Lastly, scATAC-seq analysis is highly sensitive to numerous confounding factors arising within and across samples (e.g., read depth variation and dataset-specific conditions).

Researchers have developed many computational approaches to tackle high-dimensional and sparse scATAC-seq data [92, 33, 9, 115, 35], ut each has its limitations. For instance, ChromVAR ignores the impacts of individual peaks and only groups cells by the TF motif enrichment scores from all peaks, resulting in non-optimal clustering performance [92]. SnapATAC uses Jaccard distance to calculate cell-to-cell similarities for dimension reduction with a hidden assumption that peaks are independent of each other and contribute equally to the similarity measure, which is incorrect in most cases. More recently, researchers developed the latent semantic index (LSI) for learning the lower-dimensional cell representations [81, 38, 98]. Despite their scalability, such linear techniques may not fully capture the complex dependencies of peaks. Moreover, these approaches correct for read depth effects by removing components that highly correlate with the read depth, which is heuristic and may lose the true cell-state-related information. Other nonlinear approaches, such as cisTopic and SCALE, were then developed to learn better cell representations [9, 115]. However, these methods assume constant read depths across different cells and ignore potential batch effects from multiple samples, which compromises model performance in real applications.

Here, we aimed to overcome the limitations of existing methods by designing an invariant representation learning scheme with a straightforward intuition – the true epigenetic variations from a specific cell state should remain the same across cells and samples, while variations arising from confounding factors may change substantially, even for cells within similar biological groups. In other words, we can dissect the scATAC-seq cell-to-cell variations into an invariant component representing its hidden cell states and a varying component due to non-biological factors, such as the number of fragments in a cell and batch effects in the multi-sample analyses (Figure 2.1). To this end, we developed a scalable and accurate invariant representation learning scheme (SAILER) via a deep generative model to learn a robust cell representation $\mathbf{z}$ that is only related to intrinsic cell states but is invariant to changes in the confounding factor $\mathbf{c}$ (Figure 2.1). Specifically, we remove the variations related to confounding factors from the learned latent representation by minimizing their mutual information $I(\mathbf{z}, \mathbf{c})$. Compared with previous methods, SAILER has three major advantages: i) it is easily scalable to millions of cells in large-scale analyses via accelerated computation on graphic processing units (GPUs); ii) it captures the nonlinear dependencies among peaks via the expressiveness of deep generative modeling and robustly removes confounding factors from various sources, both within and across samples, to faithfully extract biologically relevant information; iii) it provides a unified strategy for scATAC-seq denoising, clustering, and imputation.

We implemented SAILER into a Python package that is freely available to the community. To prove its effectiveness, we first benchmarked the clustering performance of SAILER with state-of-the-art methods. We utilized three simulated scATAC-seq datasets with ground-truth labels, representing different application scenarios with single- and multi-sample inputs. SAILER significantly outperformed the existing methods, providing improved cell clustering results and successfully identifying rare cell types. We also applied SAILER on real atlas-level and multi-sample scATAC-seq datasets and showed that it could efficiently learn better biologically relevant cell latent representations, which will facilitate various downstream analyses such as cell clustering and imputations.

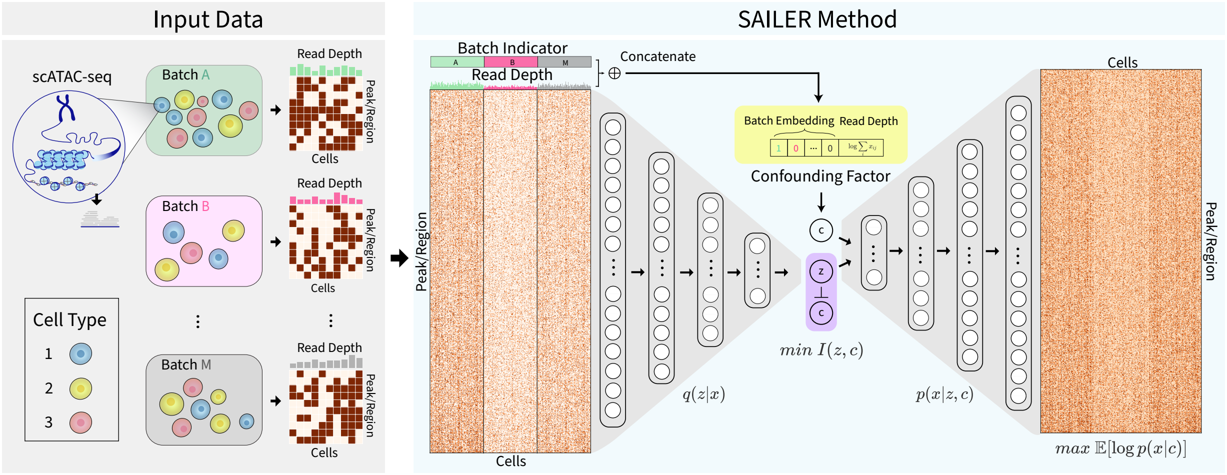


Figure 2.1: The overall design of the SAILER method. SAILER takes scATAC-seq data from multiple batches as input. Raw data is pushed through the encoder network to obtain a latent representation. Confounding factors for each single cell are concatenated and fed to the decoder along with the latent representation. Batch information is indicated by a one-hot embedding, and read depth is subject to log transform and standard normalization. To learn a latent representation invariant to changes in confounding factors, mutual information between the latent variables and confounding factors are minimized during training.

2.2 Materials and Methods

In this section, we provide the mathematical details on our SAILER model and describe methods for benchmarking with existing methods using both simulated and real datasets.

2.2.1 Effective invariant representation learning via a deep generative model

Let $\mathbf{x} \in \{0, 1\}^n$ (with n peaks or bins) denote the genome-wide chromatin profile of a cell, with x_i indicating the presence or absence of a peak in bin i. $\mathbf{x}$ depends on both the intrinsic properties of the cell and experimental confounding factors. Our goal is to derive a latent representation of $\mathbf{x}$ (also called embedding) for each cell that reflects only its intrinsic properties. Let $\mathbf{z} \in R^d$ be such a latent representation. Suppose $\mathbf{c}$ is the confounding variable that has statistical dependence on $\mathbf{x}$, and is observable together with $\mathbf{x}$. We denote $q_\theta\left(\mathbf{z}|\mathbf{x}\right)$ as the encoder probability, $p_\phi\left(\mathbf{x}|\mathbf{z},\mathbf{c}\right)$ as the decoder probability. The decoder part of our model aims to model the conditional probability of $\mathbf{x}$ on $\mathbf{c}$ through a latent variable $\mathbf{z}$,

$$p(\mathbf{x}|\mathbf{c}) = \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[p_\phi(\mathbf{x}|\mathbf{z},\mathbf{c})] \tag{2.1}$$

where $p\left(\mathbf{z}\right)$ is the prior distribution for a generative model set to be a (factorized Gaussian) in our case. $q\left(\mathbf{x},\mathbf{c}\right)$ is the empirical distribution of the data point and confounding variable, ϕ denote the parameters of the decoder network.

Following the variational autoencoder (VAE) model [55], we performed parameter inference by maximizing an evidence lower bound of the log likelihood, corresponding to minimizing the following loss function,

$$L_{\text{VAE}} = \mathbb{E}_{\mathbf{x},\mathbf{c}\sim q(\mathbf{x},\mathbf{c})}\left[-\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z},\mathbf{c})] + D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}))\right] \tag{2.2}$$

where $q_\theta\left(z|x\right)$ is the posterior distribution modeled with a neural net with parameters θ.

The distribution of the latent representation z induced by empirical data distribution and the posterior probability $q_\theta\left(z|x\right)$ potentially can depend on c, as c is involved in the data generation process. To derive a latent representation z independent of the confounding variable c, we added an additional term to the loss function to minimize the mutual information between the two variables [76],

$$L_{\text{VAE}} + \lambda I(\mathbf{z},\mathbf{c}) \tag{2.3}$$

where $I\left(z,c\right)$ is the mutual information between latent representation z and c, with their joint distribution represented by $q_\theta(z, x, c) = q(x, c)\, q_\theta(z|x)$. Based on the properties of mutual information and variational inequality, $I(z, c)$ is upper bounded by

$$I(\mathbf{z}, \mathbf{c}) \leq \mathbb{E}_{\mathbf{x}}[D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||q_\phi(\mathbf{z}))] - H(\mathbf{x}|\mathbf{c}) - \mathbb{E}_{\mathbf{x},\mathbf{c},\mathbf{z}\sim q}[\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{c})] \tag{2.4}$$

where the conditional entropy $H(x|c)$ is a constant and can be removed from the loss function.

The final loss function we aimed to minimize is

$$L(\phi, \theta) = \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\left[D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})) + \lambda D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel q_\phi(\mathbf{z}))\right] \tag{2.5}$$

$$-(1+\lambda)\mathbb{E}_{\mathbf{x},\mathbf{c}\sim q(\mathbf{x},\mathbf{c})}\left[\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{c})]\right] \tag{2.6}$$

Here is the KL-divergence between the encoder $q_\theta(z|x)$ and prior $p(z)$. is the reconstruction loss. is the KL-divergence between $q\ _\theta(z|x)$ and empirical marginal distribution $q\ _\theta(z)$. Because $q\ _\theta(z)$ depends on the distribution of both $\mathbf{x}$ and $\mathbf{c}$, minimizing the above KL-divergence will reduce the effect of $\mathbf{c}$ on $\mathbf{z}$. In the implementation, this extra term is approximated by pairwise KL-divergences between all data points in a training batch, $\sum_{\mathbf{x}} \sum_{\mathbf{x}'} D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel q_\phi(\mathbf{z}|\mathbf{x}'))$. Since latent variable z is parameterized by an isotropic Gaussian, the pairwise KL has a nice analytical form, and can be efficiently computed with matrix algebra. More detailed derivations can be found in supplementary notes section 2.

2.2.2 Model architecture and training

Considering the close to binary nature of scATAC-seq data, we use binomial likelihood to parameterize the reconstruction loss. To tackle the extreme sparsity issue, we add a positive weight ω to non-zero entries of binary cross-entropy loss $l = \omega \cdot \mathbf{x} \cdot \log \hat{\mathbf{x}} + (1 - \mathbf{x}) \cdot \log(1 - \hat{\mathbf{x}})$ with ω determined by the empirical 0/1 ratio of the input data.

The encoder and decoder are parameterized by two symmetric fully connected feedforward neural networks (with 1000-100-10 units). A sigmoid activation is used for the final output layer. For confounding factors, we use one-hot batch embedding and normalized log-transformed sequencing depth for each cell. During training, input data is pushed through the encoder network to generate the latent variable. Confounding factors are then concatenated together with latent variables and fed into the conditional decoder for reconstruction. As suggested in [34], when training our model, we adopt a deterministic warmup and cyclical annealing schedule to tackle the KL vanishing problem. Adam optimizer [54] with weight decay 5e-4 and minibatch training are used to optimize the model. The model is built with PyTorch library [79]. Hyperparameters of the model is chosen according to the log-likelihood of the validation set. λ is set to be 1 in our study. In practice, value of λ can also be selected based on empirically checking the values of the $I(z, c)$ and λ. The optimal value is determined to be the point where increasing λ does not lead to significant drop in the MI. In supplementary notes section 1, Table 2.1 2.2 2.3, and Figure 2.2 we show that performance our method is robust against hyperparameter choices.

Table 2.1: Evaluation Results under different λs.

λ	0	0.01	0.1	1	2	10	50
$I(\mathbf{z}, \mathbf{c})$	0.071	0.053	0.045	0.040	0.043	0.046	0.040
ARI	0.539	0.546	0.560	0.575	0.546	0.562	0.605
NMI	0.773	0.774	0.778	0.799	0.772	0.779	0.780

Table 2.2: Evaluation results under different latent dimensions.

$dim(\mathbf{z})$	2	5	10	15	20
$I(\mathbf{z}, \mathbf{c})$	0.105	0.059	0.040	0.038	0.0400
ARI	0.482	0.555	0.575	0.556	0.571
NMI	0.735	0.763	0.799	0.774	0.770

Table 2.3: Evaluation results under different intermediate neuron numbers.

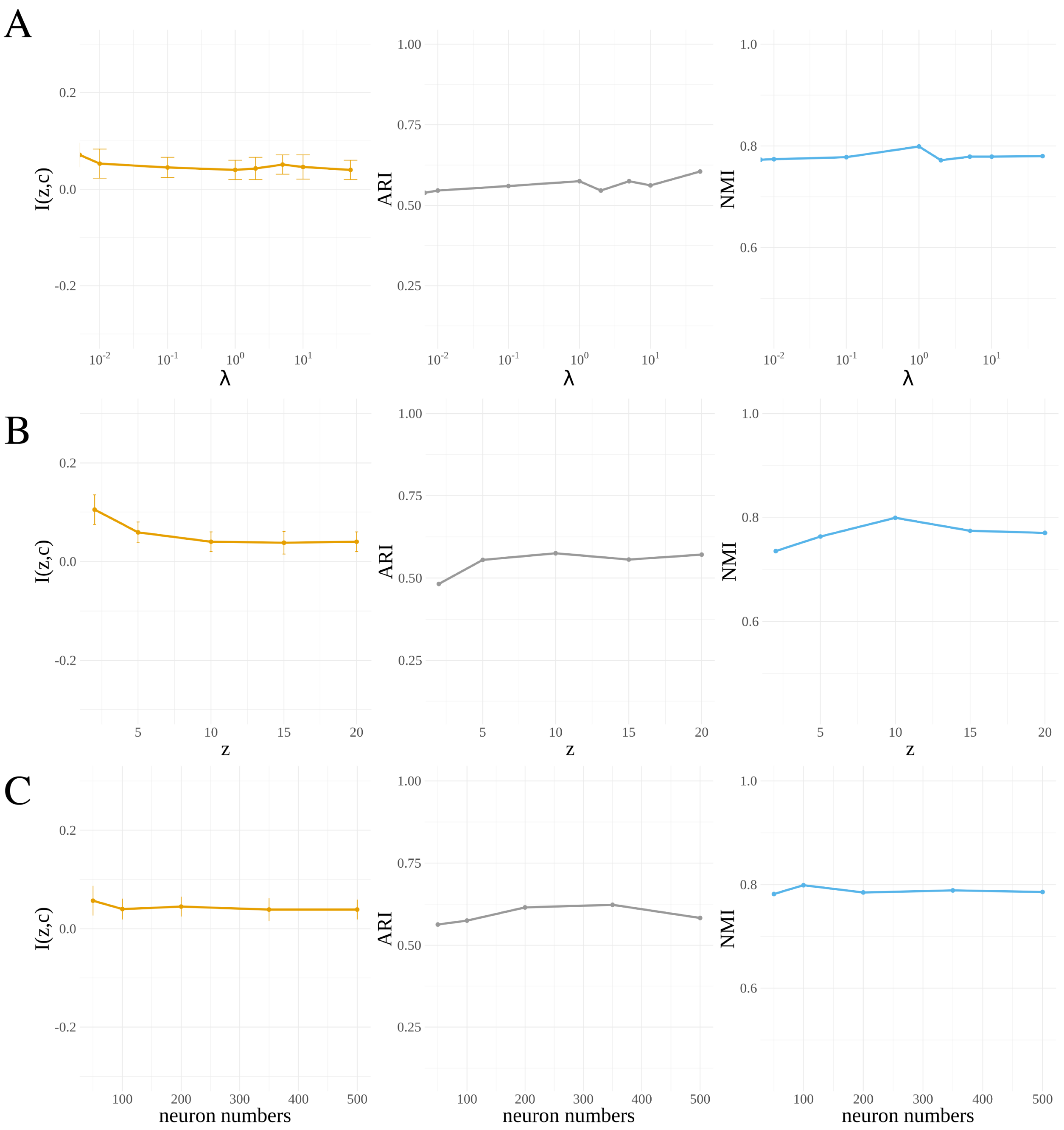


Figure 2.2: **Clustering Performance of $I(\mathbf{z}, \mathbf{c})$, ARI and NMI under different Hyperparameters.** (A) Performance variations with λ ranging from 0.01 to 100. (B) Performance variations with latent dimension ranging from 5 to 20. (C) Performance variations using number of neurons ranging from 100 to 500.

# of units	50	100	200	350	500
$I(\mathbf{z}, \mathbf{c})$	0.057	0.040	0.045	0.039	0.039
ARI	0.563	0.575	0.615	0.623	0.583
NMI	0.782	0.799	0.785	0.789	0.786

2.2.3 Dimension reduction and clustering

We project the raw high-dimensional sparse scATAC-seq data to a low-dimensional space that reflects the hidden cell states rather than noise in the sequencing experiment. Specifically, we used the raw scATAC-seq matrix x as the input to our SAILER encoder and extracted the mean of the invariant component z as the cell representation. We set the default dimension d for z to 10 in our analysis. We then acquired 2D visualizations by running t-distributed stochastic neighbor embedding (t-SNE) [107] or uniform manifold approximation and projection (UMAP) [72] on the latent mean. We further constructed a k-nearest neighbor (KNN) graph from the lower-dimensional representations, and then applied the Louvain algorithm [6] to assign cells to different clusters.

2.2.4 scATAC-seq imputation

We generated the imputation data via a reconstruction conditioned on the invariant representation $\mathbf{z}$ and fixed confounding factor $\mathbf{c}$. Specifically, we first pushed the raw data through the encoder network, and obtained the mean parameters for $\mathbf{z}$. Unlike the training process, where we calculated the depth of the raw data and loaded the one-hot embedding according to the real batch information, here we fixed the depth and batch indicator as the mean depth and the indicator of the batch with the highest data quality. Finally, we concatenated the fixed confounding values with the latent representation $\mathbf{z}$ and fed them into the conditional decoder to obtain the imputed data. As a result, we used only the invariant component $\mathbf{z}$ to reconstruct the chromatin landscape during the imputation process, while keeping the other confounding factors at a fixed level.

2.2.5 Performance benchmarking using multiple simulated datasets

We applied SAILER on three simulated scATAC-seq datasets with known cell type labels generated by SCAN-ATAC-Sim [22] to represent three major application scenarios. We used the peripheral blood mononuclear cell bulk ATAC-seq dataset provided on the SCAN-ATAC-Sim website using all default parameter settings. Each simulation includes three major parameters: ρ represents the signal-to-noise ratio (percentage of reads in the true peak regions); μ and σ denote the mean and standard deviation of the fragment count per cell, respectively.SCAN-ATAC-Sim randomly selects read counts for each cell from a log-normal distribution, and then samples reads from both peak and background regions accordingly. We first simulated a deeply sequenced scATAC-seq dataset (Sim1) with 5,000 fragments per cell (μ=5,000, σ=1.5, and ρ=0.4), representing a scenario in which we are looking for rare cell types. Specifically, we generated 10,000 cells from five cell types, with 100 cells from a rare cell type accounting for 1% of the total population. Then, we generated one shallowly sequenced sample with nine cell types, with μ=3,000, σ=1.5, and ρ=0.4 (Sim2). Lastly, we simulated a two-sample dataset with slightly mismatched cell types to represent scATAC-seq data integration applications with noticeable batch effects – one shallowly sequenced sample (μ=2,500) along with another deep-sequenced sample (μ=5,000) with different signal-to-noise ratios (ρ=0.4 and 0.5, respectively) (Sim3). In addition, we introduced one sample-specific rare cell type in Sim3 to mimic a situation in which rare cell types (e.g., tumor cells) may only exist in some samples. We benchmarked SAILER's clustering performance with the linear dimension reduction method LSI and another deep learning method, SCALE, on all three simulated datasets. Specifically, we projected the raw input matrix x to a ten-dimensional latent space, and further used UMAP to reduce the dimension to 2 for 2D visualization of the cell state landscape. We plotted colored labels according to the ground-truth cell type for visual inspection of clustering performance.

We also used the mutual information to quantify the impacts of confounding factors on the lower-dimensional representations learned by different methods. Specifically, we used a non-parametric mutual information estimation approach [58] to estimate the mutual information between the confounding factors and each dimension of the latent representation, and calculated their mean values for comparison.

2.2.6 Imputation performance on simulated datasets

We also benchmarked the imputation performance of SAILER against SCALE [115] and MAGIC [108] on the Sim3 dataset. SCALE is the only current method designated for imputing scATAC-seq data, and MAGIC, originally designed to impute scRNA-seq data, has been incorporated into many scATAC-seq computational pipelines [33, 38] for imputation purposes.

For SCALE, we directly used the binary imputation output generated by thresholding at mean values of each row and column. For MAGIC, we followed the standard pipeline by applying the recommended l1 normalization and square root transformation before imputing the data. Due to the extreme dimension, we used an approximate solver for efficiency. For SAILER, we performed imputation as described in previous chapter.

To evaluate the result quantitatively, we calculated the Dice similarity coefficient (DSC) of imputed data $\hat{x}$ generated by the three methods against the bulk ATAC-seq data x_{bulk} of the corresponding cell type used to generate the simulated data. We calculated the DSC of the raw input against the bulk data to provide a baseline.

$$DSC = \frac{2 \cdot \mathbf{x}_{bulk} \cdot \hat{\mathbf{x}}}{|\mathbf{x}_{bulk}| + |\hat{\mathbf{x}}|} = \frac{2TP}{2TP + FP + FN} \tag{2.7}$$

We also generated a 2D visualization to evaluate the landscape of the imputed data. We directly applied a randomized principal component analysis (PCA) [40] to the imputed data, and used UMAP to visualize the top ten principal components. We also provided the raw input as a baseline.

2.2.7 Performance benchmarking on the mouse atlas dataset

We then demonstrated the performance of our method on a mouse atlas dataset containing 81,173 adult mouse cells from 13 tissues and 40 cell types [26]. Each cell type is annotated by borrowing label information, inferred by marker genes, from the RNA-seq data. A previous effort applied the mouse atlas dataset to benchmark multiple computational methods on scATAC-seq data [19]. The leading method in that study, SnapATAC, was the only method that could process the entire mouse atlas dataset within a reasonable time (12 h). Given that both SAILER and SCALE are deep learning methods that can train and evaluate data in mini batches, they are capable of handling the scale of the mouse atlas dataset. Thus, we benchmarked SAILER against SnapATAC and SCALE on this dataset.

For SCALE and SAILER, we added a filtering process before loading the data. The filtering involved reducing the bin numbers according to the procedure for filtering peaks used in SCALE. For each cell, we removed bins with read counts of over 90% cells and less than 1% cells.

We used normalized mutual information (NMI) and the adjusted Rand index (ARI) to compare each method's clustering results with the given labels.

For clustering, we constructed a KNN graph and applied the Louvain algorithm [6] to assign clusters to each cell. We compared the clustering results with ground-truth labels to generate the ARI and NMI metrics. We also calculated mutual information between latent representation and confounding factors for comparison.

2.2.8 Performance benchmarking on multi-sample scATAC-seq datasets for mouse brain

To evaluate the ability of SAILER to deal with batch effects, we combined two mouse brain datasets: a mouse brain dataset from the 10X Genomics website and a mouse secondary motor cortex dataset (i.e., the MOs-M1 dataset) [33]. We first selected cells based on barcodes from the 10X mouse brain dataset. Then, we set a threshold and selected scATAC-seq profiles with a promoter ratio between 0.2 and 0.6 and a log10-transformed unique molecular identifier count [log10(UMI)] between 3 and 5. This process resulted in 4,100 cells selected from the 10X mouse brain dataset and 15,136 cells selected from the MOs-M1 dataset. Using the same filtering criteria to remove low-quality cells, we selected 9,646 cells from the MOs-M1 dataset for further analysis.

We then performed clustering on the lower-dimension representation learned by SAILER with a Louvain algorithm on a KNN graph. We applied t-SNE to generate a 2D visualization of the landscape. As cell labels are not available, we next visualized the activity scores of several marker genes to justify the clustering results. We selected several marker genes from the gene annotation file to obtain gene read counts within each cell. To avoid extreme sparsity and discontinued values, we adopted MAGIC to smooth the gene-cell matrix to obtain the final gene-level expression matrix. For each cell and each marker gene of interest, we applied gene expression values corresponding to each cell and denoted them by color in the t-SNE plot.

2.3 Results

We applied SAILER on both simulated and real datasets and carried out comprehensive performance benchmarking with existing methods, as discussed in the following sections below.

2.3.1 Extensive cell-to-cell variations in scATAC-seq data arise from confounding factors rather than biological heterogeneity

We found that, in addition to the underlying cell states, confounding factors from various sources significantly contribute to the cellular heterogeneity in scATAC-seq experiments. For instance, we extracted two mouse brain scATAC-seq datasets – one from the 10X genomics website (10X) and one from the SnapATAC website (MOs-M1) (see details in the Methods section). We uniformly processed these two datasets and found that the number of fragments within the same dataset varied significantly. For example, the uniquely mapped read counts per cell ranged from 1,500 to 6,000 for the MOs-M1 dataset (Figure 2.3). Moreover, datasets generated from different labs showed distinct signatures. Specifically, the MOs-M1 dataset sample had fewer reads per cell but was highly enriched in promoter regions (median read count 3.506 vs. 4.236, promoter ratio 0.337 vs. 0.290). Most existing methods ignore such confounding factors, resulting in biased latent cell representations in dimensional reduction.

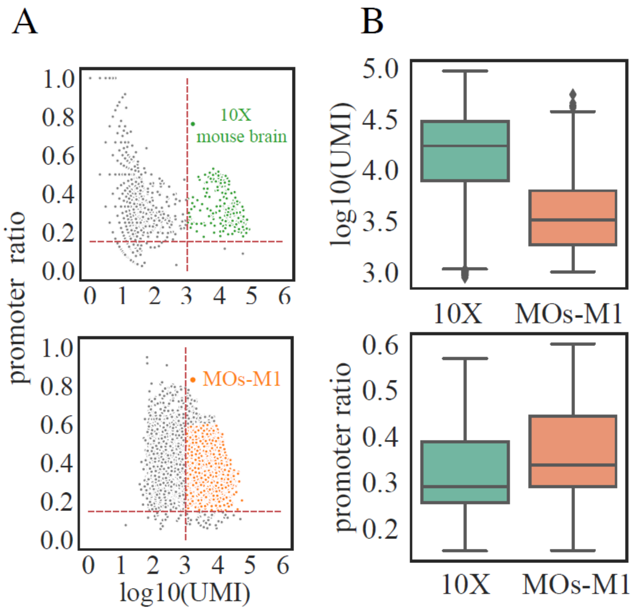


Figure 2.3: Visualization of confounding factors. (A) Scatter plots of a 10X mouse brain dataset (10X) and a mouse secondary cortex MOs-M1 dataset (MOs-M1). For all the cells in each dataset, we kept those with log10(UMI) between 0.3 and 0.5 and promoter ratio between 0.2 and 0.6. (B) Boxplots of read depth and promoter ratio comparison between selected cells from each dataset.

2.3.2 SAILER learns robust latent cell representations invariant to various confounding factors in simulated data

Here, we extensively benchmarked SAILER with existing methods using simulated data representing various application scenarios.

First, we simulated a deeply sequenced scATAC-seq dataset from five cell types, with varying mapping reads per cell. We learned the latent cell representations using SAILER, SCALE, and LSI as the input for the same clustering process. As shown in Figure 2.4A, linear methods like LSI could not capture the complex dependencies among the peaks and hence failed to distinguish the rare cell type from the major cell types (red dots in the gray cluster). In contrast, both SAILER and SCALE used a nonlinear dimension reduction via fully connected neural networks and were able to report five clearly separable clusters. Furthermore,

Table 2.4: Mutual Information between the latent representation and confounding factors on simulation datasets.

Method	Sim1	Sim2	Sim3
LSI	0.610	0.500	0.130
SCALE	0.290	0.224	0.087
SAILER	**0.107**	**0.100**	**0.005**

LSI and SCALE have a limited or no explicit module for correcting read depth effects. As a result, their L-shaped cell clusters are severely confounded by fragment counts, as reflected by the smooth transition from shallowly sequenced cells to densely sequenced ones within each cluster (the yellow to red pattern in Figure 2.4A, Sim1). Such artifacts would be further amplified in the downstream imputation analysis, because cells with more mapped reads will exhibit even larger read counts after incorporating information from their similarly deeply sequenced neighbors. Such artifacts would be further amplified in the downstream imputation analysis, because cells with more mapped reads will exhibit even larger read counts after incorporating information from their similarly deeply sequenced neighbors. On the contrary, SAILER penalizes such depth effects by introducing an extra penalty term to force the latent cell representations to be as independent as possible to fragment counts per cell, resulting in compact round-shaped clusters with almost random read count distributions (Figure 2.4A, Sim1). This observation is consistent with the quantitative measure of the mutual information $I(\mathbf{z}, \mathbf{c})$ between read counts and cell embeddings, where SAILER reported the lowest $I(\mathbf{z}, \mathbf{c})$ at 0.107 among all three methods (0.290 and 0.610 for SCALE and LSI, respectively, Table 2.4, Sim1). Thus, SAILER effectively removes confounding factors and learns robust cell representations.

We further simulated another shallowly sequenced dataset with fewer fragments per cell but more cell types, in order to conduct clustering performance benchmarking under more complicated (and realistic) scenarios. As shown in Figure 2.4B, SCALE and LSI failed to separate two major cell types by reporting completely overlapped clusters (yellow and purple dots in Figure 2.4B). Similar to the previous simulation, we observed clear low-to-high read

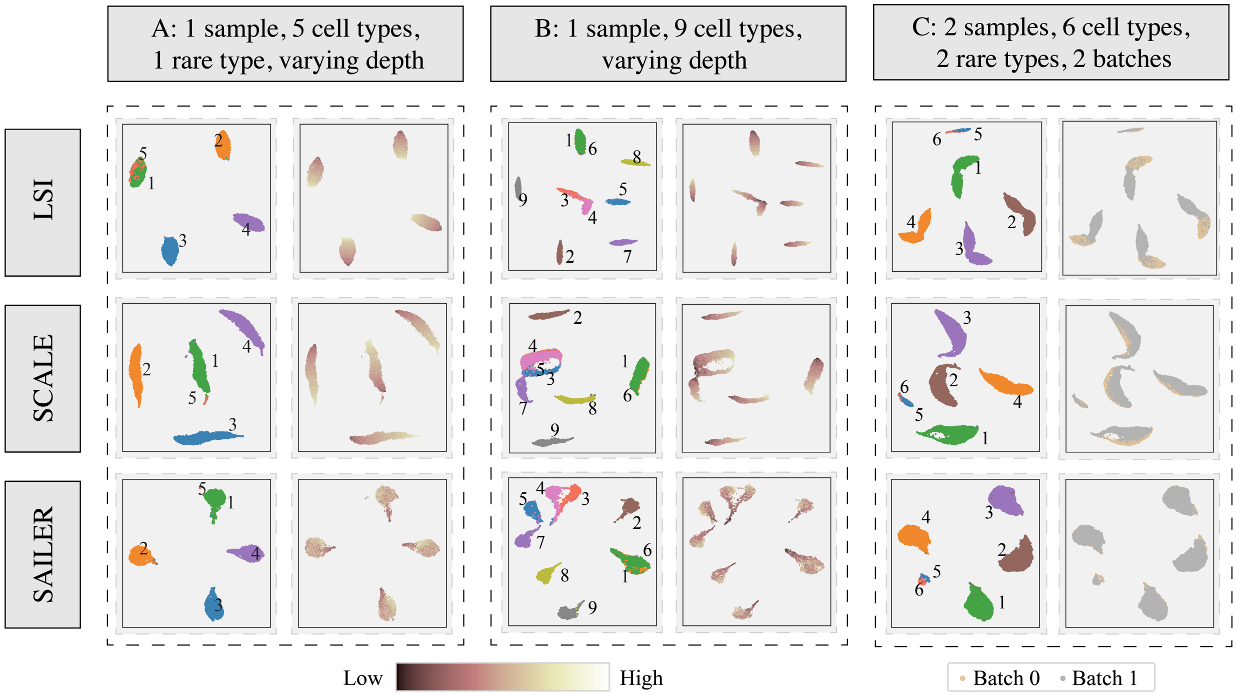


Figure 2.4: Results on simulation datasets. (A) 2D visualization of learned latent representations of LSI (top), SCALE (middle), and SAILER (bottom) on the Sim1 dataset. The left column shows the distribution of cell types. The right column shows the distribution of read depth indicated by color depth. (B) 2D visualization of learned latent representations of LSI (top), SCALE (middle), and SAILER (bottom) on the Sim2 dataset. The left column shows the distribution of cell types. The right column shows the distribution of read depth indicated by color depth. (C) 2D visualization of learned latent representations of LSI (top), SCALE (middle), and SAILER (bottom) on the Sim3 dataset. The left column shows the distribution of cell types. The right column shows the distribution of cells from different batches.

count transitions within their reported clustering, indicating severe read depth artifacts. By contrast, SAILER distinguished cell types from distinct cell states into clear groups and demonstrated homogeneous read counts within each cluster (bottom row, Figure 2.4B), indicating effective read depth bias removal. As expected, SAILER also showed the smallest amount of mutual information between fragment counts and latent cell representations (0.100 vs. 0.224 for SCALE and 0.500 for LSI, Table 2.4, Sim2), confirming the efficacy of its invariant representation learning scheme.

Lastly, we designed a third simulation dataset to mimic the scATAC-seq integration scenario with obvious batch effects for all three methods. We used latent representations to generate 2D visualizations with UMAP, as shown in Figure 2.4C and Figure 2.5A. We applied both batch information (right column) and cell-type information (left column) to annotate the plots. As shown in the right column, even though LSI and SCALE can marginally cluster the same type of cells, there are still clear boundaries between these batches. However, SAILER merges different batches very well, indicating that this method can remove batch information and retrieve the true distribution of cell biological states via the invariant latent representations. In order to quantitively measure how well these two batches are merged using different methods, we also calculated the mutual information between the batch information and each dimension of the latent representations (i.e., $I(\mathbf{z}, \mathbf{c})$), as shown in Table 2.4. SAILER still had the lowest value of mutual information (0.005, compared to 0.130 and 0.087). Note this dataset contains two sample-specific rare cell types (red and green dots, Figure 2.4C), representing a potentially common situation in which certain rare cell types only appear in a few batches. LSI and SCALE completely merged the rare cell types together; however, SAILER was able to distinguish these two cell types after removing depth variation and batch effects from the latent representation.

We also compared SAILER with pipelines involving specific mechanism for batch effect removal. SnapATAC incorporates Harmony [57] into their pipeline after dimensional reduction to remove batch effect. However, when processing Sim3 dataset, in which two different rare cell types appear in different batches, Harmony aligned two different rare cell types together by mistake, while SAILER is able to marginally distinguish these two rare cell types while keeping major cell types from different batches well mixed (Figure 2.6 A-C).

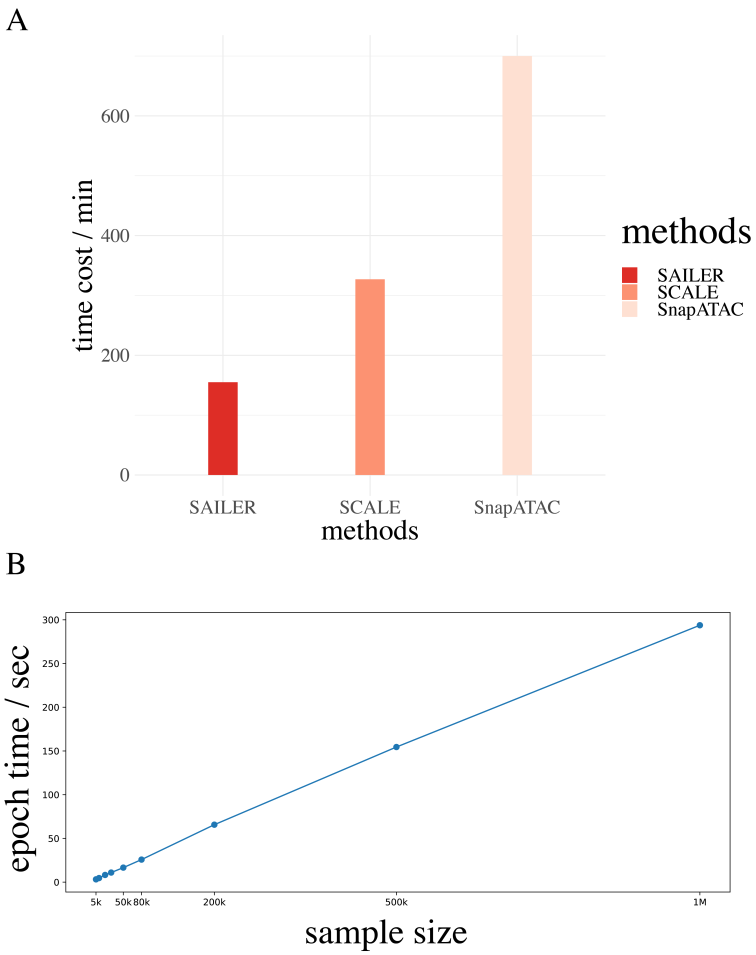
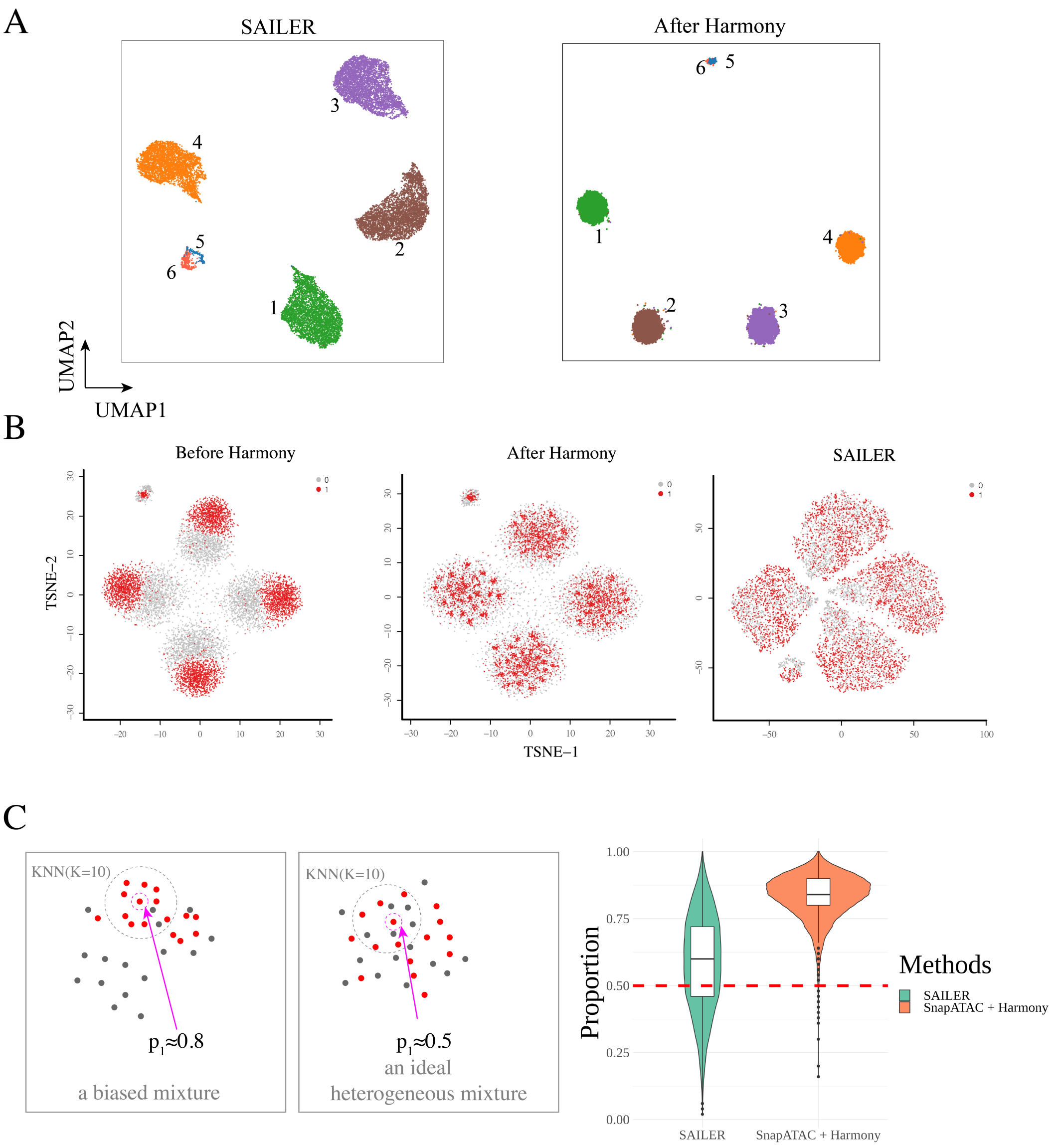


Figure 2.5: **Runtime and Scalability Summaries.** (A) Runtime comparison of SAILER, SCALE and SnapATAC on mouse atlas dataset. (B) Scalability of SAILER for different sample sizes.

2.3.3 SAILER outperforms existing methods in atlas-scale data analysis by reporting clearly separable clusters

To test the efficiency and accuracy of SAILER in a large-scale analysis, we benchmarked our method on a mouse atlas scATAC-seq dataset with 80k cells from 40 cell types with substantial read depth variations, as shown in Figure 2.7. We benchmarked SAILER with the GMM VAE in SCALE, and SnapATAC, the leading and only algorithm that was able to perform large-scale scATAC-seq analysis in a previous benchmarking study [19]. As shown in Figure 2.7, SAILER can learn robust cell representations that generate tight and clearly



(a) (b)

Figure 2.6: **Batch effect correction comparison.** (A) SAILER (left) compared with other batch-effect correction SnapATAC w/ Harmony (right) on Sim3 dataset. Color indicates cell types. (B) SAILER compared with SnapATAC w/ Harmony on batch-effect correction. (C) Neighbor composition distribution for SnapATAC w/ Harmony and SAILER.

separable clusters, as compared to other methods.

Besides, due to the lack of effective read depth removal, clustering results from SCALE are significantly confounded by the total number of fragments per cell. Specifically, the direct neighbors of deeply sequenced cells in SCALE's reports are mostly those with higher read

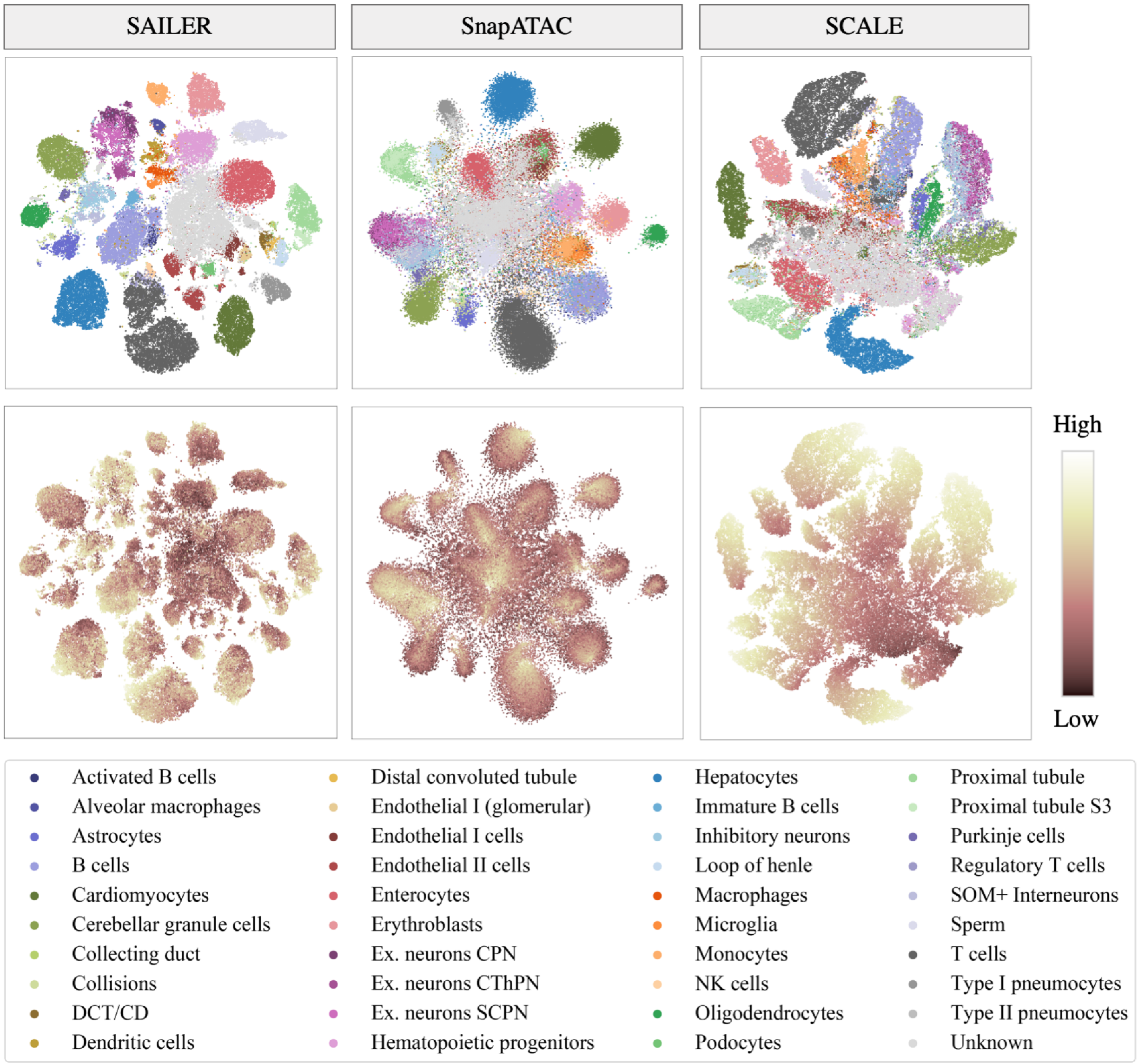


Figure 2.7: Results on the mouse atlas dataset. t-SNE visualization of lower-dimensional representation generated by SAILER (left), Snap-ATAC (middle), and SCALE (right). The first row shows the distribution of cell types. The second row shows the distribution of read depth indicated by color depth.

counts in each cluster (light dots in the bottom line, Figure 2.7). This read depth effect will severely impact the subsequent imputation analysis, as depth imbalance among cells will be amplified when considering the neighbors. SnapATAC tends to remove such depth effects by regressing out fragment counts per cell in the cell-to-cell similarity calculation. As a result, its identified clusters are less affected by read depth. However, several internal groups were

Table 2.5: Evaluation results on the mouse atlas dataset.

Method	ARI	NMI	$I(\mathbf{z}, \mathbf{c})$
SAILER	**0.575**	**0.799**	**0.040**
SnapATAC	0.538	0.748	0.127
SCALE	0.315	0.557	0.279

mixed together without clear separation, probably due to its independence and the equal contribution assumption among various genomic regions in the Jaccard distance calculations. Unlike SnapATAC, which requires a separate process for depth variation removal, SAILER integrates depth removal into the learning process – the fully connected neural network layers in SAILER allow nonlinear interactions among different genomic regions to better separate cells from different biological states, while the extra mutual information penalty term effectively removes read depth effects. This unified framework of SAILER makes each task aware of the other tasks, resulting in noticeably improved clustering results. This noticeable improvement can also be seen in the resulting NMI and ARI scores (Table 2.5). For instance, SCALE and SnapATAC reported NMI scores of 0.557 and 0.748, respectively, using known cell type-level labels, whereas SAILER showed a significantly higher NMI of 0.799. Moreover, SAILER reported lower mutual information (0.04), compared with 0.127 in SnapATAC and 0.279 in SCALE, suggesting successful depth effect removal for this method.

It is worth mentioning that the complexity of the batch-based training process increases linearly with the size of the input dataset (Figure 2.5B), resulting in better scalability of SAILER to efficiently process millions of cells in multi-sample analyses. However, the polynomial regression approach used in SnapATAC increases quadratically as the number of cells increases. Chen et al. reported that Snap-ATAC takes nearly 12 hours to process the entire mouse atlas dataset [19], while SAILER can complete this process within 4 hours trained for 400 epochs. We compared the runtime against another deep learning method SCALE on the mouse atlas dataset. As the result shown in Figure 2.5A, SAILER achieves the lowest running time in all the three methods. Benefitting from the batch-based training scheme and GPU parallel acceleration, SAILER could handle the running process even when running sample increases to large scale in a reasonable memory cost (Supplementary Note 6 and Figure 2.5B). This further demonstrates the advantage of the deep learning method when scaling to very large datasets.

Moreover, we also followed the preprocessing procedures for subsampling by 10k cells for performance benchmarking with 17 other methods, as most methods cannot handle an atlas-scale dataset. Instead of cell-type labels, we used the same tissue-level cell labels for comprehensive clustering benchmarking. When applied to the subsampled dataset, SAILER still achieved the highest ARI (0.397) among all methods (with the 17 other methods ranging from 0.009 to 0.363). This further demonstrates the effectiveness of our method.

2.3.4 SAILER can effectively remove batch effects in multi-sample scATAC-seq integration

Another common source of confounding factors are batch effects in multi-sample scATAC-seq analysis, where samples may be processed and sequenced from different labs or even sequencing platforms with distinct sample-specific signatures. To evaluate the performance of our method in such scenarios, we applied SAILER on two mouse brain scATAC-seq samples from two sources – one mouse brain dataset from the 10x Genomics website (10X) and one generated from mouse secondary cortex brains [33].

For fair performance benchmarking, we uniformly processed these two datasets to identify cells from random barcodes using the default parameters in SnapATAC [33]. Specifically, after removing barcodes with less than 1,000 fragments and keeping the remaining ones with promoter ratios between 0.2 and 0.6, we identified 4,100 and 9,646 cells from these two samples (see details in the Methods section). Starting from the same tissue, we found that these two samples generated from different labs showed distinct fragment signatures.

For instance, the dataset from the 10X Genomics website demonstrated a higher mean read coverage per cell (log(UMI) = 4.149 vs. 3.547, P-value = 10e-15 using the two-sided Wilcoxon test) and a lower mean promoter ratio (0.320 vs. 0.367, P-value = 2.48e-87 using the two-sided Wilcoxon test). After pre-processing, we projected the remaining cells into a ten-dimensional space using SAILER and SCALE, and then generated a KNN graph (k=16) and performed clustering via the Louvain algorithm. We also used t-SNE to map the ten-dimensional cell representations onto a 2D space for visualization and labeled the sample IDs using different colors in Figure 2.8. In the ideal case, a good computational method should overcome batch effects by reporting cell clusters with homogenous sample ID distributions. However, due to the lack of an appropriate batch effect removal module, we found that clusters reported by SCALE were predominantly driven by sample effects rather than the true biological states of the cells (Figure 2.8A). In contrast, SAILER effectively removed batch effects by introducing an additional penalty to reduce the mutual information $I(\mathbf{z}, \mathbf{c})$ between the variant component and the batch component in the objective function. As a result, the different samples were homogeneously mingled in the clearly separated clusters reported by SAILER (yellow and grey dots in Figure 2.8A). Furthermore, we also compared the embeddings generated by SAILER with SnapATAC (with Harmony) on these datasets to measure the ability of handling platform-to-platform variations. Similar clustering and mixing result of the two compared methods on these two datasets further demonstrating the potential of SAILER dealing with platform-based batch effects (Figure 2.9A).

To test whether these SAILER-reported clusters represent distinct biological cell states, we calculated the overall chromatin accessibility scores of well-known marker genes [33] and labeled cells using the activity scores of the marker genes. As shown in Figure 2.8B, SAILER identified clearly separable cell clusters that correspond well with the activities of the marker genes (*sst*, *pvalb*, *gad2*, and *plp1*). For instance, *sst* is a well-known marker gene widely expressed in inhibitory neurons. SAILER homogeneously grouped together *sst*-enriched cells from different batches, demonstrating its ability to appropriately remove batch

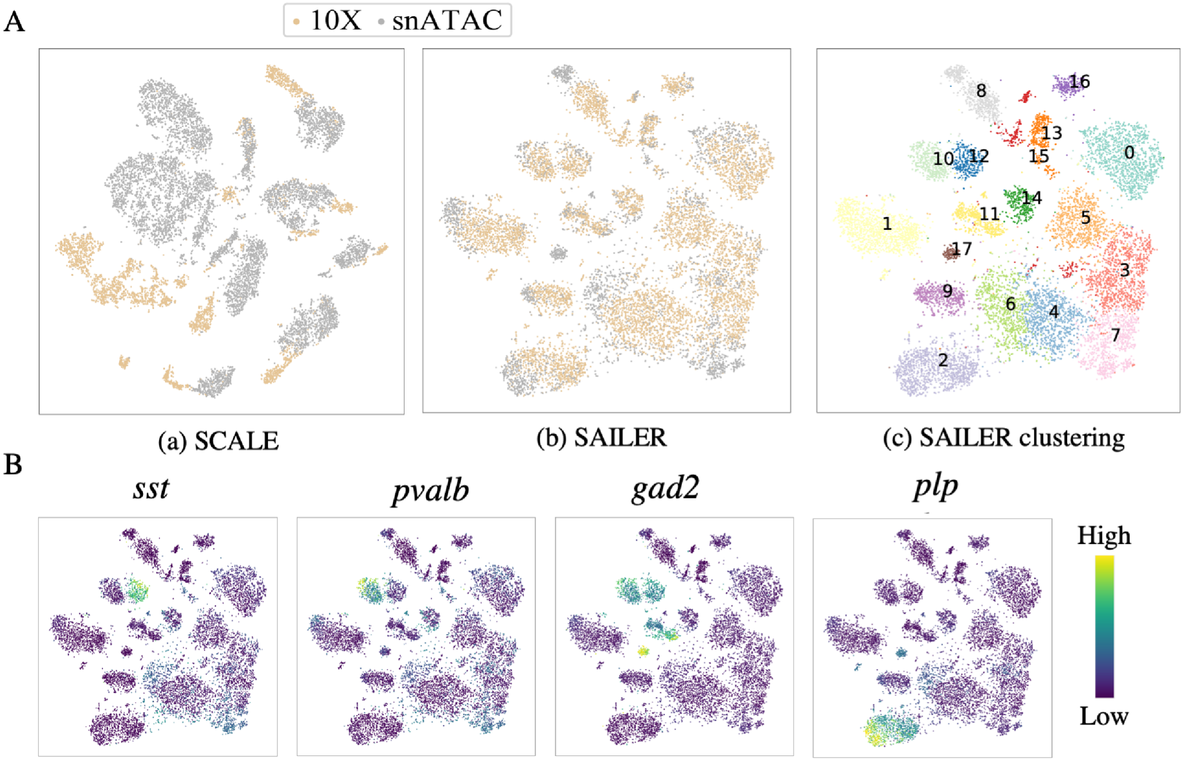


Figure 2.8: Results on mixed mouse brain datasets. (A) Clustering result comparison of SCALE and SAILER on two batches of mouse brain cell samples. Clustering result (a) using SCALE, (b) using SAILER, and (c) using SAILER but colored and labeled with numbers calculated using the Louvain method based on the KNN graph. (B) Clustering result of SAILER on two batches of datasets but colored with four marker gene scores, namely sst, pvalb, gad2, and plp1. The brighter the color, the higher the gene score shown for those cells.

effects while retaining the true cell-cell variability.

2.3.5 SAILER can precisely reconstruct a chromatin accessibility landscape free of various confounding factors

Despite high throughput in revealing epigenetic heterogeneity, scATAC-seq experiments suffer from severe missingness by reporting only a few thousand fragments in the entire genome. Therefore, accurate chromatin landscape reconstruction and imputation are essential to un-

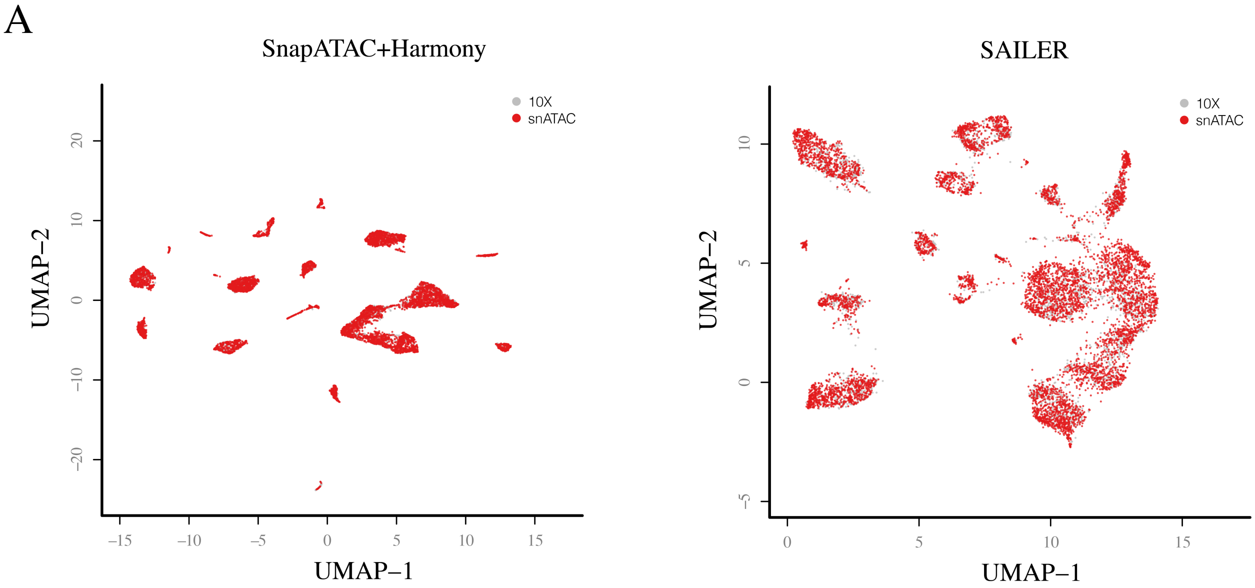


Figure 2.9: Batch correction methods comparison on different sequencing platforms. (A) UMAP visualizations of latent landscapes generated by SnapATAC+Harmony (left) and SAILER (right) on merging two datasets of mouse brain generated with combinatorial indexing single nucleus ATAC-seq (MOs-M1/ snATAC) and droplet-based platform (Mouse Brain 10X / 10X) respectively.

covering the full regulatory potential within a cell. However, very few computational methods are designed explicitly for chromatin accessibility imputation.

Here, we took advantage of the deep generative model and its invariant representation to reconstruct a full chromatin accessibility landscape that is independent of sequencing depth and batch effects. During imputation, we fixed the values of the confounding variables, such that the variations of the reconstructed scATAC-seq data only depend on the invariant representation z, which reflects the intrinsic variation of biological states.

To further demonstrate this, we performed imputation on the third simulation dataset (Sim3) with two simulated samples. SCALE is currently the only available method designated for imputing scATAC-seq data. LSI has no direct imputation module, we added MAGIC as suggested for benchmarking [38]. First, SAILER, MAGIC, and SCALE generated the imputed data. These data, along with the raw data, were then processed by PCA and visualized with UMAP in 2D. From the PCA embeddings shown in Figure 2.10, we found

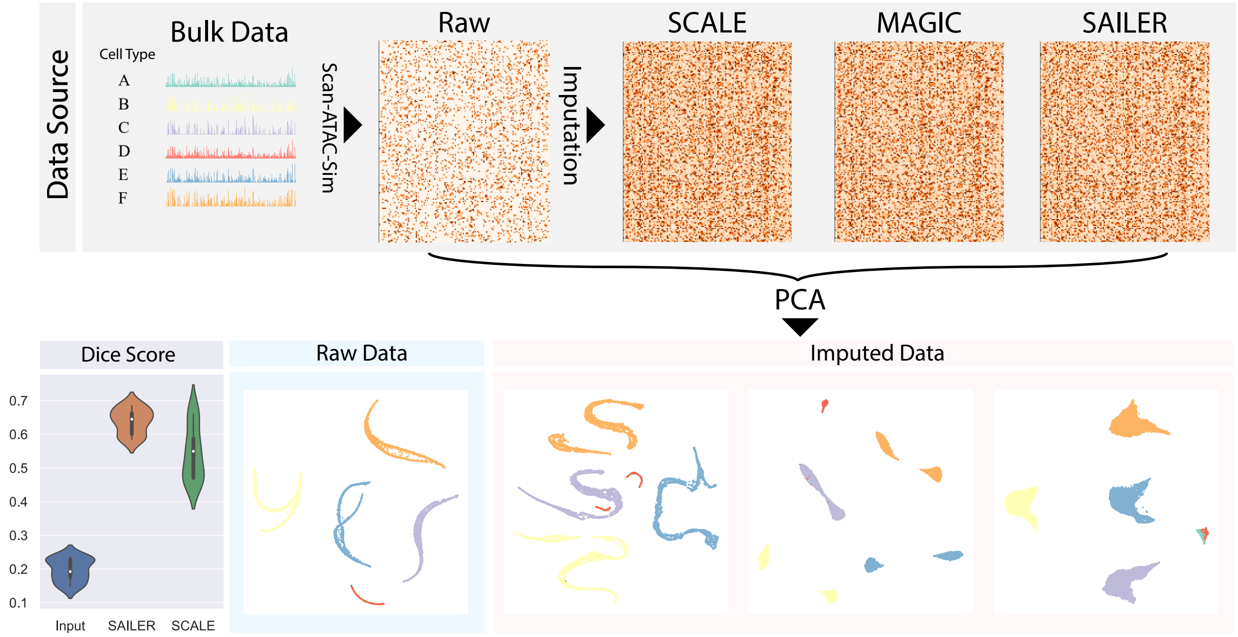


Figure 2.10: Imputation pipeline and results. Simulated data (Sim3) with 2 batches is generated by Scan-ATAC-Sim tool. Imputed data is generated by running SCALE, MAGIC, and SAILER, respectively. Imputed data is then subject to PCA and visualized by UMAP. Dice Score is computed between each imputed data and the Bulk Data. The Dice score between Input data and Bulk data is also shown as baseline.

that the imputation data of SCALE were severely affected by depth variation and batch effects. We observed similar results with MAGIC, where after imputation, the same types of cells from different batches were divided into separate clusters in the PCA embedding. However, the imputed data by SAILER did not show separate clusters from different batches. Moreover, the rare cell types (shown in green and red, Figure 2.10) were separable in the PCA embedding, which was not the case for SCALE or MAGIC. The results indicate that, without proper removal of confounding factors during imputation, the imputed data show clear variations that correlate with confounding factors. In addition, the data diffusion strategy used in MAGIC is not friendly to rare cell types, as the rare cells can be easily overwhelmed by the major cell types. Thus, compared with SCALE and MAGIC, SAILER is the only method capable of removing confounding factors from imputation data, while preserving unique information from rare cell types.

As the bulk ATAC-seq data used to simulate the single-cell data is available, we used the bulk data as the ground truth and calculated the DSC for each imputation method. The DSC (also known as the F1 Score) is a harmonic mean of the precision and recall. Because scATAC-seq is imbalanced in 0/1 entries, we used DSC as a balanced metric to evaluate the imputation performance. We generated a violin plot to show the DSC distributions of raw single-cell data, SAILER, and SCALE. As shown in Figure 2.10, SAILER and SCALE both achieved higher DSC scores compared to the raw data, indicating that both methods generate reasonable imputation results. SAILER achieved a higher mean DSC compared with SCALE (0.64 vs. 0.54), further demonstrating the effectiveness of invariant representation learning.

2.4 Discussion

In this work, we developed a scalable and accurate single-cell ATAC-seq processing and integration method called SAILER via efficient invariant representation learning. As compared with previous methods, SAILER has three distinct characteristics designed explicitly for single-cell data analysis – 1) it utilizes nonlinear dimension reduction via fully connected neural networks in a deep generative framework to handle complex dependencies among various peaks; 2) it dissociates cell-state-related biological variations from those arising from confounding factors (e.g., read depth and batch effects) to faithfully embed the cells into a low-dimensional latent space to facilitate various downstream analyses, such as cell clustering and imputation; 3) it is easily scalable to large-scale single-cell data analysis accelerated using GPU parallelism.

We applied SAILER to various simulated and real scATAC-seq datasets and comprehensively compared its performance with state-of-the-art analysis pipelines. We showed that SAILER's robust cell embeddings can effectively remove noise impacts from different sources and improve clustering and imputation results on all of the benchmark datasets. We should note that the invariant representation learning framework presented here is general and can be applied to other types of high-throughput genomic data like scRNA-seq and single-cell DNA methylation, or to joint analysis of multi-modality single-cell genomics data. Specifically, several single-cell multi-omics technologies have recently emerged for measuring multiple types of molecules in the same cell [50]. To achieve this, we could apply a multi-modal VAE to encode a variational posterior jointly from single-cell multimodal omics inputs using deep neural networks, where the resultant latent space factors into a shared subspace to profile cell states or functions for individual cells and private subspaces could be used to solve specific technical issues for each modality.

In summary, we developed a deep generative model, SAILER, for learning robust latent cell representations invariant to changes in various noise factors, which has not been possible with most current scATAC-seq analysis tools. Given the fast-expanding collection of publicly available single-cell sequencing data, we envision that the SAILER framework can serve as a powerful tool to remove impacts from confounding factors and uncover cellular heterogeneity across diverse cell states and conditions in large-scale single-cell omics data analysis.

2.5 Supplementary Notes

2.5.1 Results on Hyperparameter Robustness

In this section, we present evaluation results under different hyperparameter settings. In particular, we show the mean mutual information $I(\mathbf{z}, \mathbf{c})$, Adjusted Rank Index (ARI), and Normalized Mutual Information (NMI) between cluster assignments and ground truth labels evaluated on the mouse atlas dataset for different hyperparameter settings. The default setting is $\lambda = 1$, $dim(\mathbf{z}) = 10$, number of intermediate neuron units is 100. For each experiment, we change one of the hyperparameter listed above. The results are shown in

Table 2.1, 2.2, 2.3 and Figure 2.2.

2.5.2 Derivations of the Mutual Information Objective

In this section, we show the detailed derivation of the mutual information objective [76] used in our model.

With properties of Mutual Information and a variational inequality, we have

$$\begin{aligned} I(\mathbf{z},\mathbf{c}) &= I(\mathbf{z},\mathbf{x}) - I(\mathbf{z},\mathbf{x}|\mathbf{c}) && (2.8)\\ &= I(\mathbf{z},\mathbf{x}) - H(\mathbf{x}|\mathbf{c}) + H(\mathbf{x}|\mathbf{z},\mathbf{c}) && (2.9)\\ &\leq I(\mathbf{z},\mathbf{x}) - H(\mathbf{x}|\mathbf{c}) - \mathbb{E}_{\mathbf{x},\mathbf{c},\mathbf{z}\sim q}[\log p_\theta(\mathbf{x}|\mathbf{z},\mathbf{c})] && (2.10)\\ &= \mathbb{E}_{\mathbf{x}}[D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||q_\phi(\mathbf{z}))] - H(\mathbf{x}|\mathbf{c}) - \mathbb{E}_{\mathbf{x},\mathbf{c},\mathbf{z}\sim q}[\log p_\theta(\mathbf{x}|\mathbf{z},\mathbf{c})] && (2.11) \end{aligned}$$

$H(\mathbf{x}|\mathbf{c})$ doesn't involve $\mathbf{z}$, thus it could be ignored during the optimization. Terms from the above equation looks similar as the VAE objective, with some modifications on the conditional log likelihood.

$$L_{\mathrm{VAE}} = \mathbb{E}_{\mathbf{x},\mathbf{c}\sim q(\mathbf{x},\mathbf{c})}\left[-\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z},\mathbf{c})] + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}))\right] \quad (2.12)$$

According to the equation above, the ELBO of VAE is modified to minimize the negative log likelihood conditioned and the mutual information between latent variable $\mathbf{z}$ and confounding factors $\mathbf{c}$ for invariant representation learning.

$$\min\ L_{\mathrm{VAE}} + \lambda I(\mathbf{z},\mathbf{c}) \quad (2.13)$$

Putting above equations together, we have the final objective,

$$
\begin{aligned}
L(\phi, \theta) = \mathbb{E}_{\mathbf{x} \sim q(\mathbf{x})} \left[D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})) + \lambda D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel q_\phi(\mathbf{z})) \right] \\
-(1+\lambda)\mathbb{E}_{\mathbf{x},\mathbf{c} \sim q(\mathbf{x},\mathbf{c})} \left[\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z},\mathbf{c})] \right] \qquad (2.14)
\end{aligned}
$$

2.5.3 Results on Batch Effect Correction Benchmarking

In this section, we show benchmark results on Sim3 dataset using SnapATAC [33] with Harmony [57] and SAILER. Sim3 dataset contains 2 batches, 6 types of cells including 2 rare cell types. Harmony is applied after dimensional reduction in SnapATAC pipeline to remove batch effect. We show UMAP visualization of latent landscape colored by cell type (Figure 2.6A), t-SNE visualization of latent landscape colored by batch (Figure 2.6B). As shown in the Figure 2.6A, SnapATAC with Harmony failed to separate the two batch specific rare cell types (blue+orange cluster), while SAILER's unified framework separated these two cell types successfully. In Figure 2.6B, SnapATAC without batch effect correction clearly shows separated batches even within the same cell type. Harmony can align different batches together, but with obvious sub-cluster patterns. On the contrary, SAILER merges these cells quite well by reporting locally homogeneous mixing from different batches.

We also calculated a quantitative measure of the mixing of cells from different batches in three steps. 1) Build the KNN graph for each cell (K=50); 2) Find the 50 nearest neighbor of each cell in the embedded space; 3) For each cell, calculate the proportion of its nearest neighbors from batch 0 and batch 1, denoted by p_0 and p_1 separately.

Intuitively, a good batch effect correction method will provide p_0 and p_1 approximately 0.5 (after cell number normalization) if two batches from different platforms is well mixed, otherwise will result in a biased mixture, as shown in Figure 2.6C(a). From Figure 2.6C(b), we can see that SAILER has better local mixture of two batches by reporting balanced

p_1 values, as compared with SnapATAC with Harmony, which indicates improved local homogeneity, proving that our method is more robust in dealing with multiple batches.

2.5.4 Runtime and Scalability

We compared the runtime of three methods benchmarked with the mouse atlas dataset[26], namely SAILER, SCALE [115], and SnapATAC. Figure 2.5A shows the runtime of three methods. Both deep learning methods SCALE and SAILER are trained thoroughly for 400 epochs using a NVIDIA RTX 2080Ti GPU. Scalability of SAILER is tested on re-sampled mouse atlas dataset with sample size ranging from 5k to 1M. Results are shown in Figure 2.5B. SAILER achieves the shortest runtime. In the meantime, runtime of SAILER scales linearly up to sample size of 1M cells in our experiment.

2.5.5 Results on Batch Effect Correction on different platforms

Last but not least, batch effects are often caused by experiments from different platforms. These platform-to-platform variations also play a vital role in separating cells apart even if they are originated from the same cell type. To evaluate the clustering performance of our method SAILER on cells from different platforms, we choose SnapATAC (with Harmony) as comparison and draw the UMAP visualization plot on two mouse brain datasets [33] generated using combinatorial indexing single nucleus ATAC-seq platform (MOs-M1/ snATAC) and droplet-based platform (Mouse Brain 10X / 10X) respectively. As the result shown in Figure 2.9, both SnapATAC (with Harmony) and SAILER performs quite well in mixing these two platform cells together, further demonstrating that our model is quite robust in dealing with batch effects.

2.5.6 Results on memory cost

In terms of memory usage, since CPU based method like SnapATAC is different from GPU based methods like SAILER and SCALE, so to evenly compare its cost, we monitor the memory cost of the two GPU-based, namely SAILER and SCALE, on mouse atlas dataset during training process, their memory cost are 4319 megabytes and 4553 megabytes respectively using NVIDIA 2080ti GPU, which is just 1/3 of the maximum memory of one GPU card. Thus, the model could be expected to apply to more memory consuming datasets. The reasonable memory cost also denotes a useful application for other memory consuming datasets.

Chapter 3

Integrating Multimodal single-cell data with structural similarity

Multimodal single-cell sequencing technologies provides unprecedented information on cellular heterogeneity from multiple layers of genomic readouts. However, joint analysis of two modalities without properly handling the noise often leads to overfitting of one modality by the other and worse clustering results than vanilla single-modality analysis. How to efficiently utilize the extra information from single cell multi-omics to delineate cell states and identify meaningful signal remains as a significant computational challenge. In this work, we propose a deep learning framework, named SAILERX, for efficient, robust, and flexible analysis of multi-modal single-cell data. SAILERX consists of a variational autoencoder with invariant representation learning to correct technical noises from sequencing process, and a multimodal data alignment mechanism to integrate information from different modalities. Instead of performing hard alignment by projecting both modalities to a shared latent space, SAILERX encourages the local structures of two modalities measured by pairwise similarities to be similar. This strategy is more robust against overfitting of noises, which facilitates various downstream analysis such as clustering, imputation, and marker gene detection. Furthermore, the invariant representation learning part enables SAILERX to perform integrative analysis on both multi- and single-modal datasets, making it an applicable and scalable tool for more general scenarios.

3.1 Introduction

Single cell sequencing (sc-seq) offers genome-wide measurements of genetic information from individual cells [101, 14, 30, 26, 77, 53, 96]. Recent technology advances allow simultaneous profiling of multiple modalities in the same cells [20, 69], allowing us to dissect cellular heterogeneity from multiple layers and investigate the transcriptomic and epigenomic interplays at the finest possible resolution.

Several computational methods have been developed to deal with some key factors of data integration, such as correcting batch effect while maintaining biological patterns for scRNA-seq data (scVI, scANVI, Scanorama, Harmony etc.) [66, 45, 57, 116], and embedding multi-modal data to the same embedding without corresponding information [65, 3, 59, 111, 95, 32, 15]. Readers can refer to [3] for a more detailed comparison of data integration methods. However, it is still remaining a challenge to effectively utilize information cross different modalities due to problems such as unbalanced signal-to-noise ratio (SNR), datasets with missing modalities, handling modality-specific noise factors and batch effects. Recently, many computational methods have been developed to analyze multimodal single cell data [36, 112, 50, 74, 114, 120]. A common strategy used by many methods is to project data from different modalities to a shared latent space. For example, existing methods like scAI, scMM, scMVAE, BABEL and Cobolt [50, 74, 114, 120, 37] use either Nonnegative Matrix Factorization (NMF) or Encoder-Decoder types of neural networks to project multiple modalities to a common latent space. Their underlying assumption is that measurements from different modalities are equally informative and share a common distribution, which does not hold

Table 3.1: Comparisons on the functionality of benchmarked methods.

Method	Approach	Nonlinear	Scalability	Multiome	Missing Modality	Bias Correction
Signac	LSI	×	×	✓	×	×
Schema	QP	✓	×	✓	×	×
SAILER	VAE-Inv	✓	✓	×	×	✓
Cobolt	MVAE	✓	✓	✓	✓	×
SAILERX	VAE-Inv	✓	✓	✓	✓	✓

under many circumstances. For instance, a typical scATAC-seq experiment usually reports 1,000 to 20,000 mappable fragments per cell over the entire 3.2 billion base pair genome, resulting in noticeably higher dropout rates and coverage variations as compared to the RNA modality from the same cell. As a result, lines of literatures pointed out that direct fusion of modalities with neural networks can introduce severe overfitting across modalities, resulting in poor separation of cell clusters in learned latent representation [94]. In observance of this, Sigh et al. proposed Schema framework by learning an affine transformation of similarity matrices through metric learning to find a joint representation of cells which is regularized to be similar to a reference embedding [94]. However, the flexibility of the transformation could limit the expressiveness of the joint embedding, and it does not explicitly handle batch effect and other technical noises. In another strategy, Signac [42] used weighted nearest neighbor (WNN) graph to generate a joint embedding based on predictability of data from two modalities of each cell. However, information fusion is done after separate embeddings are generated without considering latent interaction between the two modalities, potentially limiting the overall performance. Besides, most existing methods cannot handle sc-multiome data with missing modalities (due to either possible QC failures in one modality or data integrations from different sequencing protocols) or contain explicit mechanisms to handle technical noises in each modality, which are common in real data analysis (Table 3.1).

Hereby, to tackle these issues, we propose a deep learning framework, named SAILERX, to improve analysis of multiomics or hybrid of single- and multi-modal single cell sequencing datasets (Table 3.1). Distinct from existing methods, SAILERX can handle both parallel scRNA-seq and scATAC-seq multiome data, single modal scATAC-seq data, and a hybrid of these two types of data. To address the modality heterogeneity and avoid overfitting, we use the more robust gene expression information as a reference modality, to regularize the learning process of the chromatin accessibility modality. Specifically, scATAC-seq data is modeled with a Variational Autoencoder (VAE) and embeddings of scRNA-seq data are pre-trained and not explicitly modeled at training time. We further impose regularization via minimizing the distance between the pairwise similarity in the embedding space between two modalities (Figure 3.1), which encourages local structures of cells to be similar to the reference modality while accommodating substantially different technical noises across modalities. The resulting representation of cells implicitly contains information from two modalities and avoids the risk of overfitting. In the meantime, an invariant representation learning objective [76, 16] is used in the VAE framework to eliminate observable technical noises and allows integration of multiple datasets through end-to-end training. The modeling choice of SAILERX allows hybrid integration of datasets with scATAC-seq measures and datasets with paired scRNA-seq and scATAC-seq, effectively utilize the information from high quality multimodal data to improve the analysis of single-modal datasets.

We benchmark SAILERX with existing state-of-the-art (SOTA) methods for multi/single-modal single cell data analysis on three popular single cell datasets with different sequencing technologies and types of tissues. We show that SAILERX generates representations of cells that provide better clustering and imputation. We also demonstrate how the single modal scATAC-seq dataset could benefit from hybrid training. For biological applications, those improvements significantly benefit the downstream analysis of chromatin accessibility data. SAILERX is implemented in a python package freely to the community.

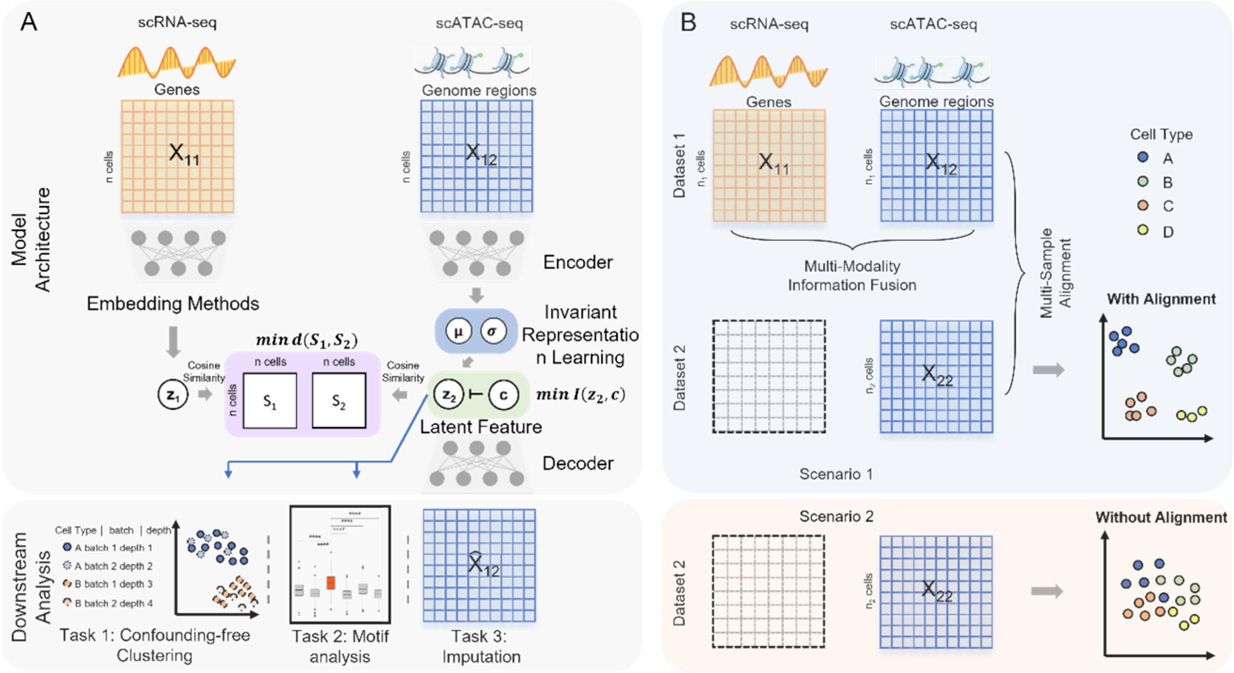


Figure 3.1: Overall design of SAILERX. (A) SAILERX takes co-assayed single cell RNA-seq and ATAC-seq data as input. scATAC-seq data is modeled with invariant representation learning through VAE, while embedding of scRNA-seq is processed during pre-training and not explicitly modeled in the training process. A regularization is imposed to encourage the local structure of cells in the embedding space to be similar between two modalities through minimizing the distance between pairwise cosine similarity matrices of two modalities. Latent scATAC-seq feature is further used to perform downstream analysis. (B) SAILERX is also capable of integrating single modal scATAC-seq with multimodal datasets through hybrid training, which could further enhance the clustering performance on single modality data.

3.2 Materials and Methods

In this section, we provide details on our SAILERX model and datasets for benchmarking, as well as describe methods.

3.2.1 Datasets

In this study, we focus on multimodal single cell sequencing data with paired scRNA-seq and scATAC-seq measurements. For this purpose, three popular public single cell multiomics datasets with different cell types and sequencing technologies are used in this study, namely

10x Genomics PBMC dataset [42], Share-seq dataset [69] and SNARE-seq dataset [20].

PBMC dataset

10X Genomics offers multiple datasets with PBMC cells, we collect PBMC 10k Multiome and PBMC 3k from the 10X genomics website. The PBMC 10k dataset is mainly used for benchmarking cross modality integration performance. For the PBMC 3k dataset, we only use the chromatin accessibility data for hybrid joint analysis with 10k dataset. The gene expression modality of 3k dataset is not used in hybrid training and only used for identifying ground truth labels of cells from the 3k dataset in this case. For integration of two sc-multiome datasets, the gene expression modality is used normally. For these two datasets, cell types are annotated through label transfer using an existing PBMC reference dataset via tools in the Seurat [42] and SeuratDisk package. Specifically, we use a high-quality dataset [42] as the reference dataset to transfer cell type labels to PBMC 3k and PBMC 10k datasets respectively.

For scenario one (cross modality integration), the 10k Multiome data is acquired from 10X genomics website. We first download PBMC 10k expression matrix and chromatin accessibility matrix as well as its fragment file from 10X Genomic Multiome dataset, and we follow the same quality control protocol as Signac [98] to filter out low quality cells. This retains 11,331 cells for further analysis. For scRNA-seq, we then normalize scRNA-seq data using SCTransform function with default parameters. After that, principal component analysis (PCA) is used to extract top 50 PCs for further clustering and joint analysis with scATAC-seq. As for scATAC-seq, since the set of peaks identified using CellRanger often merges nearby peaks, which would potentially cause bias in tasks like motif enrichment analysis, in our study, peak calling is performed on PBMC 10x dataset by using fragment file to generate unique peaks using MACS2 software [119]. After that, we follow the same process described in [114] and keep the autosome data and get the final scATAC-seq peak-by-cell matrix. This matrix is further used to process and benchmark with all the other methods. For instance, in Signac, TF-IDF is performed on the scATAC-seq matrix and then SVD is adopted on the TF-IDF output matrix to get the 50-dimension latent embedding, which is further used for clustering and joint analysis with scRNA-seq data.

Regarding to the second scenario (hybrid joint analysis), we use the aforementioned multimodal PBMC 10k data, which consists of scRNA-seq and scATAC-seq data as a reference and perform joint analysis with the chromatin accessibility data from PBMC 3k dataset. We retrieve PBMC 3k scATAC-seq data from 10X Genomics and treat it as a single modality dataset. We reason that 3k dataset with scATAC-seq contains less information than the multiomics dataset, however, since they come from the same types of cells, we could use 10k multiomics dataset as a reference to assist the analysis of 3k scATAC-seq data. We use reduce function from GenomicRanges package [61] to merge common peaks from scATAC-seq 10k and 3k dataset, and the peak by cell matrix is reconstructed separately for the two scATAC-seq data, which is further used to train and evaluate our model, as illustrated in Figure 3.1B.

Share-seq dataset

For Share-seq dataset, we retrieve Share-seq mouse skin dataset from Ma et al. (9), which contains 34,474 cells of both modalities of scRNA-seq and scATAC-seq data. For scRNA-seq data, we normalize its gene by cell matrix by using SCTransform function with default parameters from Signac package, then PCA is utilized to get top 50 PCs for further analysis. For scATAC-seq data, we keep the preprocessed peak by cell matrix used in Ma et al. The gene by cell and peak by cell matrices are used for evaluation on other methods.

Snare-seq dataset. For Snare-seq dataset, we download adult brain cortex data of two modality matrices from Chen et al. [20]. For scRNA-seq data, we follow the same processed steps as previous by normalizing gene by cell matrix using SCTransform function [42] with default parameters. After that we adopt PCA on the normalized matrix and use top 50 PCs as latent embedding for further analysis. As for the scATAC-seq data, after retrieving the processed scATAC-seq matrix from Chen et al, we also follow the same processed procedure as BABEL [114] and filter out low quality cells while keeping the original peaks unchanged. In details, genes that are encoded on sex chromosomes are first removed, and cells expressing fewer than 200 genes, or more than 2,500 genes are also filtered.

3.2.2 Model

Here, we describe details and implementation of our SAILERX model. SAILERX combines information from the gene expression measures to improve the downstream analysis of chromatin accessibility. SAILERX could also perform integrative analysis on multiple datasets with one or multiple modalities.

The model takes the co-assayed single cell multimodal data $x_i, i \in 1, 2, \ldots, M$ as input. We denote the gene expression data as $x^g_{1:M}$ and the peak data as $x^p_{1:M}$ (M indicates the total number of multimodal data samples). Our model could also take single modal scATAC-seq datasets $x^p_{M:B}$ (B indicates total number of sample batches) as input and perform integrative analysis among all$x^p = [x^p_1, x^p_2, \ldots, x^p_M, ..., x^p_B]$. The overall method follows the invariant representation learning framework based on Variational Autoencoders (VAEs) [76, 16].

$$L_{Inv} = L_{VAE} + \lambda I\left(z, c\right) \tag{3.1}$$

$$\geq E[-KL[q(z|x)||p(z)]] + (1+\lambda)E[logpxz, c] - \lambda KL[q(z|x)||q(z)] \tag{3.2}$$

In order to utilize the gene expression information provided by multimodal single cell samples, we add an extra term to regularize the local data structure in the chromatin accessibility posterior $q_\phi\left(z_{1:M}|x^p_{1:M}\right)$ to be close to the local structure measured by gene expression.

We use pairwise cosine similarity to describe the local data structure, where the cosine similarity is computed as

$$S = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{j=1}^{n} A_j B_j}{\sqrt{\sum_{j=1}^{n} A_j^2}\sqrt{\sum_{j=1}^{n} B_j^2}} \tag{3.3}$$

For each sample batch i, A and B are two single cell data vectors from $f\left(x^g_i\right)$ (where $f\left(x^g_i\right)$ is a transformation of raw gene expression data) and $q_\phi\left(z_i|x^p_i\right)$ for S^g_i and S^p_i respectively. In general, $f\left(\cdot\right)$ can be any embeddings of gene expression data preferred by user (e.g., a VAE or top PCs from PCA) since it is not parameterized by our neural network model here and only serving as a reference. For the convenience of comparing with existing methods, in our study, we mainly use the PCA results generated by Signac/Seurat [42] as the reference embedding. Some other scRNA-seq embedding methods (scVI [66], scANVI [116], Scanorama [45]) are also tested.

During the training, we minimize a distance-based objective $d\left(\cdot,\cdot\right)$ between the local pairwise cosine similarity matrix for each sample batch i calculated by gene expression data S^g_i and the pairwise similarity matrix calculated by latent distribution of peak data modeled by invariant VAE S^p_i, where both S's are b by b symmetric matrices with batchsize b for each minibatch during training.

$$L_{Local} = \sum_{i=1}^{M} d\left(S^g_i, S^p_i\right) \tag{3.4}$$

By choosing a proper differentiable distance metric $d\left(\cdot,\cdot\right)$, we can fuse this term into the end-to-end training of our deep generative model. The overall loss function would be the sum of the canonical VAE objective, a mutual information penalty, and the local similarity regularization. Here, we multiply a weight vector $\gamma = [\gamma_1, \gamma_2, \ldots, \gamma_b]$ with length b equals to the number of cells in current mini batch. This weight γ_j is calculated based on the ratio between read depth from gene expression modality and read depth from chromatin accessibility modality for each cell j. This weight vector is then subject to log transformation and min-max normalization to ensure stability $\gamma_j = MinMaxNorm\left[log\left(\frac{depth_{RNA}}{depth_{ATAC}}\right)\right]$. After scaling it with a constant scalar, we have our final weight vector $\gamma \in {R^{+}}^{b}$. The relationship between the scaling factor and the final L_{Local} is shown in Figure 3.2A. We note that after certain point, further increase of this scaling factor will no longer reduce the final L_{Local}. We recommend using this point as the choice for the scaling factor, as further increase of this weight does not transfer more information from the reference modality. Meanwhile, it may compromise the invariant representation learning objective, which could lead to problems in confounding factor removal or imputations. Also, from Figure 3.2B, we can see clustering metrics of SAILERX are robust in a relatively large range of weight values. In terms of choice of λ and dimension of latent variable, similar as in [16], the framework is robust against the choice of λ and dimension of latent variable.

The final loss of SAILERX is a summation of the invariant representation learning objective from equation 1 and the local alignment loss from equation 3 weighted by γ.

$$L = L_{Inv} + \gamma L_{Local} \tag{3.5}$$

In our implementations, we chose the Euclidean distance for $d(\cdot, \cdot)$ since it is differentiable and easy to calculate.

For the architecture of neural networks, we adapt the encoders and decoders structures from BABEL [114], where each chromatin is independently modeled by a two-layer dense encoder network, and outputs from each encoder network are concatenated with each other before being input to the final linear layer which yields the latent variable. The decoder

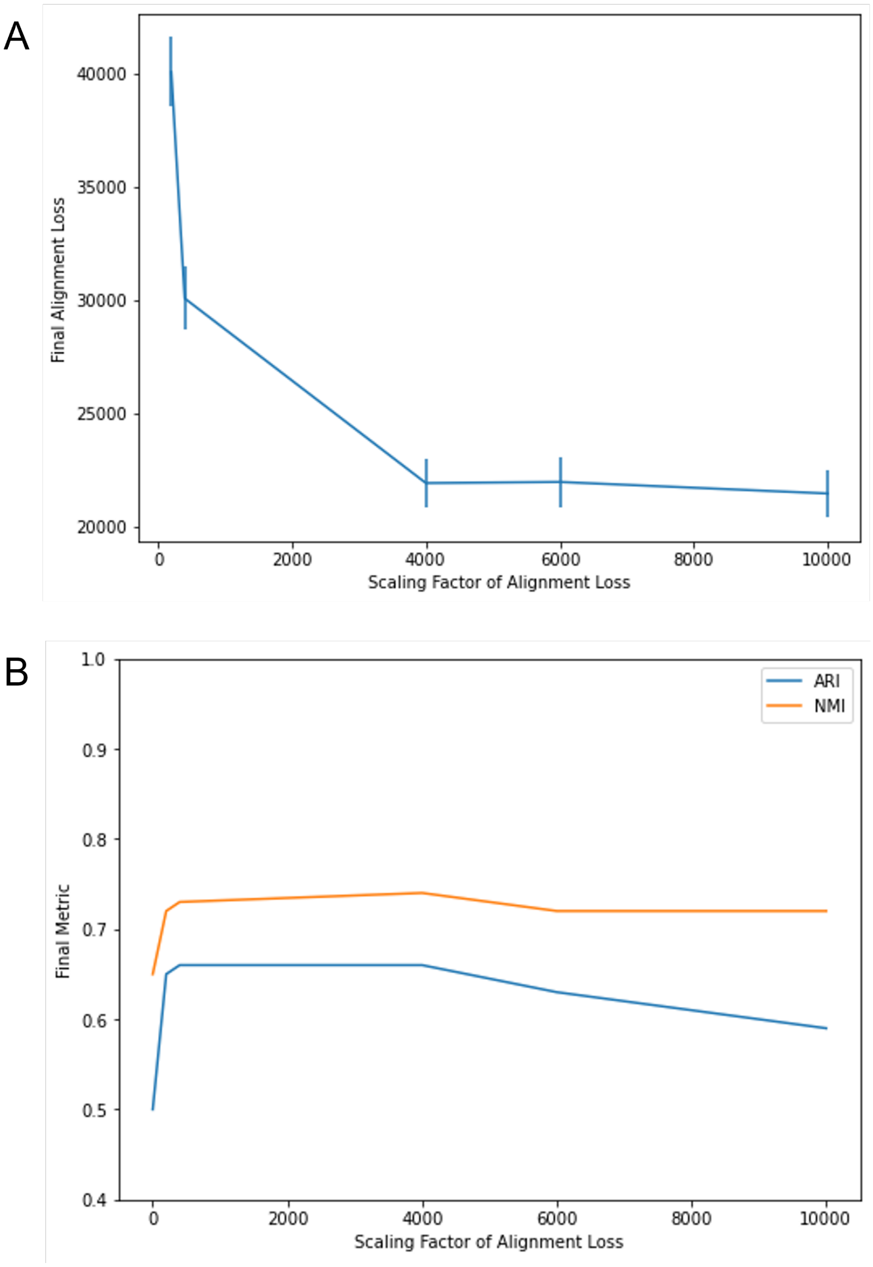


Figure 3.2: Results on hyperparameter stability. (A) The alignment loss L_{Local} decreases as the scaling factor increases. (B) Clustering metrics ARI and NMI as a function of scaling factor.

is symmetric to the encoder network, taking the latent and confounding variables as input and reconstructing the data. The assumption here is that interactions between genes and regulating factors are mainly within each chromatin. This type of modeling is efficient in memory consumption since it significantly reduces the total number of parameters. For fair comparison, the original SAILER encoder and decoder networks are also updated to the same structure.

3.2.3 Hybrid Training

One characteristic of SAILERX is that it allows integration of datasets with missing modalities (when $B > M$). In this scenario, for datasets with both modalities measured ($x_i, i \in 1, 2, \ldots, M$), the loss function follows the form of equation (3.5), where a reference embedding is available for calculatingL_{Local}; for datasets with only one modality ($x_i, i \in M, \ldots, B$), we no longer calculate or backpropagate the gradient for L_{Local}, since no reference embeddings are available for these datasets. For these scATAC-seq datasets, we still perform batch effect correction through the invariant representation learning objective (equation (3.2)), where the batch effect is represented as the confounding variable c, along with the read depth for each single cell.

3.2.4 Evaluations

For all methods, we project the input data to a lower-dimensional space (dimension of embedding is 50 by default, unless specified by other methods) that delineates the latent cell states. For Seurat, we use the scTransform function to normalize the raw counts and use the normalized data as input for PCA; for Signac, we use its multimodal integration analysis, which uses the same normalized gene expression data and additional TF-IDF transformed peak data as input; for SAILER we use the peak data as input; and for Cobolt and Schema, we follow their tutorials and use data from both modalities as input. To generate a lower-dimensional embedding for benchmarking, for Seurat, we use the top 50 PCs after PCA; for Signac, we use the results of Weighted Nearest Neighbor (WNN) analysis as a joint embedding of gene expression and chromatin accessibility modalities; for SAILERX, we extract the mean of the posterior latent distributions as the cell representation; for Cobolt, we use the latent variable z with dimension 50 calculated from its multimodal variational autoencoder; and for Schema, we use the 50-dimensional latent feature retrieved by using its fit_transform function. Other compared reference embeddings are generated with scVI [66], scANVI [116] and Scanaroma [45] using scIB package [67]. We set the default dimension as 50 for compared methods, including Seurat, Schema, Cobolt, SAILER, and Signac in our analysis in order to fairly compare all these works. As for the rest methods, we keep the default latent dimension settings in the scIB package for scVI, scANVI and Scanaroma (30, 30 and 100 respectively). 2D visualizations are acquired by running uniform manifold approximation and projection (UMAP) [72] on the latent embeddings.

One major task of these dimensional-reduction methods is to project the input genomics data to a lower dimensional embedding that is informative on cell type identification through clustering. To evaluate how the clustering generated from these embeddings are compared to the ground truth cell labels, we use quantitative metrics of Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), and Silhouette Score to assess the performance of different methods. ARI and NMI evaluate how well computational clusters overlap with ground truth labels, and the Silhouette coefficient evaluates the separation of the cell clusters. These metrics are common metrics used for benchmarking single cell clustering methods [118, 115]. Specifically, to generate cluster assignment for each cell, we construct k-nearest neighbor (KNN) graphs from the lower-dimensional embeddings of different methods respectively, and then apply the Louvain algorithm [104] to assign individual cells to different clusters. Each method generates its own set of clusters, and these clusters are then used to calculate quantitative metrics of ARI, NMI, and Silhouette Score for benchmarking. The calculations of metrics are carried out by functions from scikit-learn [80] library. For analysis in Figure 3.3 and 3.7, for fair comparisons, all methods are producing the same number of clusters. To determine the effect of clustering parameters and cluster numbers, we provide a wide range of resolutions and KNN numbers to the Louvain algorithm to determine the final clustering assignments. During the process, we record the number of clusters identified based on each combination of parameters (resolution and KNN number) for each experiment, as well as the metric scores for that clustering assignment. The effect of clustering parameters and cluster numbers are summarized in Figure 3.5.

3.2.5 Imputation

We generate the imputation data via a reconstruction conditioned on the invariant representation and fixed confounding factors. Specifically, we first push the raw data through the encoder network, and obtain the mean parameters for latent distributions. Unlike the training process, where we calculate the depth of the raw data and load the one-hot embedding according to the real batch information, here we fix the depth and batch indicator for reconstruction. As a result, we use only the invariant component z to reconstruct the chromatin landscape during the imputation process, while keeping the other confounding factors at a fixed level.

When evaluating imputation results, we first generate imputed data with each method. Then use randomized PCA to project the imputed data to a lower dimension. We then use UMAP to visualize the landscape of imputed data in 2D. For benchmarking against MAGIC, we use both graphs generated by scRNA-seq modality and scATAC-seq modality for fair comparison. The RNA graph is based on the Seurat embedding and ATAC graph is based on MAGIC's own pipeline. For benchmarking with scOpen, we follow the manual on its GitHub site to generate a dense imputation matrix. The imputed matrices are then subject to randomized PCA and visualized with UMAP. Quantitative scores (ARI, NMI, Silhouette Score) are calculated based on clustering results generated from the top PCs.

Table 3.2: List of cell-type specific marker genes used to visualize expressions.

Cell Type	Marker Gene Names
Pvalb	Erbb4, Cemip, Lrrc4c, Slit2, Cntnap4, Btbd11, Zfp536, Esrrg, Kcnc1,Cntnap5c
L4	Car10, Unc5d, Rorb, Pcdh15, Dcc, Gria4, Prkg1, Fstl4, Kcnh5, Cpne9
CD4 Naive	Bach2, Fhit, Igf1r, Ccr7, Ak5, Apba2, Lef1, Maml2, Sell,Satb1-as1
B Naive	Ighm, Ighd, Tcl1a, Bach2, Col19a1, Il4r, Skap1, Camk2D, Foxp1, Khdrbs2

3.2.6 Marker Gene Expression Analysis

To further evaluate the quality of cell clusters, we visualize the expression of marker genes in clusters labelled as CD4 naïve cells and B naïve cells from the PBMC dataset and L4 cells and Pvalb cells from the SNARE-seq dataset. To associate the cell clusters to biological cell types, the cell cluster labels are called based on a majority vote of the ground truth labels of the cells contained in each cluster. The four types of cells are chosen for this analysis because they are similar to other cell types and are challenging to cluster them. The CD4 cluster sits very close to CD8 naïve and other CD4 subtype clusters in the embedding space. The L4 cluster sits close to the L2/3 and L6 IT cell clusters. In particular, gene expression information alone cannot well separate subtypes of B cells.

The cell-type specific marker genes used for the visualization are called by the FindMarker function in Seurat [97]. These genes are identified as marker genes because they show significant differential RNA expression in the cells labeled with the corresponding cell types vs. other cells. Cell-type labels are based on ground-truth labels. The top 10 chosen marker genes associated with each cell type are shown in Table 3.2.

For each cell type, we use boxplots to visualize the mean normalized expression of marker genes (Figure 3.6) of the cells from the cluster labeled with the corresponding cell type. The gene expression values are normalized by scTransform, and the mean values are shown in Table 3.3, Pairwise t-tests between SALIERX and other methods indicate whether the marker genes from SALIERX show significantly higher expression than those from other methods. T-test p-values are indicated by ns (p-value > 0.05, i.e., not significant), * (p-value < 0.05), ** (p-value < 0.01) and *** (p-value < 0.001).

3.2.7 Motif Analysis

We perform motif analysis on several key motifs to demonstrate a case of discovering cell-type specific motif enrichment between different cell types. The putative cell types are determined by same procedures in previous section through clustering and majority vote. We first compute a per-cell motif activity score by running chromVAR [92]. It converts the peak by cell matrix to a motif by cell matrix, allowing us to get the motif activity score per cell, which provides an alternative method for identifying differentially active motifs between diverse cell types. In order to discover differential motif activities, we also utilize z-score, calculated by chromVAR, and FindMarkers function, provided by Signac [98], to get the average z-score differences between different cell types. Then these motifs are sorted according to their p-values. We set the parameter mean.fxn=”rowMeans” and fc.name=”avg_diff” in the FindMarkers function following Signac tutorial to compute the average difference in z-score in terms of fold-change calculation between the groups.

After that we apply MotifPlot to plot the 4 of the top 6 motifs that represents the most differential expressed motifs between the two cell types. Finally, we also get the clustering result with regards to specific cell types that we use to compare differential motifs. We use Louvain algorithm to assign a specific cluster number to each cell cluster, and then collect all the cells that belong to the same cluster number, which overlaps with the most cells of that specific ground truth cell type. We refer the z-score of those cells of that motif calculated from chromVAR and draw barplot to show the z-score distribution on that plot.

3.3 Results

Joint analysis of single cell multi-omics data with paired measurements often suffers from imbalanced SNR from different modalities [94]. In our study, we mainly focus on paired measurements of scRNA-seq data and scATAC-seq data. In practice, data from the scATAC-seq modality is often more affected by read depth variations and limited coverage rate, which would greatly impact the joint embedding when fusing data from two modalities together. In order to address the aforementioned issues, we design a framework SAILERX by using the structural similarity for the integration of the two-modality data and achieve satisfactory result. Here, we benchmark SAILERX with other methods that are able to cluster single/multi-modal single cell data. We also demonstrate that SAILERX could be used to align datasets with missing modality and improve analysis by applying joint analysis with a high-quality multimodal dataset. We include Table 3.1 to better illustrate the differences between our methods and others. After that, we further demonstrate the benefits of our method on downstream analysis such as motif discovery. Details are described in the following subsections.

3.3.1 SAILERX generates better clustering by fusing information from two modalities

We first benchmark our framework on PBMC 10k dataset, which consists of paired transcription and chromatin accessibility sequenced on 11,331 cells of human PBMC. This dataset is generated by 10X genomics. Some mature and differentiated blood cells from PBMC dataset have clear separation of cell types such as B cells and T cells. However, within those cell types, some sub-cell types such as monocytes are still ongoing differentiation process, resulting in continuously distributed cell clusters which often pose challenges to clustering algorithms.

During training, the regularization term in SAILERX encourages the local structure of the posterior distribution on the scATAC-seq data to be close to its scRNA-seq correspondence. The embedding from scATAC-seq is generated by the encoder network of a VAE, and the embedding for scRNA-seq modality for this dataset is generated by one of the scRNA-seq embedding methods. In this study, we mainly use PCA from Seurat as the scRNA-seq reference embedding, but other methods are also demonstrated in this dataset (Figure 3.3C). During training, we also assign a weight for each cell on this regularization term based on the read depth of two modalities (Methods). Cells with poor quality on scATAC-seq measurements will have higher weights. With this flexible weighting mechanism, cells with poor scATAC-seq measurements could get more information from its scRNA-seq correspondence, and cells with better data quality from scATAC-seq side could preserve their informative parts. After training, we retrieve the posterior mean of latent variable as our final embedding and cluster those cells accordingly. We benchmark our methods with three state-of-the-art (SOTA) methods that could handle multiomics data integration, i.e., Signac, Schema and Cobolt, as well as SOTA methods that only work on single modality data (i.e., Seurat, scVI, scANVI, Scanorama on scRNA-seq, and SAILER on scATAC-seq).

2-D visualizations of the embeddings generated by different methods are shown in Figure 3.3A, with cells colored by ground truth cell type labels. The ground truth cell type labels are inferred through Seurat-style mapping strategy from [42]. We validate these ground truth cell type labels by visualizing some enriched expressions of known cell type-specific marker genes (Figure 3.4), such as pDC cells (with known marker genes CLEC4C and NRP1) [24, 93, 70, 85], and Treg cells (with known marker gene FOXP3 and RTKN2) [90, 5]. From the results, we can see that the ground truth cell types here correspond well with the well-known cell-type markers, so we consider these labels as “ground truth” labels for the following analyses.

To quantitatively assess these clustering methods, we use ARI, NMI, and Silhouette metrics

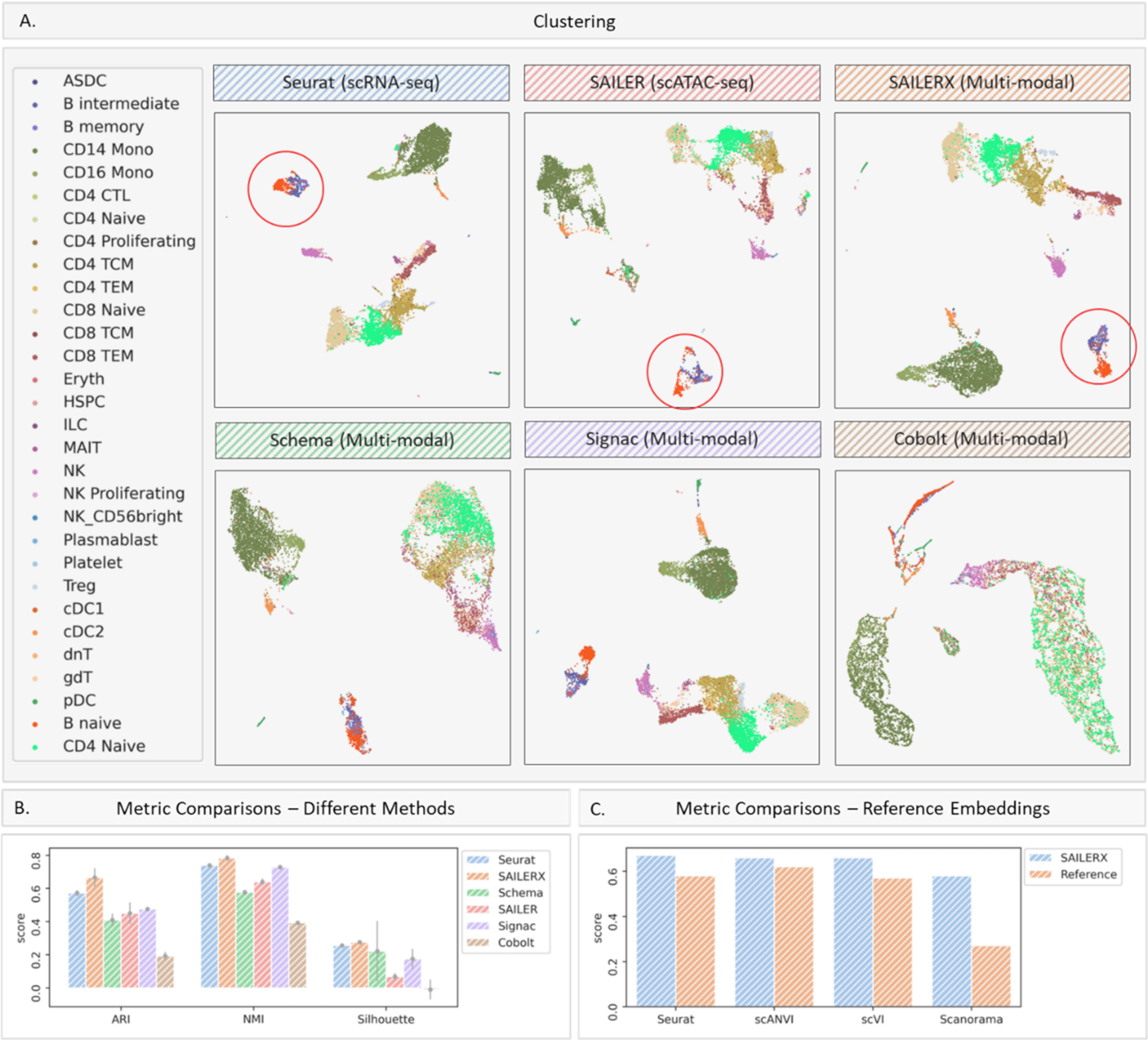


Figure 3.3: Results on PBMC 10k Multiome dataset. Cells colored by ground truth label. (A) UMAP visualizations of embeddings on PBMC 10k Multiome dataset generated by different methods. Red circles show separation of sub clusters of B cells under Seurat (scRNA-seq only), SAILER (scATAC-seq only) and SAILERX (multimodal). (B) Quantitative metrics of ARI, NMI, and Silhouette Score on clustering generated by different methods. Error bars are generated by repeating experiments with 90% randomly subsampling. (C) Quantitative metrics of ARI Score on Reference Embeddings on gene expression modality and Integrated Embeddings generated by SAILERX.

to evaluate the clustering results. ARI and NMI evaluate how well the computational clusters derived from lower-dimensional embeddings overlap with ground truth cell labels; and the Silhouette coefficient measures the separation of the cell clusters in the embedding space.

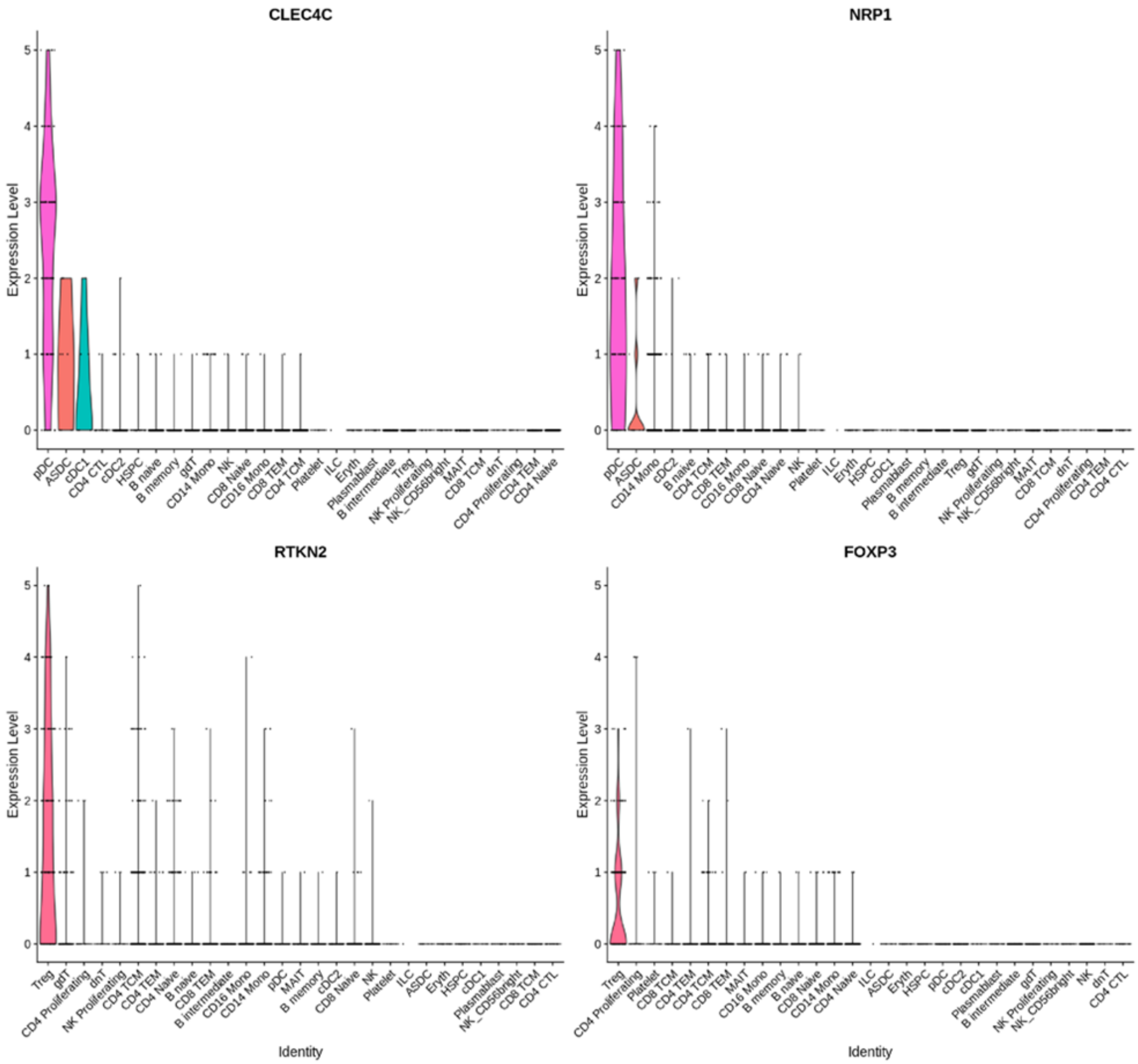


Figure 3.4: Visualizations of marker gene expressions by inferred ground truth cell types in the PBMC 10k Dataset. CLEC4C and NRP1 are marker genes for pDC cells; RTKN2 and FOXP3 are marker genes for Treg cells.

Higher scores indicate better matchings and separations. The metric scores are shown in Figure 3.3CB-C and Figure 3.5C, with SAILERX achieving the highest scores in ARI, NMI, and Silhouette coefficient. From the scores, we can see Seurat achieves a great performance on overall clustering results. In the figure, we can see it forms tight and separable clusters for most cell types. Some other multimodal integration methods do not perform as well as Seurat when adding extra information from chromatin accessibility, showing that adding ex-

Table 3.3: Mean expressions of markers on cells clustered by different methods.

Cell Type	SAILERX	Seurat	Signac	Cobolt	Schema	SAILER
Pvalb	**7.10**	4.25	7.07	0.72	5.01	3.71
L4	1.05	**1.11**	0.97	-0.01	0.92	0.77
B naive	**6.29**	3.60	6.09	3.55	3.68	3.60
CD4	**1.36**	1.34	1.34	1.05	1.07	1.23

tra information without properly handling the noise could harm the overall clustering result. However, when we compare SAILERX with Seurat, we can see the embedding generated by SAILERX keeps the robust separation of cell clusters inherited from its reference gene expression modality, while preserves the useful signals appearing in the chromatin accessibility modality. This could also be demonstrated by the separation of sub clusters of B cells colored in red and blue (Figure 3.3A red circles), and the higher marker gene expressions for cells identified as B naïve cells (Figure 3.6A, Table 3.3). This shows that through proper integration of information from both modalities, SAILERX could discover new (sub)types of cells previously unidentifiable with gene expression modality only. Also, from the results, we can see that our integration benefits the delineation of continuously distributed cell types, e.g., CD4 cells. CD4 cells are previously reported to be more identifiable using chromatin accessibility information [88]. This can be demonstrated when we try to identify subtypes of CD4 cell. Compared with other methods, CD4 naïve cells identified by SAILERX have higher marker gene expressions (Table 3.3). This shows our cross-modality integration can also benefit the cell type identifications for ambiguous subtypes.

For robustness evaluation, we further test if our method could consistently improve upon different reference embeddings. Here we use three other scRNA-seq embedding methods (scVI, scANVI, and Scanorama) to generate reference embeddings and then use these embeddings to help train SAILERX models. As shown in the FFigure 3.3C and Figure 3.8, the joint embeddings combine information from two modalities and constantly outperform their reference embeddings. This shows effectiveness and robustness of SAILERX's information fusing strategy.

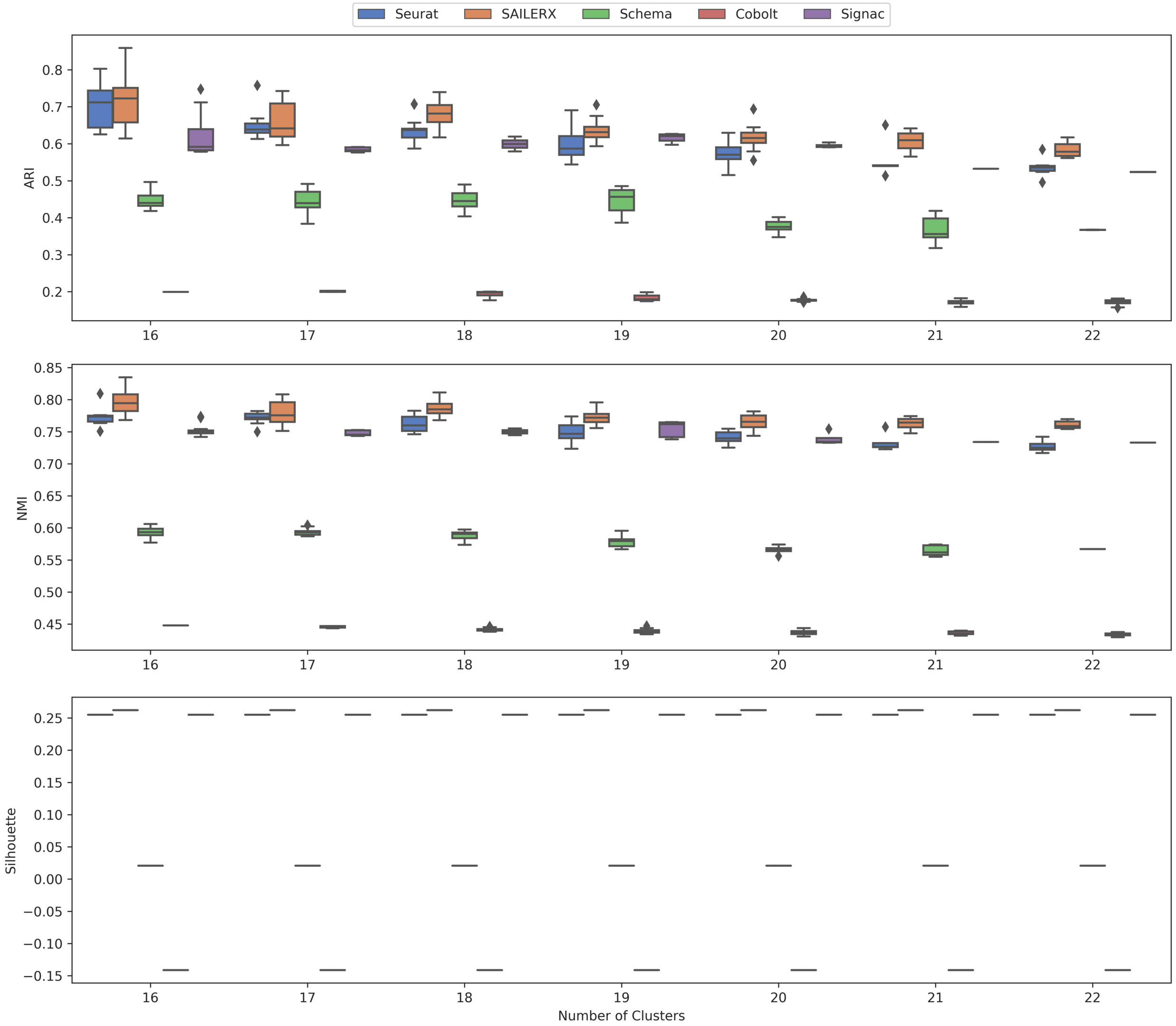


Figure 3.5: Clustering scores of PBMC 10k dataset by different number of identified clusters.

Similar analyses are performed on the SNARE-seq dataset [20] with a different sequencing technology. SNARE-seq data are from mouse brain tissue. A great majority of cells in this dataset are found in a quiescent state, and thus is more stable compared with PBMC cells. Compared with PBMC 10K from 10X genomics, the SNARE-seq data tends to have much shallower read depth in chromatin accessibility reads, which makes this chromatin accessibility data here sparser than the scATAC-seq data in the previous analysis. From the results (Figure 3.7), we can see some integration methods severely suffer from this when projecting data from two modalities into one shared latent space. In this scenario, embedding gener-

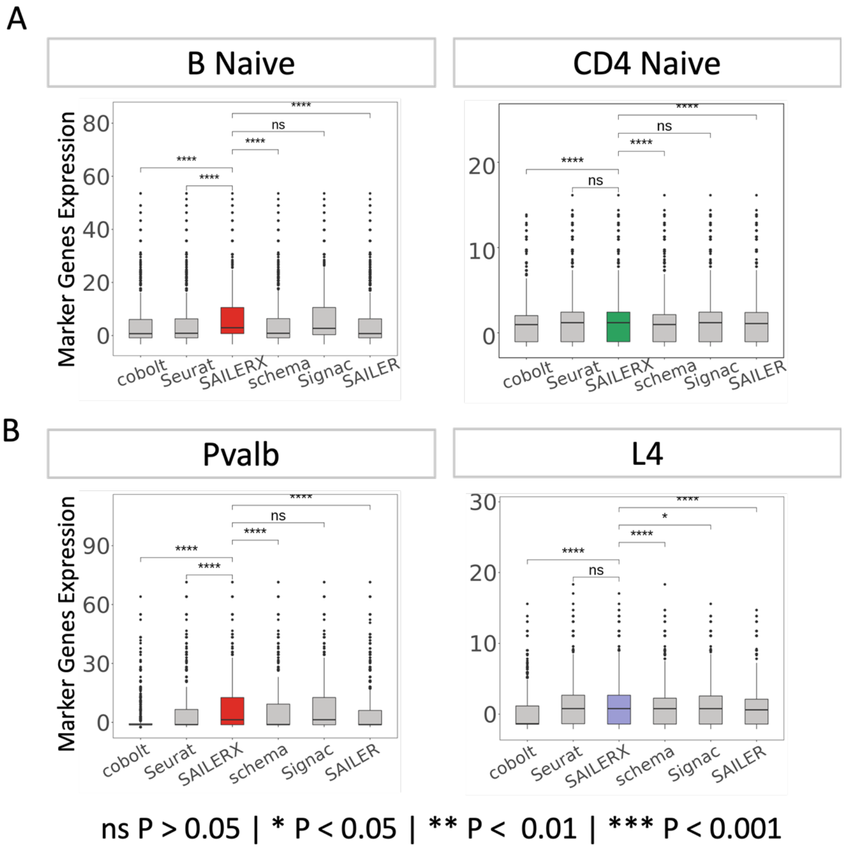


Figure 3.6: Comparing the expression of marker genes in clusters derived by different methods. (A) Mean expression of marker genes of B Naive cells and CD4 Naive cells from PBMC 10k dataset. (B) Mean expression of marker genes of Pvalb cells and L4 cells from SNARE-seq dataset.

ated by SAILERX forms tighter clusters (Figure 3.7A) and achieves the best performance in terms of quantitative results (Figure 3.7B). The separation of cell types is also demonstrated by marker gene expressions of cells identified by different methods (Figure 3.6B, Table 3.3), where SAILERX shows higher results compared with other methods.

We also perform clustering analyses on a more recent Share-seq dataset [69] on mouse skin tissues. The results are shown in Figure 3.9, where SAILERX achieves better results in terms of quantitative scores. Among all different types of tissues and sequencing technologies, the integration strategy used by SAILERX robustly outperforms other methods, showing the effectiveness of our framework.

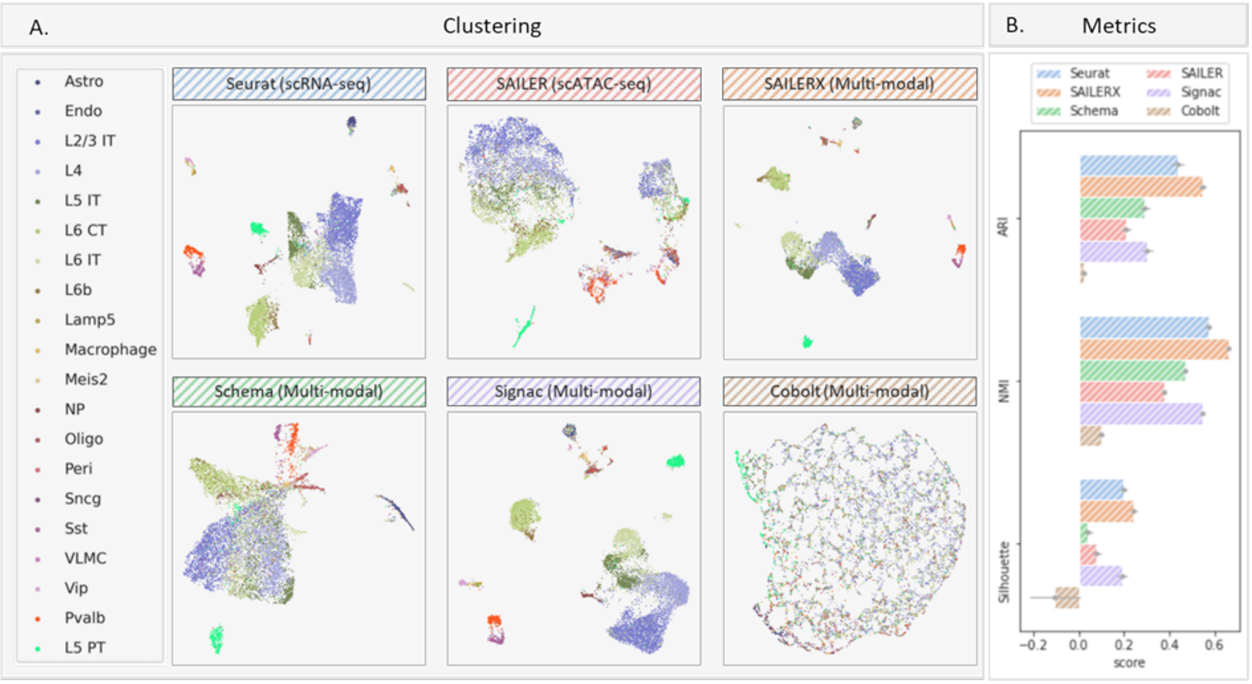


Figure 3.7: Results on SNARE-seq dataset. (A) UMAP visualizations of embeddings generated by different methods on SNARE-seq dataset. Cells are colored by ground truth labels. (B) Quantitative metrics of ARI, NMI, and Silhouette Score on clustering generated by different methods. Error bars are generated by repeating experiments with 90% subsampling.

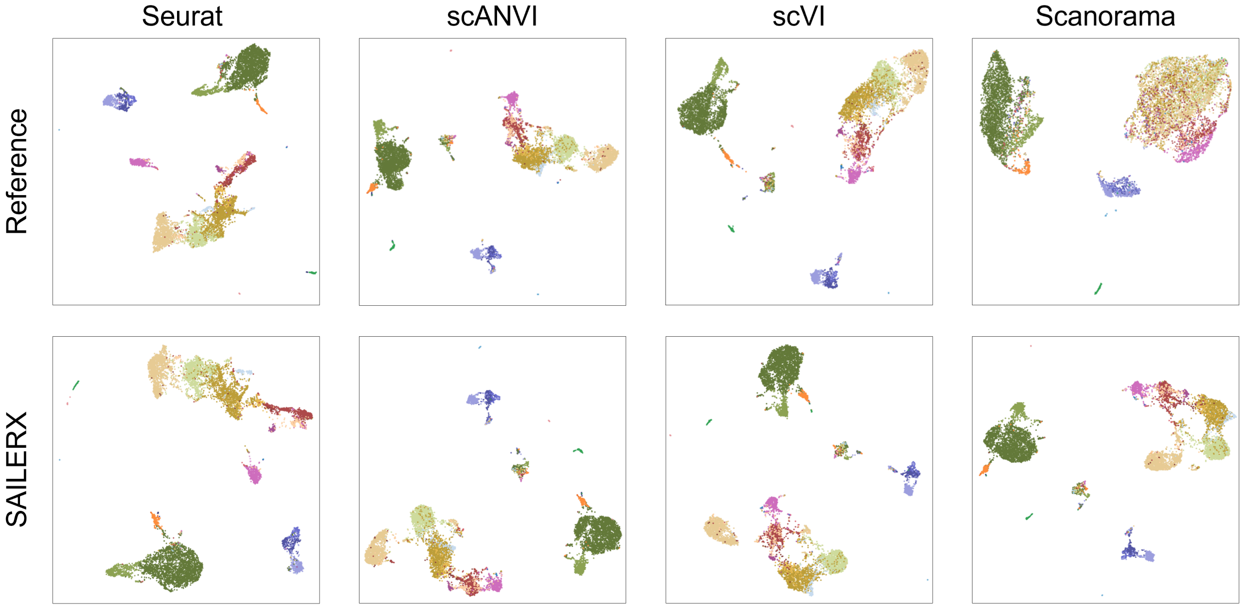


Figure 3.8: UMAP Visualizations of reference embeddings vs SAILERX embeddings. Top row: UMAP visualizations of reference gene expression embeddings generated by different methods. Bottom row: joint embeddings generated by SAILERX after training.

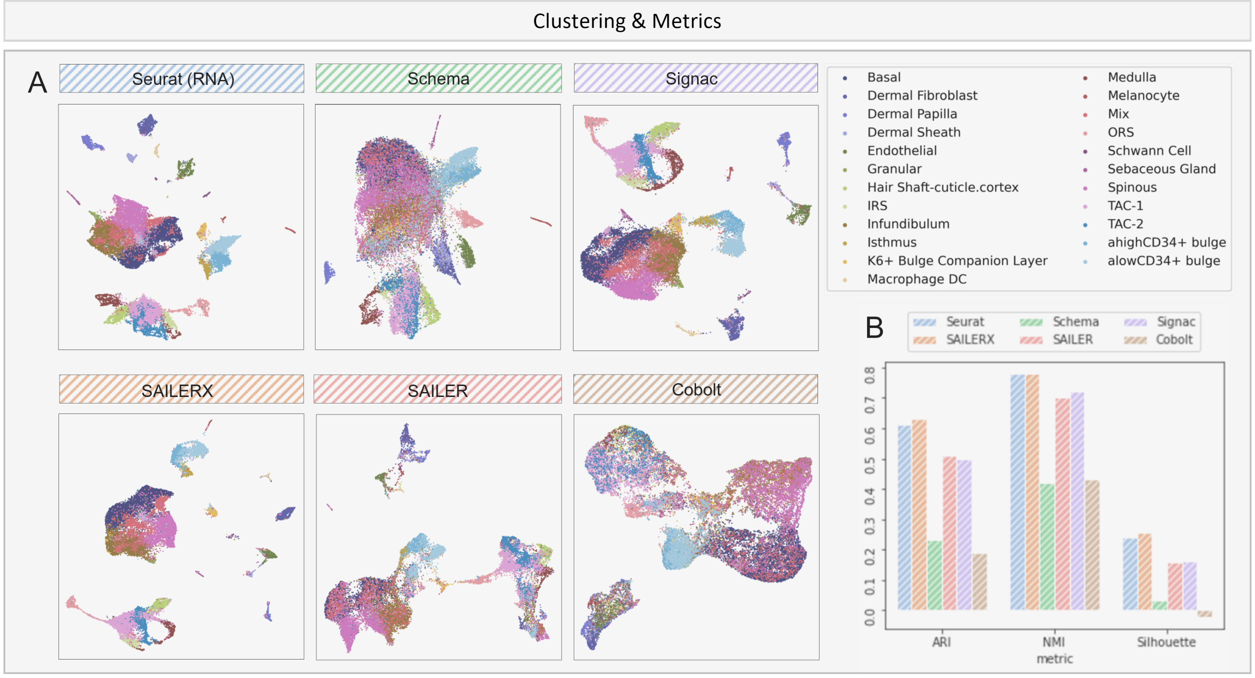


Figure 3.9: Results on Share-seq dataset. Cells colored by ground truth label. (A) UMAP visualizations of embeddings on mouse skin Share-seq dataset generated by different methods. (B) Quantitative metrics of ARI, NMI and Silhouette Score on clustering.

3.3.2 SAILERX improves analysis of single modal scATAC-seq dataset by aligning it to multimodal datasets

Besides fusing information from two modalities within one dataset, SAILERX is also capable of performing multi-sample data alignment even for datasets with missing modalities. This is achieved by the invariant representation learning objective of our framework. By assigning a batch indicator variable as a confounding factor, the model automatically corrects for the batch effect during training. When integrating datasets with missing modalities, we ignore the regularization term for those cells with only one type of measurements. For this case, we use PBMC 10k Multiome dataset with paired scRNA-seq and scATAC-seq measurements, together with a single-modal PBMC 3k dataset with scATAC-seq only as described in Methods. Two datasets are jointly trained as described above. We then obtain the latent representation and perform clustering on cells from PBMC 3k dataset using

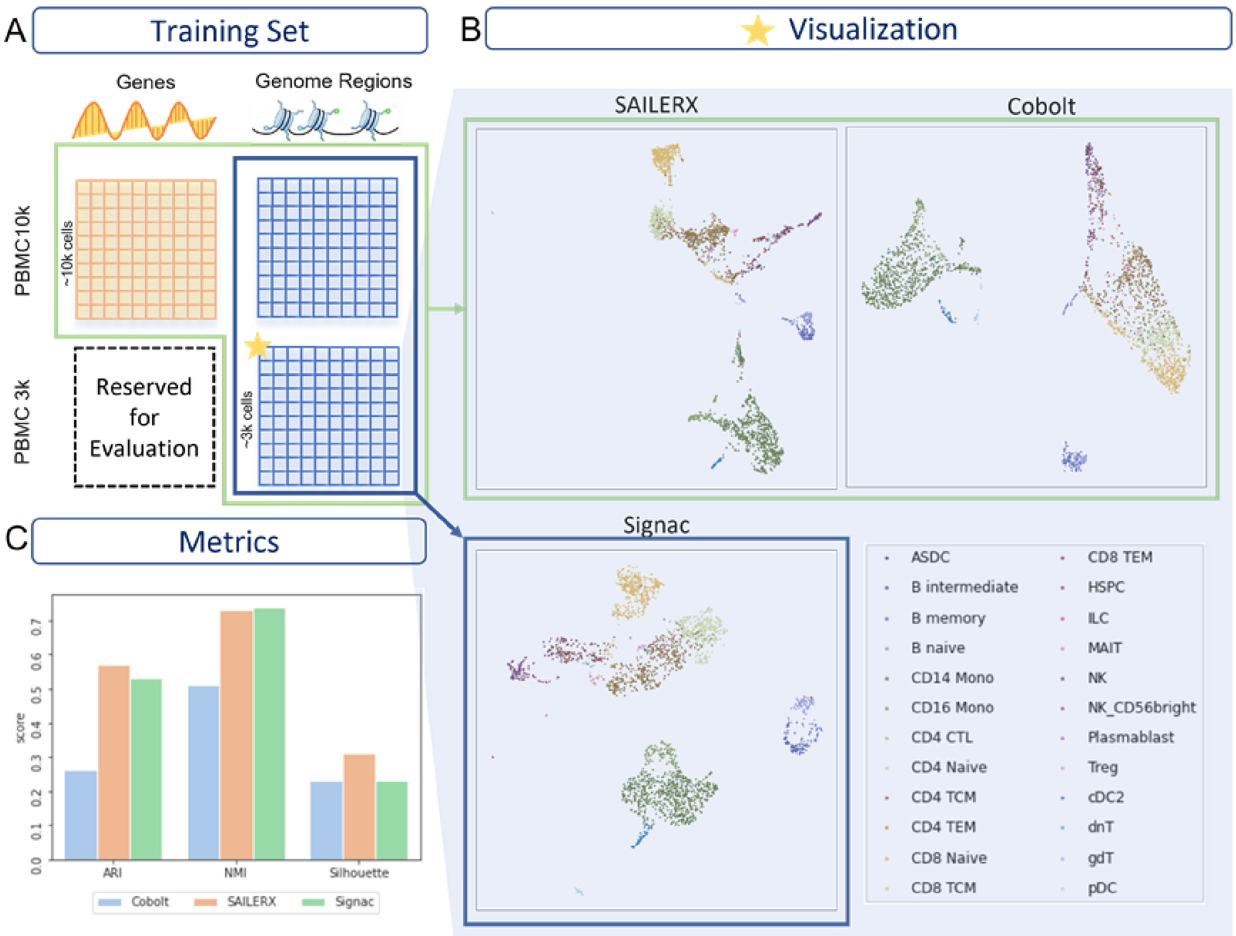


Figure 3.10: Hybrid training result on PBMC 3k dataset. (A) Datasets used for training. (B) UMAP visualizations of PBMC 3k dataset. (C) Metrics on clustering for PBMC 3k dataset.

Louvain community detection. The results are shown in Figure 3.10, and ground truth cell types are identified by marker genes as in Hao et al [42]. Here we evaluate the clustering metric, and compare it with Cobolt [37], which is also capable of integrating multimodal data with missing modality, and Signac, which only performs integration with scATAC-seq modalities. The Cobolt method adopts a multimodal VAE with shared latent space. As shown in Figure 3.10 B and C, SAILERX achieves the best clustering metrics, showing that the flexible fusing mechanism works better on the noisy single cell multiomics data compared with Cobolt, and the single modal data with lower data quality could benefits a lot from this type of multi-sample alignment.

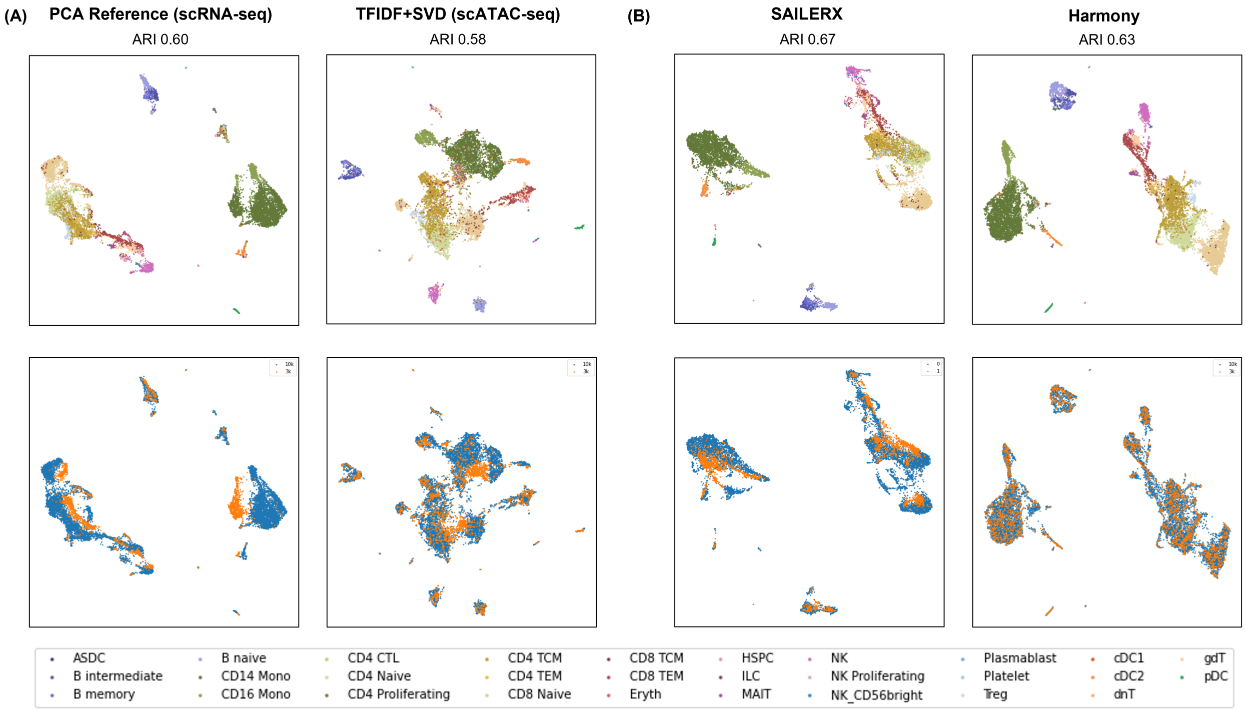


Figure 3.11: Results on batch effect correction on PBMC 10k and 3k datasets. (A) UMAP Visualizations of PCA (left) embedding on gene expression modality and TF-IDF + SVD (right) embedding on chromatin accessibility modality before batch effect corrections. (B) UMAP visualization of embeddings after batch effect correction. Top row: colored by cell types; Bottom row: colored by batches.

In addition to batch alignment between one multi-modal and one single modal dataset, SAILERX could also align data from multiple multimodal datasets. We demonstrate this with complete PBMC 3k and 10k datasets. As shown in Figure 3.11, SAILERX could align data from different batches when there exists a clear batch effect while preserving a high quality of clustering results. And in Figure 3.12, SAILERX is trained in a situation with cell type heterogeneity: we mimic this by dropping one unique cell type from each batch. When these data are processed together for batch alignment, we find that the unique cell clusters are preserved. This shows that SAILERX can preserve biological signals when performing batch effect corrections.

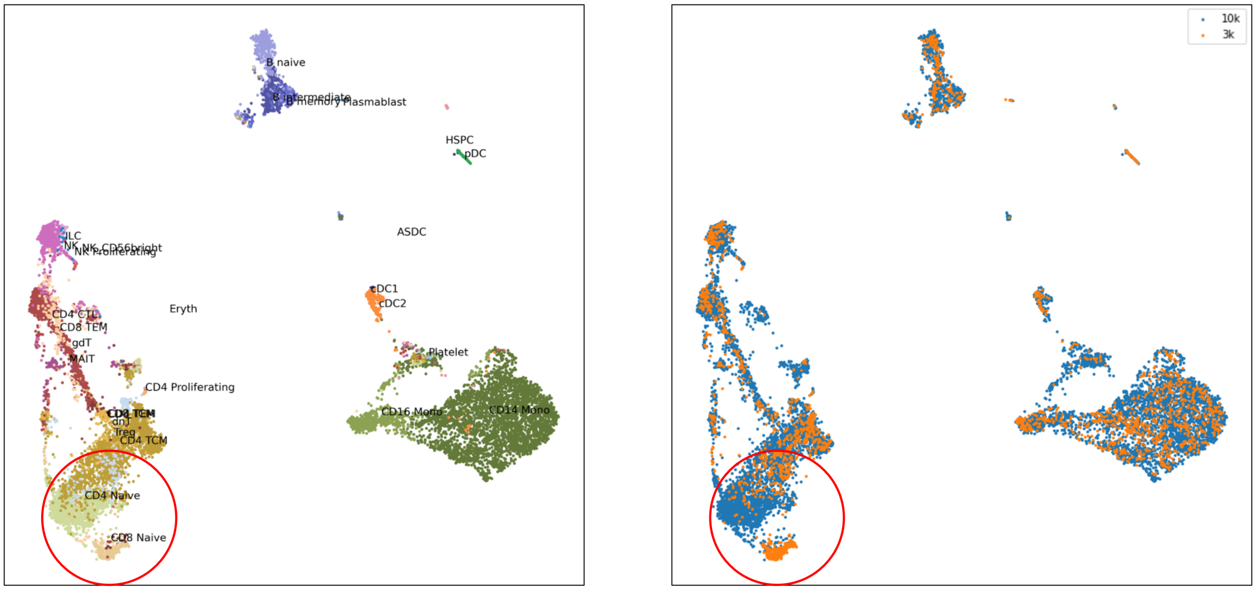


Figure 3.12: UMAP visualizations of the embedding generated by SAILERX. Left: colored by cell types; Right: colored by batches.

3.3.3 Cross modality integration facilitates downstream analysis of chromatin accessibility data

In previous sections we have demonstrated that SAILERX is able to generate better embeddings under different scenarios. Here, we explore how this advantage could benefit downstream analysis of chromatin accessibility data. Here, we perform motif enrichment and motif activity analysis on the SNARE-seq data mentioned above, which suffers more from the sparsity and dropouts on the chromatin accessibility signals.

We first perform differential testing using the chromVAR [92] deviation z-score as described in Methods. Here we use Pvalb and Sst cells (colored in red and purple in Figure 3.7A) to calculate the differential motifs between these two cell types. Then we plot the top 6 motifs that are mostly enriched between the two cell types by p-value calculated by FindMarkers function from Seurat. As shown in Figure 3.13. Mef-family motifs are greatly enriched in Pvalb-specific peaks in scATAC-seq data, with 4 out of 6 Mef-family motifs enriched in those Pvalb-specific regions. These findings are consistent with previous reports [97, 33].

Moreover, the Mef2c motif is also reported to be involved in the development of Pvalb interneurons [71], and also shown enriched as one of the differential motifs (Figure 3.13, Figure 3.14). To quantify the performance of these enriched motifs, we select those groups of cells from clustering results of each method, which most likely represent Pvalb cells, and then we calculate the value of z-score within those cells (details in Methods). We compare the results generated by five other methods that are able to integrate multimodal scRNA-seq and scATAC-seq data or work only on scATAC-seq modality. As shown in Figure 3.13, our method achieves the highest value of motif deviation z-score among all the methods with the differential significance of pairwise t-test p-values all less than 0.05, showing that SAILERX is more likely to discover novel motifs based on this clustering. In addition, we compare L4 and L5 PT cells and compute the enriched motifs between those cells. Previous reports claim that POU3F2 protein associates with bipolar disorder and is involved in the neocortex development in mice [18]. From the top 6 enriched motifs we could find, there are several POU family related motifs enriched in the cells including POU3F2. Therefore, we explore the motif enrichment results on L5 PT cells using POU1F1 and POU3F2 motif deviation z-score calculated by chromVAR. Results are shown in Figure 3.13B. We find that SAILERX still achieves the highest motif deviation z-score, further demonstrates the effectiveness of our method on facilitating downstream analysis of chromatin accessibility data.

3.3.4 SAILERX recovers the cell type landscape in chromatin accessibility space through imputation

The high throughput of sc-seq measurements provides expressions and chromatin accessibility information at the finest resolution. However, due to the limitations of read depth and coverage, sc-seq data suffers from severe sparsity due to random dropouts during the sequencing stage. Imputation is often applied during data analysis to recover the missing values. Here we test how our methods denoise the raw scATAC-seq data after integrating

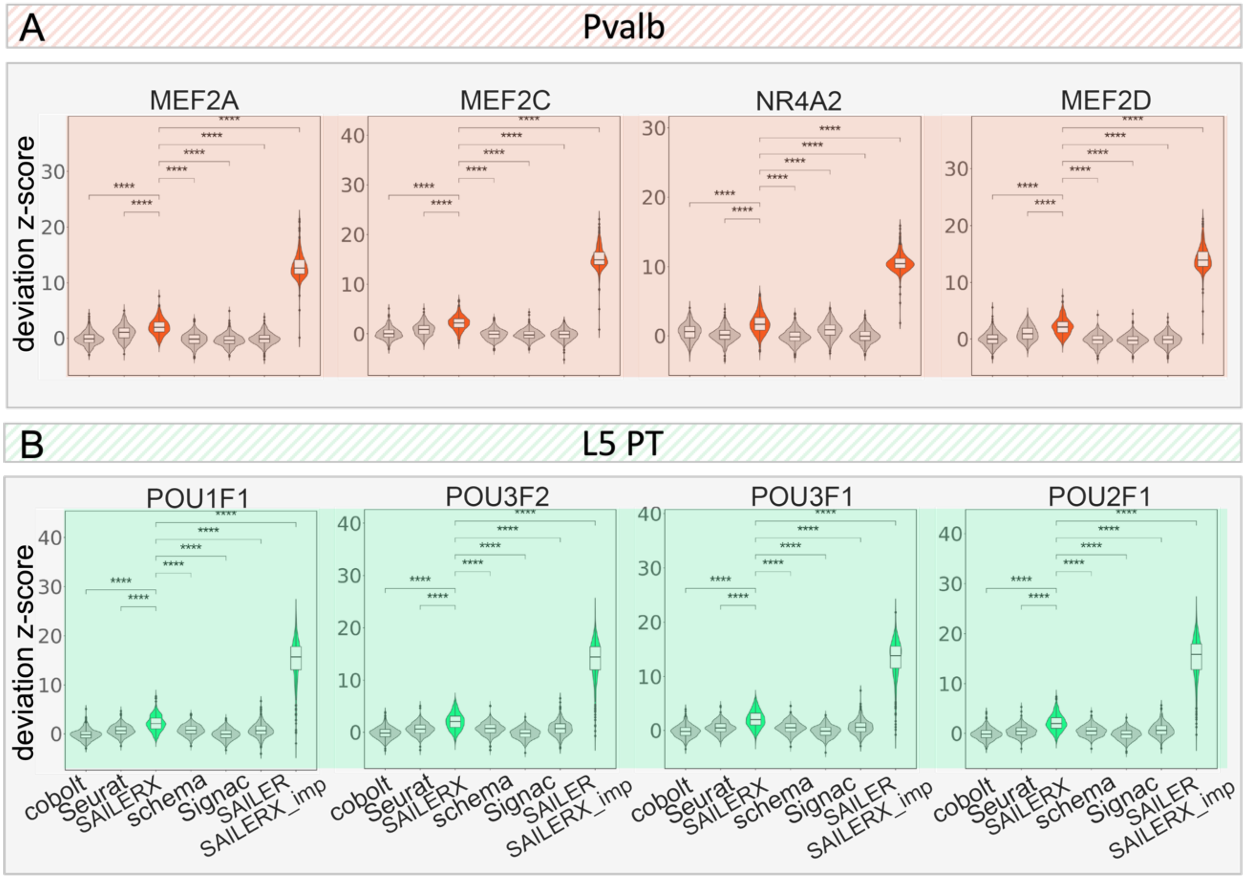


Figure 3.13: Motif enrichment scores. Motif deviation z-scores on cells identified as (A) Pvalb and (B) L5 PT by different methods from SNARE-seq dataset and the imputed dataset (imputation done by SAILERX). For each cell type, four enriched motifs are selected. Pairwise t-tests are performed between SAILERX and all other methods. Three-stars refers to differential significance between two methods (p-value less than 0.05).

information from the scRNA-seq modality. We benchmark against MAGIC [28], which utilizes data diffusion to perform data imputation, and scOpen which is a matrix factorization based method.

Here, imputed data is generated by SAILERX, MAGIC, and scOpen respectively. For MAGIC, since one key factor for imputation quality is the neighborhood graph, we provide graphs generated by scRNA-seq and scATAC-seq to MAGIC (details in Methods), and show the visualizations of imputation results in Figure 3.15. As we can see, compared with MAGIC and scOpen, imputed data generated by SAILERX better preserves the cell type

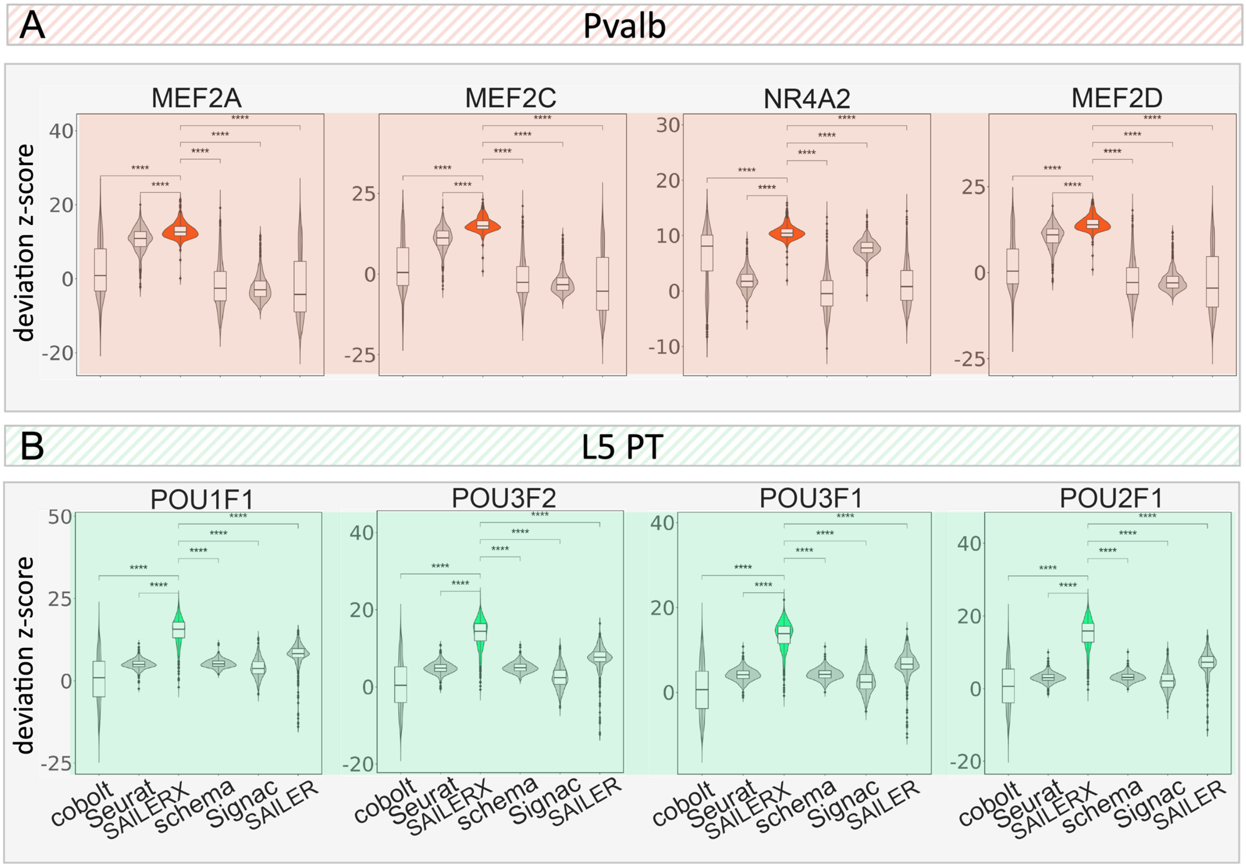


Figure 3.14: Motif deviation z-scores on cells identified as (A) Pvalb and (B) L5 PT by different methods from SNARE-seq imputed data. The data is imputed through SAILERX. For each cell type, four enriched motifs are selected. Pairwise t-tests are performed between SAILERX and all other methods. Three-stars refers to differential significance between two methods (p-value less than 0.05).

landscape, where cells of different types are forming distinct clusters. Since SAILERX can control the read depth at imputation stage, imputed data is free of these technical artifacts. Compared with other imputation strategies, imputation done by deep generative models better preserves the cell clusters and keeps distinct features of cells. To further validate the imputation result, we use imputed SNARE-seq data generated by SAILERX and redo the motif enrichment analysis on Pvalb and L5 PT cells (previous section). Motif deviation z-scores are visualized with violin plots as shown in Figure 3.13 (see SAILERX_imp column). From the results, we can see that data imputed by SAILERX shows significantly higher enrichment score, which indicates that some missing peaks are imputed for certain

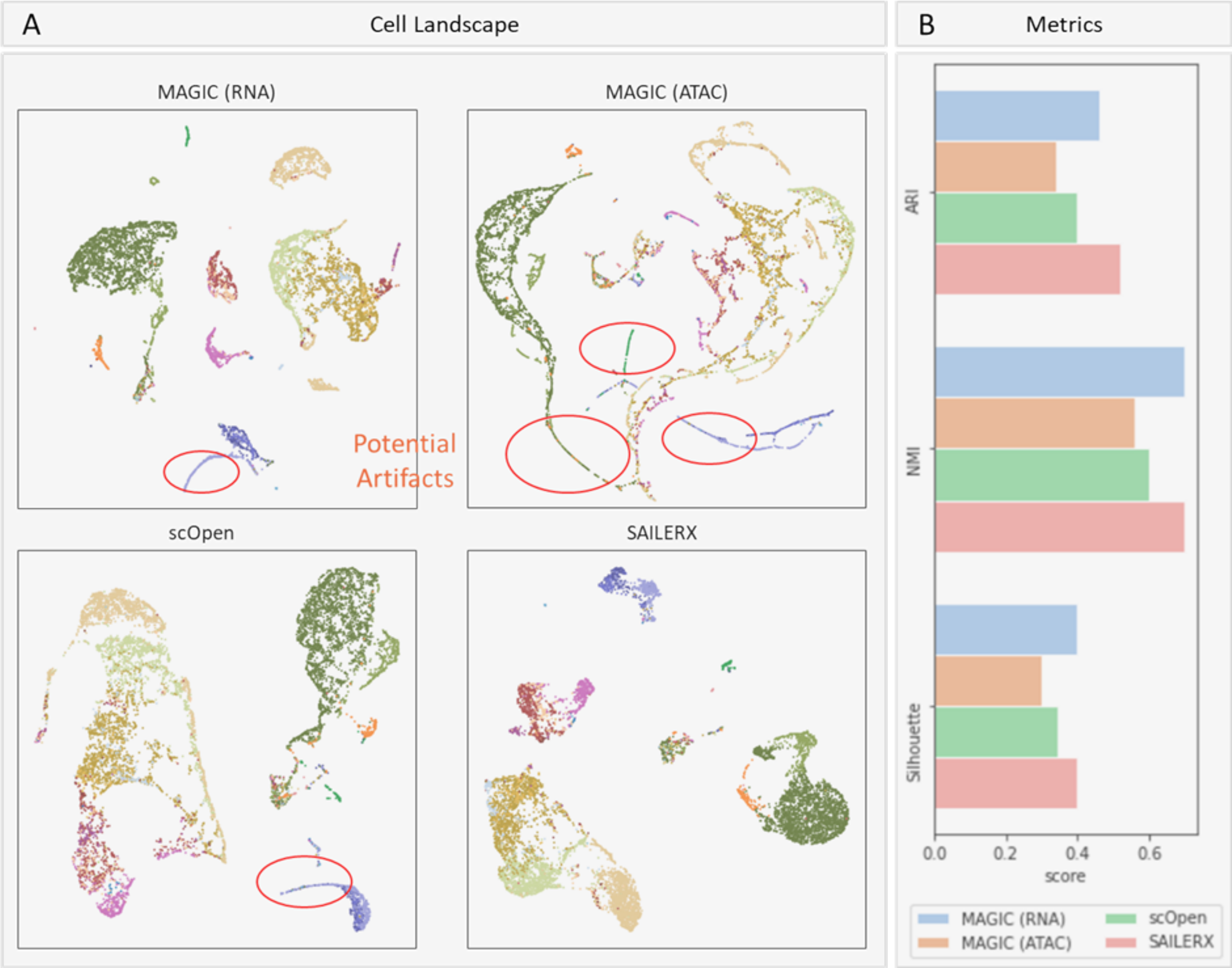


Figure 3.15: Results of imputations on PBMC 10k. (A) UMAP visualizations of imputed 10x Genomics PBMC chromatin accessibility data generated by SAILERX, scOpen, and MAGIC (MAGIC imputations are done with graphs generated by scRNA-seq and scATAC-seq respectively). Cells are colored by ground truth labels. (B) Quantitative metrics on the cell landscape.

cell types.

3.4 Discussion

Multimodal single cell data provides a more comprehensive way of measuring cell manifold. However, it is computationally challenging to leverage these multiomics data to better depict the biological view of cell-cell specificity still poses challenges for researchers due to imbalanced SNR cross modalities. Some modalities in nature have lower coverage rate, thus suffers more from noises like dropout. Current methods often fuse these multimodal data by projecting them to a same latent space [37, 94, 98]. These approaches assume measurements from two modalities have the same distribution, and both modalities are equally informative on cell state information. In reality, these assumptions barely hold because chromatin accessibility changes usually prior to the changes of gene expression states [69]; and scATAC-seq measurements tend to suffer more from sparsity but could potentially provide more detailed information on cell states. In the meantime, since there exist technical noises during sequencing process, which could bias the observed state of a cell toward different directions, projecting the observed data from different modalities to a same point could be problematic. Experiments have shown that projecting two modalities to a shared latent space could result in overfitting of noises and lead to worse delineation of cell state landscape, especially when using powerful models like neural networks [94].

To tackle these issues, in SAILERX, we use a more stable way by representing the more robust gene expression modality as a reference embedding, and guide the inference of a VAE modeling chromatin accessibility data. Instead of regularizing the latent variable for different modalities to be the same, we encourage the pairwise distances between cells to be similar across different modalities, in the meantime, use invariant representation learning to remove technical noises that are observable at training time. This flexible information fusing framework encourages the local structure of data to be similar and weights cells differently to better retrieve information from heterogeneous modalities. According to our results, this type of information fusion is able to preserve the informative parts from both modalities and constantly achieves better embeddings and downstream analysis. The final clustering results implicitly contain information from two modalities and can constantly improve upon any single modalities. SAILERX could also be used on dataset with missing reference modality. This allows SAILERX to be used under more scenarios (when datapoints from reference modality is missing during QC or analyzing a dataset with different sequencing protocols), using multimodal single cell data as a reference to facilitate the analysis of scATAC-seq data which usually suffers from low signal-to-noise ratio. With the help of SAILERX, researchers could rescue those low-quality single modality data through hybrid data integration and discover more informative features underneath those noises.

Chapter 4

Learning Semantic Representation of Genes through Large Scale Pretraining on Single-cell Sequencing Data

Recent advancements in single-cell technologies have ushered in an era of unprecedented data generation, yielding atlas-level datasets encompassing millions of individual cells. These collective datasets offer a rich tapestry of insights into the intricate gene interactions underpinning the diverse functionalities of distinct cell types. In this study, we introduce "Expresser" (short for Expression Transformer), a foundational model designed for effective pretraining on large-scale single-cell RNA sequencing (scRNA-seq) datasets. Our primary objective is to harness the model's capacity to distill meaningful quantitative annotations of genes, thereby empowering a multitude of downstream tasks. Expresser was pretrained with self-supervision on a diverse spectrum of single-cell datasets, comprising approximately 6 million cells. To evaluate its efficacy, we subject the model to fine-tuning on various downstream tasks. Our experimental results corroborate the proficiency of the pretrained Expresser model in several key aspects. Firstly, it excels in accurately imputing missing expression values, a vital operation in scRNA-seq data analysis. Furthermore, the semantic gene representations acquired through pretraining prove to be highly transferable across a diverse spectrum of downstream tasks. These tasks encompass predicting gene interactions, quantifying loss-of-function scores, assessing dose sensitivity, and inferring protein-protein interactions. In summation, our study illuminates the advantages of pretraining expansive foundational models on diverse scRNA-seq datasets, providing meaningful gene representations and facilitating their utility for quantitative gene annotation.

4.1 Introduction

The human body is made up of 37 trillion cells, each with their own composition and function. Genes are the basic building blocks of cellular molecular systems. Even though researchers have almost identified the entire transcriptome, the fundamental rules of how genes interact with each other and give rise to the function of cells have yet to be fully understood. To solve this puzzle, we need sufficient experimental data to characterize the gene expression patterns and functions of cells systemically and comprehensively, as well as innovative computational methods that take advantage of these large-scale datasets, and learn fundamental knowledge of genes that are transferrable under different scenarios.

From the data perspective, recent advances in single cell sequencing technologies offer genome-wide measurements of genetic information from individual cells and have produced a number of large-scale reference data to characterize the complexity and diversity of human cells. Specifically, single-cell RNA sequencing (scRNA-seq) provides quantitative measures of the expressions of all genes in single cells, up to millions of cells in one experiment. Developments in single cell technologies have led to projects like Human Cell Atlas and others

[84, 89, 14, 51], providing comprehensive measurements with whole transcriptome data from millions of single cells.

On the computational side, recent research demonstrates that pretraining large models on broad data enable the models to learn fundamental features from data, which allow them to be adapted to a wide range of downstream applications with little finetuning while achieving state-of-the-art performance. These models are referred as foundation models or Large Language Models (LLMs) in natural language processing. Foundation models have recently transformed how machine learning systems are built in both computer vision and natural language community. Examples include large-scale language models such as BERT [27], GPT [10], and PaLM [23], vision models such as ResNet [44], Inception [99], and MAE [43], and multimodal models such as CLIP [82] and DALL-E [83]. The rise of foundation models has significantly lowered the barrier of building AI models for downstream applications, with users focusing on adapting and transferring knowledge from these models to new applications, instead of building new models from scratch. Here comes an interesting question: how researchers could fully utilize the potential of abundant single cell datasets to decipher complex and diverse patterns of interactions between genomic elements, and then to produce novel annotations of genes that could be finetuned for many different downstream purposes?

During the past few years, many computational methods have been developed for analyzing single cell sequencing data [68, 67, 105, 118, 46]. The majority of these methods aims to project the data to a lower dimensional manifold [75], and then conduct clustering [66, 29, 2, 113, 17, 86], batch effect correction [39, 57, 13], and imputation [63, 47, 108] accordingly. However, most methods focus on learning embeddings that characterize the similarity between cells, and this could be confounded with batch effect and random dropouts from the sequencing process. In the meantime, in order to achieve best performance under these noises, some methods can only work with a subset of highly variable genes instead of the entire transcriptome. These aspects hinder the methods of this category from effectively learning from multiple large datasets.

More recently, there are also efforts on pretraining LLMs with single cell data. scBERT [117] trained BERT-based transformer encoder models and demonstrated its utilities in annotating cell types. Geneformer [103] uses masked language modeling pretext to pretrain BERT transformers on scRNA-seq data ranked by expression levels to generate contextualized gene embeddings that could be finetuned for other downstream tasks including predicting functional annotations of genes related to network biology. These two methods showed the benefits of LLMs pretrained on single cell, however, both of them rely primarily on model structures originally designed for language modelling; and their modeling choices with BERT structure limited the models' potential for learning informative quantitative correlations between genes.

Despite the tremendous progress that have been made, current methods do not fully harness the potential of large-scale single cell datasets. To address these challenges in modeling single cell data, in this study, we proposed Expression Transformer, a modified transformer encoder-decoder architecture for pretraining foundation models on single cell data to learn fundamental features of genes that are transferable among datasets. We collect data from multiple atlas level datasets generated by different sequencing technologies for pretraining the foundation model. In order to facilitate the model to learn meaningful embeddings of genes, we design a novel decoder architecture that learns interactions between genes and directly regresses the quantitative expression values under a masked prediction pretext. During training, we focus on modeling non-zero expression values only, since it carries more reliable signal and prevents the model from overfitting the noises. To demonstrate the effectiveness of our model architecture and the utility of the embeddings, we benchmark on downstream tasks including imputing missing values from scRNA-seq experiments, predicting protein interactions, predicting gene loss of function score, and prediction tasks related to network biology. In summary, we explore novel model designs that could be used to pretrain foundation models in single cells, and demonstrate how pretrained semantic embeddings of genes could benefit downstream tasks while minimizing the computation that needs to be carried out by end users.

4.2 Materials and Methods

4.2.1 Datasets and Preprocessing

Datasets

In order to demonstrate the effectiveness of foundation models in single cell, we constructed a training dataset with 6M single cells from multiple tissues and donors. Datasets used here are downloaded from the Human Cell Atlas[84, 89, 14, 51] website. Multiple datasets are aligned to the same dimension of genes and concatenated into one memory mapping array for effective training with deep foundation models. We kept all the pro-tein coding genes with uniquely identifiable gene symbols, which resulted in 18,483 genes used in training and benchmarking. Given that our training objective only focuses on entries with nonzero values, datasets with missing genes are filled with zeros. Genes are then modeled with differentiable tokens as part of model parameters. For dataset with m genes, the tokens can be represented as

$$T_g^{m\times d} = [t_1^d, t_2^d, ..., t_m^d] \tag{4.1}$$

where $t_j \in \mathbb{R}^d$ represents the embedding vector with dimension d corresponding to the j^{th} gene in the dataset. Gene embedding tokens T_g are implemented with the Pytorch embedding module with dimension d equals to 200 throughout most of our experiments unless specified elsewhere. The gene embedding tokens are randomly initialized at the beginning of training and optimized during the training process.

Data Preprocessing

Data preprocessing involves standard normalizations for scRNA-seq data and a binning process mapping a continuous scalar representation of expression value to an expression embedding. Following the standard Scanpy [113] pipelines, the normalization process normalizes total sum of expression values of each cell to 1e4 and then applies the log transformation (scanpy.pp.log1p) to the data. After normalization, the expression matrix with n cells and m genes can be represented as a positive matrix $X^{n \times m} \in \mathbb{R}^{+}$. To convert the expression values to vector embeddings that could be effectively used by trans-former-based model, we adapt the expression value binning [117, 102] to map scalar representation of expression values to a series of relative expression embeddings.

Specifically, we assign k tokens to represent the expression values. For cell i, we first retrieve the largest expression value $L_i = max(X_{i,:})$, and then assign k uniformly distributed continuous intervals $[a_b, a_{b+1}],\ b \in [0, k-1]$ between $(0, L_i)$, where $a_0 = 0$ and $a_k = L$. For a given expression value of gene j in cell i, we denote $B_{i,j} = b,\ if\ X_{i,j} \in [a_b, a_{b+1}]$. Unless specified elsewhere, in most of our experiments we set number of bins k to be 50. This process would generate a new representation of data denoted by bin IDs $B^{n \times m} \in \{1, 2, ..., k\}$. During the modeling, similar as gene tokens, we maintain a set of differentiable expression tokens $E_g^{k \times d} = [e_1^d, e_2^d, ..., e_k^d]$ as a part of parameter of the model, and retrieve certain values through a dictionary lookup fashion implemented with Pytorch embedding module during training and inference.

4.2.2 Model architecture and pretraining

For the pretraining of the foundation model, we adapt a masked expression prediction pretext, where at each iteration we randomly mask a certain ratio of non-zero expression values for each cell, and use the rest of observations to predict the masked values.

Our ExpressionTransformer consists of two parts: an encoder block with N layers of traditional transformer encoders, and an expression decoder designed for masked expression prediction pretext. For the encoder block, we adapt the original transformer encoder modules [109]. The transformer encoder has a multi-head self-attention mechanism to learn increasingly com-plexed representation of input tokens. The inputs are query matrix Q, key matrix K, and a value matrix V. Each attention head computes the attention of its inputs,

$$Attention_i(Q, K, V) = softmax\left(\frac{QW_i^Q(KW_i^K)T}{\sqrt{d_k}}\right)VW_i^V \tag{4.2}$$

where W_i^Q, W_i^K, W_i^V are the learnable parameters of head i. For transformer encoders with self-attentions, the Q, K, and V are the same input matrix copied three times. The multi-head attention concatenate outputs from each single attention head and produce the output for each layer.

$$\begin{aligned} &MultiHead(Q, K, V) = Concat(\\ &\qquad Attention_1(Q, K, V), Attention_2(Q, K, V), ...Attention_h(Q, K, V)) \end{aligned} \tag{4.3}$$

We use ϕ_j to denote a set of p genes with non-zero expressions in cell j. The input to the encoder block is the gene tokens $T_{\phi_j}^{p \times d}$ of non-zero genes. The input is then push through the encoder block and produce a contextualized representations of gene set ϕ_j which we denote as $T'^{\,p \times d}_{\phi_j}$. Within the encoder block, each layer's output has the same shape as its input, and then will be used as the input for next layer. We set number of encoder layers N=12 and number of attention heads h=4 during most of our experiments. After we encode the inputs and get the representation of genes ${T'}_{\phi_j}^{p\times d}$ from the last layer of the encoder block, we use a masking process to mask expression levels of certain genes and train the model to use the rest of observations to predict the masked values. The masking process masks a portion of the genes from ϕ_j. Here, we denote the set of q masked genes as ψ_j, and their contextualized embeddings as ${T'}_{\psi_j}^{q\times d}$. During the pretraining, for each iteration, we randomly select a set of genes ψ_j from ϕ_j for cell j. The contextualized embedding of the observed genes is then ${T'}_{\phi_j-\psi_j}^{(p-q)\times d}$.

In order to facilitate the model to learn useful representations of gene tokens, we propose an expression transformer decoder layer: we set Q as embeddings of the masked genes ${T'}_{\psi_j}^{q\times d}$; K as the observed gene token embeddings from the encoder outputs ${T'}_{\phi_j-\psi_j}^{(p-q)\times d}$; and V as the expression values of the observed genes $E_{\phi_j-\psi_j}^{(p-q)\times d}$. The expression decoder first computes the cross attention between the masked genes and unmasked genes tokens, and then applies the cross-attention map and a learnable parameter W^V to the expression value V. This process will encourage the model to learn useful interactions between masked and unmasked genes based on their contextualized embedding and then apply multi-head attentions and transformations to the expression values based on these interactions to predict the masked outputs.

To improve pretraining efficiency, we focus on genes with non-zero expression values for each cell. This allows us to reduce the sequence length of each cell to around 1000 throughout the experiments, since most of the cells do not have observed expressed genes over 1000. In implementation, we pad the input sequences of all cells to the same length of 1100, while we use logic mask to set the attention score of padded tokens to -inf to zero out the attentions on these padded tokens.

The output from the expression decoder $P_{\psi_j}^{q\times d}$ will then be sent to a multi-layer perceptron (MLP) with softplus activation for final prediction. We also concatenate the largest expres-

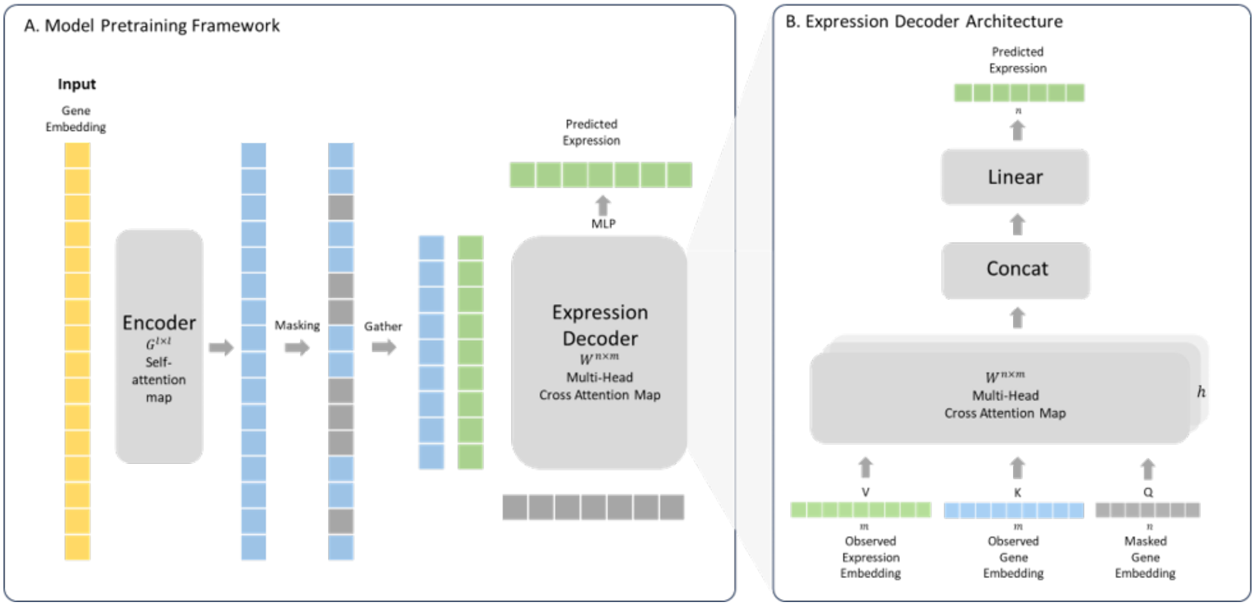


Figure 4.1: Overview of pretraining framework and Expression Transformer model design. (A) Model pretraining framework. The model is pretrained on large-scale single cell datasets with masked expression value prediction pretext. During the pretraining, for each iteration, the expression values of a subset of randomly selected genes are masked, and the model is trained to infer the masked values based on the rest of observed gene expressions. (B) Architecture of the Expression Decoder. Cross attention maps are computed based on the observed gene embeddings (K) and masked gene embeddings (Q), and then act on the observed expressions (V) to eventually generated the predictions for expression values of the masked genes.

sion value L_j of the given cell in order to provide the MLP enough information to predict the exact expression values.

$$\hat{X}_{j,\psi_i} = SoftPlus(MLP(concatenate(P_{\psi_j}, L_j))) \tag{4.4}$$

We trained the model with mean square error (MSE) loss function using the Adam optimizer with learning rate 1e-4.

$$l_j = \sum_{v\in\psi_i} \left(X_{i,v} - \hat{X}_{i,v}\right)^2 \tag{4.5}$$

4.2.3 Evaluation of gene embeddings

To demonstrate that the model learns gene embeddings that contain meaningful information, we tested these gene embeddings on three downstream tasks: predicting gene loss of function (LoF) score, predicting protein interactions, and predicting gene dose-sensitivity. For each task, we retrieve the gene embedding tokens T_g after pretraining and, only fine-tuned them with simple models using Scikit-learn library. We benchmarked our method with existing methods that produces gene embeddings, including Gene2vec [31] and Geneformer [103]. Specifically, the Gene2vec embeddings are provided at their GitHub site (https://github.com/jingcheng-du/Gene2vec) and we retrieve the Geneformer gene tokens by loading their pretrained model provided at (https://huggingface.co/ctheodoris/Geneformer) and extracted the weights from the embedding layer. To test the effectiveness of the expression decoder, we benchmarked our model with a version replacing the final decoder with a BERT-style encoder (denoted as "w/o Decoder").

For predicting protein interactions, we collected the paired interaction data from the STRING database [100, 110]. It contains experimental evidence of interactions between pairs of proteins. We labeled protein pairs with experimental interaction evidence over 200 as positive samples, and those with low-er than 200 as negative samples. We retrieved the embedding for each method as described pre-viously and match the genes with corresponding proteins according to their Ensembl gene and protein IDs. We concatenated the embeddings for each pair of the samples as inputs, and trained a Gradient Boosting Classification Tree with default settings for all the methods. The dataset is randomly divided into training and testing, and the results are reported based on the test set. We reported the test accuracy and area under the curve (AUC) of the receiver operating characteristic (ROC) curve for each method.

For predicting the gene loss of function z-score, we collected the data from [52] and match the gene ID with the genes in our pretraining. We retrieve the embeddings in the same way as described above, and fit a liner regression with the provided data. We visualize the predictions with scatter plots with ground truth versus predictions. In addition, we also report the Pearson correlation between predictions and ground truth.

For predicting the gene dose sensitivity, we followed the procedures in [103]. We collected the data from [78] and retrieved the list of MRDS and MRDIS genes. Here, we reference the AUC score of the Geneformer from the original paper [103]. For the rest of the methods, we used the same procedure to get the embeddings for each gene. After we align each gene's embedding, we treat this as a binary classification task, by using each gene's embedding as input and predict whether the gene is from MRDS or MRDIS list. We fitted trained a Gradient Boosting Classification Tree with default settings and reported test AUC and accuracy on the leftout test set (20%).

4.2.4 Evaluation of Imputation

To evaluate if the model could accurately predict the missing values from scRNA-seq data, we collected the several popular scRNA-seq datasets that used for benchmarks [63, 108, 48, 64]. For datasets with count data, we use the same normalization and preprocessing as described above, by first normalizing the total counts of each cell to 1e4 with l1-normalization and then applying the log transformation to the data points. Following similar procedures [48], genes with non-zero expression values are randomly dropped out and used for evaluation. For other methods, we followed the recommended workflow to generate imputation data, and for Expression Transformer, similar as the workflow during pretraining, we treated dropouts as masked genes and use the prediction from the decoder as the imputed values. The evaluation metrics used here are the mean absolute error (MAE) and the Pearson correlations between imputed and ground truth values.

4.3 Results

4.3.1 Overview of the pretraining process

In order to fully utilize the large-scale datasets and information measured within each single cell, we developed a transformer-based model (Figure 4.1) and pretrained it on a masked value prediction pretext. The Expression Transformer model contains an embedding module, an encoder block, and an expression decoder which is designed to perform the masked prediction pretext and facilitate the model to learn useful representation of genes. The embedding module maintains embeddings of all the genes selected for modelling (Methods) as learnable parameters. It returns the corresponding embedding of a given gene when that gene is being used, and updates the embedding vector when loss is backpropagated. The encoder block consists of standard transformer encoder layers [109], learning specific context of each single cell. Since scRNA-seq data contains lots of observed zeros of which most are considered as random dropouts, we only input genes with non-zero expression values during the pretraining since these measurements are considered to be more reliable. The en-coder block then takes embeddings of genes with non-zero expressions as input. For each forward pass, these embeddings are pushed through the transformer encoder layers to generate specific cotextualized embeddings of each cell given the set of observed genes. These contextualized embed-dings are then subjected to random masking, and then sent to the Expression Decoder for masked prediction. The masked expression value prediction is a standard pretext for self-supervised learn-ing. It only requires gene expression values and no other labels, and could learn generalizable features genes by training on broad datasets. For each training iteration, input genes are randomly subset as observed or masked. Expression values of observed genes are provided through expression embeddings (Methods) to the model, while the expression values of masked genes are substituted with a mask token. During the pretraining, the Expression Decoder is trained to learn relationships between genes, then utilize this information to predict masked values based on the ob-served expression values. To facilitate this, we design the Expression Decoder with a cross-attention mechanism learning correlations between observed and masked genes. Then the cross-attention maps act on the expression embeddings of observed genes through multi-head attention, and then produce the final prediction on the masked genes (Methods).

After pretraining, we retrieve the embedding module as semantic representations of genes. We demonstrate the utility of these embeddings by finetuning it for several different downstream tasks with simple prediction models that could run on devices without GPU supports. We benchmarked the performance of over embeddings with Geneformer [103] and Gene2Vec [31]. Also, to test if the Expression Transformer could make predictions on gene expressions generalizable to new datasets, we tested our model for imputing missing expression values and benchmarked it with other popular scRNA-seq imputation methods.

4.3.2 Inferring missing expression values with pretrained Expression Transformer model

High sparsity has been a longstanding issue for analyzing single cell sequencing data. Due to the limited read depth and coverage for each single cell, the measurements are filled with abundance of zeros, of which a significant amount of them are caused by expressed reads not captured by sequencing process. These noises hinder the downstream analysis of the data, and often requires an imputation process to recover some expression values. Many computational methods have been developed to tackle this problem. Most of the methods utilize similarity between cells to impute missing values. Here our model takes a different approach by pretraining on inferring relationships between genes and then make predictions based on the available observations. To test if our approach is valid, we used the 10x PBMC [48] dataset and Human Cell Landscape [41] (HCL) dataset for benchmarking. Each dataset

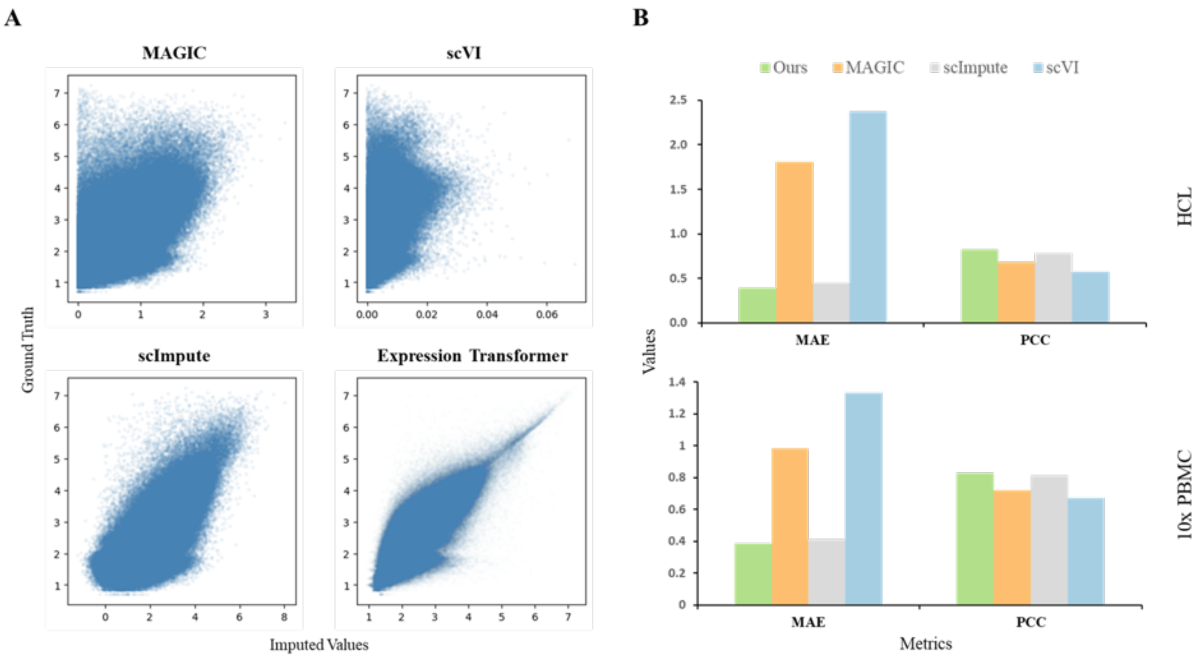


Figure 4.2: Results on imputing scRNA-seq data. (A) Scatter plots of imputed predictions of different methods against the ground truth values on HCL dataset. (B) Quantitative metrics of Mean Absolute Error (MAE) and Pearson Correlation Coefficient (PCC) calculated based on imputed predictions and ground truth from 10x PBMC and HCL datasets.

is first subject to artificial dropouts, and the dropped non-zero expression values are used as the ground truth for evaluation. We followed the same data preprocessing (Methods) pipeline as the pretraining, and feed the data to each method as instructed in their original tutorials. We benchmarked our method with MAGIC [108], scImpute [63], and scVI [66], and the results are shown in Figure 4.2. Figure 4.2A shows the scatter plot of imputed values plotted against the ground truth values on the HCL dataset. Imputations generated by Expression Transformer algins better with the ground truth values. This is in line with the quantitative scores shown in Figure 4.2B, where Expression Transformer has lower Mean Absolute Error (MAE) and higher Pearson Correlation Coefficient (PCC) between ground truth and imputed value. This demonstrates that foundation models pretrained on broad datasets could make meaningful predictions on missing expression values. The improved performance showed the benefits of large scale pretraining and transfer learning.

4.3.3 Predicting gene and protein interactions with pretrained semantic representation of genes

Providing functional annotations for each gene is an important yet challenging tasks. Most of the existing Gene Ontology (GO) based approaches try to categorize genes by knowledge of biological processes, pathways, and etc. However, these annotations are not entirely objective, given that it is impossible to provide all aspects of experimental evidence for each gene; and often given a specific task, the relationship between different genes could be varying. Here in the following tasks, we demonstrate that through large-scale pretraining, we provide meaningful annotations of genes as quantitative embedding vectors with a complete data-driven manner. To demonstrate that the embeddings capture meaningful features of genes, we use the pretrained embeddings as input and finetune a very simple model to transfer it to other prediction tasks.

In the first task, we used paired gene embeddings generated from different methods as input, to finetune a Gradient Boosting Tree to classify whether the gene pairs or its encoded proteins have interactions. The gene-gene interaction dataset is collected from [31], where gene pairs that share GO annotations with experimental evidence are marked as positive samples, and gene pairs that do not share any GO terms as negative samples. The protein interaction dataset is collected from STRING database [100, 110], and we labeled protein pairs with experimental interaction evidence over 200 as positive samples, and those with lower than 200 as negative samples. The downstream tasks here are considered as binary classification, with embeddings of gene pairs used as input and the classifier will predict whether these two genes have interactions. We finetuned separate models for embeddings from different method, and for different tasks as well (for predicting gene-gene interactions and protein interactions respectively). We benchmarked embeddings of Expression Transformer with embeddings generated by Gene2vec [31] and Geneformer [103] (Methods). Gene2vec trained a Word2vec [73] based model on many bulk RNA-seq co-expression datasets; Geneformer is a BERT-

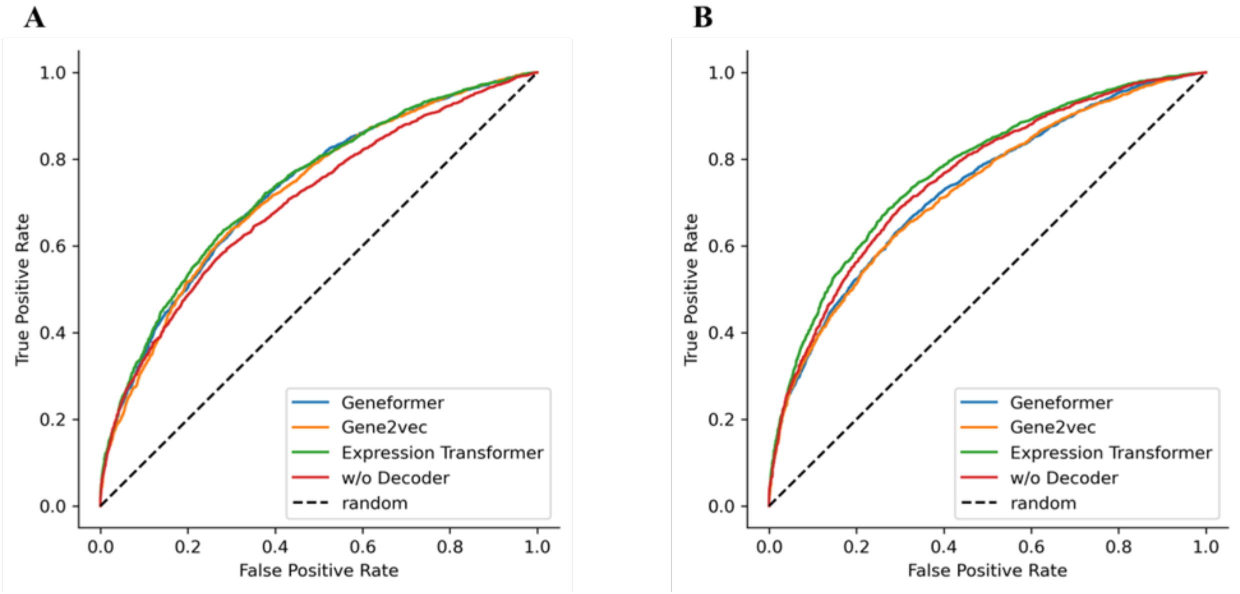


Figure 4.3: Results on predicting gene and protein interactions. (A) ROC curve of classifiers trained on different embeddings to predict gene-gene interactions. (B) ROC curve of classifiers trained on different embeddings to predict protein interactions.

based LLM trained on broad scRNA-seq datasets with gene IDs ranked by expression values. In the meantime, in order to test the effectiveness of the Expression Decoder design, we also bench-marked with an Expression Transformer model with only transformer encoders, which we denote as "w/o Decoder" (Methods). In the "w/o Decoder" model, we replace the decoder module to a standard transformer encoder and add the expression values to the observed genes. Figure 4.3 shows the test set ROC curve of classifiers trained on different embeddings. We can see embeddings from Expression Transformer have the largest area under the curve (AUC). And if we compare it with the AUC of the model without Expression Decoder, we can see that the Expression Decoder helps to learn more meaningful embeddings of genes to better predict gene-gene and protein interactions

4.3.4 Predicting gene loss-of-function score

A gene loss of function (LoF) z-score is a statistical measure that is used to quantify the extent to which a genetic variant is predicted to disrupt the function of a gene. LoF score is associated with gene's tolerance against mutations, and is useful for identifying disease-related genes, understand-ing genetic variability, and studying gene functions. In this task, we used a simple linear regression to finetune the gene embeddings for predicting gene LoF score. The LoF score dataset is collected from [52]. Again, we benchmarked with Gene2vec, Geneformer, and a "w/o Decoder" model. Embeddings of each gene from different methods are used as input for the linear regression to regress the corresponding gene's LoF score. From Figure 4.4, we can see that predictions based on Expression Transformer's Embedding align better with the true LoF score compared with other methods. In the meantime, embeddings trained on single cell datasets performs better than Gene2vec which trained on bulk RNA-seq data. This might be due to that scRNA-seq datasets have greater coverage and better resolution of cell types and tissues. One thing to note is that even though variant information is not used at all during the pretraining, the embeddings benchmarked here are still able to make predictions on mutational constraints. This indicates that gene's tolerance of variants may also relate to its functional relationship with other genes.

4.3.5 Predicting gene dosage sensitivity

Gene dosage sensitivity refers to the intolerance of a gene to variations in its copy number. Copy Number Variants (CNVs) are sequence variations in the genome that result in deletion or duplication of segments of DNA that can affect the copy number of certain genes. For dosage-sensitive genes, CNVs could disrupt the normal function of the gene that would lead to developmental dis-orders and diseases. The dosage sensitivity of genes is related to the compensatory mechanism of the gene regulatory network. Predicting gene dosage sensitivity

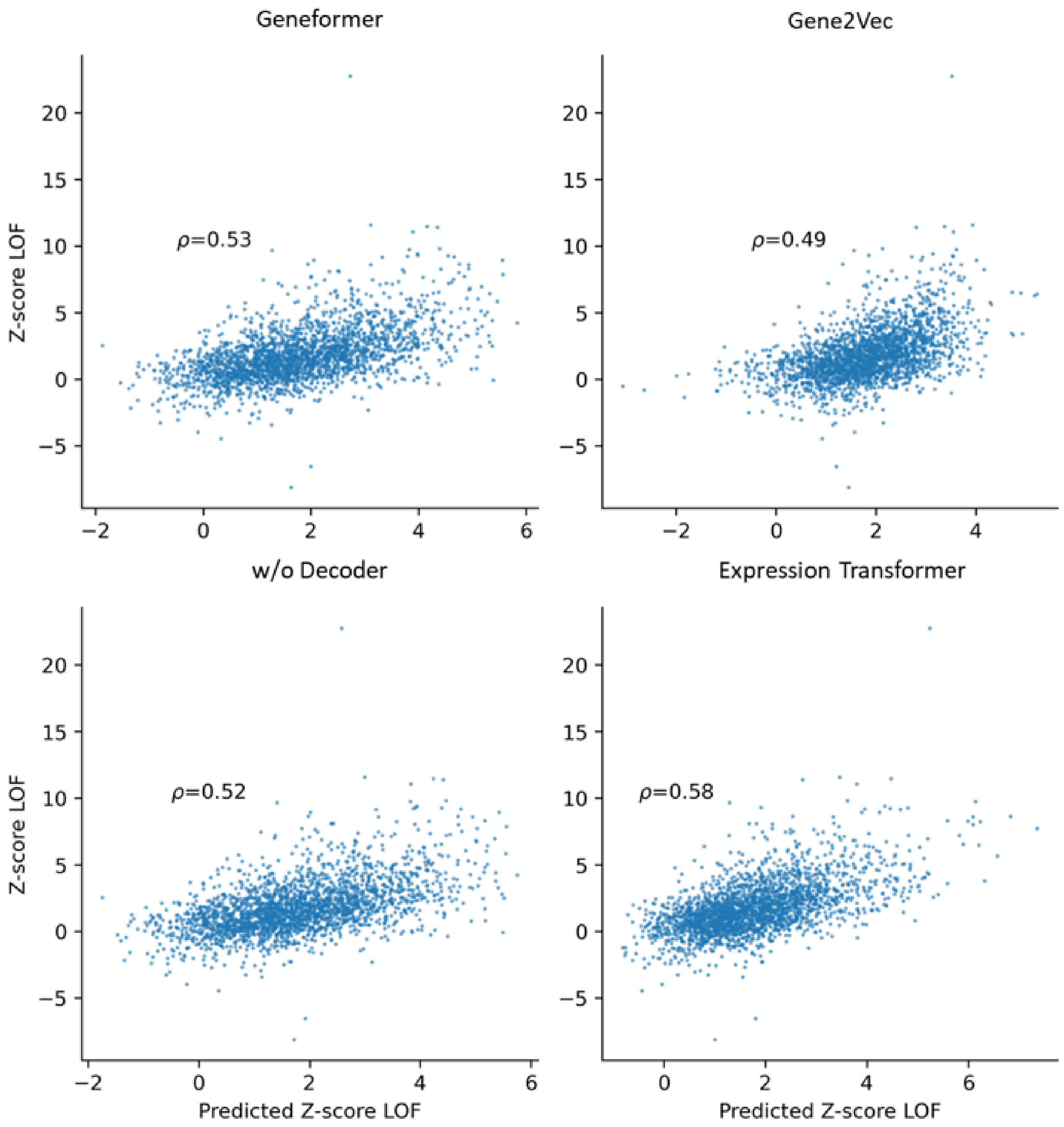


Figure 4.4: Results on predicting LoF score. The scatter plots show the predicted LoF score with embeddings from different methods plotted against the ground truth. ρ indicates the Pearson correlation between predictions and the ground truth.

has implications on therapeutic tar-gets of certain diseases. Here, we collected gene sets previously reported to be dosage-sensitive or not [103, 102], and consider this a binary classification task to predict whether a given gene is dose sensitive or not providing this gene's

Table 4.1: AUC score of gene dosage sensitivity prediction with embeddings from different methods.

Methods	Geneformer	Expression Transformer
AUC	0.91	0.92

embedding. Following the evaluation pipeline in Geneformer [103], we trained a gradient boosting tree based classifier on the gene list and report the AUC of ROC curve in Table 4.1. We can see Expression Transformer achieved comparable results as the Geneformer. This could also demonstrate the representation learned by Expression Transformer could be transferred to tasks related to network biology.

4.4 Discussions

In this study, we explored pretraining foundation model on broad scRNA-seq datasets with masked value prediction pretext. Instead of focusing on learning embeddings of cells and perform batch integration and imputation through cell-cell similarity, the Expression Transformer learns relationship between genes that generalizable across different datasets. In our experiments, the pretrained model is able to capture useful interactions that could be used to accurately impute missing expression values. In order to facilitate the model to learn meaningful representations of genes, we designed an Expression Decoder based on the cross-attention mechanism. The Expression Decoder first computes attention maps based on the contextualized gene embeddings between observed and masked genes, and then applied multi-head attention accordingly to predict the masked expression values. We retrieved the semantic embeddings after pretraining, and demonstrated the utility of the embeddings by finetuning them to other downstream tasks with relatively simple model. The downstream tasks include predicting gene-gene and protein interactions, predicting gene loss of function (LoF) scores and predicting gene dosage sensitivity. Embeddings retrieved from Expression Transformer showed better results in these downstream tasks, which demonstrates that the pretraining generates biologically meaningful semantic representations of genes; and the embedings from Expression Decoder generates better results compared with standard transformer encoder, which shows the benefits of our design. These findings highlight the potential for large foundation models to generate fundamental insights on biology, as well as assist analysis of small datasets through transfer learning. For future works, exploring different disease conditions or treatments as independent controllable factors could be an interesting direction, as it could provide more insights on how gene interactions are interrupted under different conditions and potentially offers candidate targets for diseases treatments. Also, the utility of single-cell contextualized embeddings is under-explored in our study. A contextualized embedding based on each single cell's measurement could be useful for many cell-based tasks, including cell annotations, perturbation studies, drug responsive study, etc.

Bibliography

[1] M. AlQuraishi and P. K. Sorger. Differentiable biology: using deep learning for biophysics-based and data-driven modeling of molecular mechanisms. *Nature methods*, 18(10):1169–1180, 2021.

[2] M. Amodio, D. Van Dijk, K. Srinivasan, W. S. Chen, H. Mohsen, K. R. Moon, A. Campbell, Y. Zhao, X. Wang, M. Venkataswamy, et al. Exploring single-cell data with deep multitasking neural networks. *Nature methods*, 16(11):1139–1145, 2019.

[3] R. Argelaguet, A. S. Cuomo, O. Stegle, and J. C. Marioni. Computational principles and challenges in single-cell data integration. *Nature biotechnology*, 39(10):1202–1215, 2021.

[4] Ž. Avsec, V. Agarwal, D. Visentin, J. R. Ledsam, A. Grabska-Barwinska, K. R. Taylor, Y. Assael, J. Jumper, P. Kohli, and D. R. Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18(10):1196–1203, 2021.

[5] R. Bhairavabhotla, Y. C. Kim, D. D. Glass, T. M. Escobar, M. C. Patel, R. Zahr, C. K. Nguyen, G. K. Kilaru, S. A. Muljo, and E. M. Shevach. Transcriptome profiling of human foxp3+ regulatory t cells. *Human immunology*, 77(2):201–213, 2016.

[6] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[7] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[8] A. P. Boyle, S. Davis, H. P. Shulha, P. Meltzer, E. H. Margulies, Z. Weng, T. S. Furey, and G. E. Crawford. High-resolution mapping and characterization of open chromatin across the genome. *Cell*, 132(2):311–322, 2008.

[9] C. Bravo González-Blas, L. Minnoye, D. Papasokrati, S. Aibar, G. Hulselmans, V. Christiaens, K. Davie, J. Wouters, and S. Aerts. cistopic: cis-regulatory topic modeling on single-cell atac-seq data. *Nature methods*, 16(5):397–400, 2019.

[10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[11] J. D. Buenrostro, B. Wu, H. Y. Chang, and W. J. Greenleaf. Atac-seq: a method for assaying chromatin accessibility genome-wide. *Current protocols in molecular biology*, 109(1):21–29, 2015.

[12] J. D. Buenrostro, B. Wu, U. M. Litzenburger, D. Ruff, M. L. Gonzales, M. P. Snyder, H. Y. Chang, and W. J. Greenleaf. Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature*, 523(7561):486–490, 2015.

[13] A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*, 36(5):411–420, 2018.

[14] J. Cao, D. R. O'Day, H. A. Pliner, P. D. Kingsley, M. Deng, R. M. Daza, M. A. Zager, K. A. Aldinger, R. Blecher-Gonen, F. Zhang, et al. A human cell atlas of fetal gene expression. *Science*, 370(6518):eaba7721, 2020.

[15] K. Cao, X. Bai, Y. Hong, and L. Wan. Unsupervised topological alignment for single-cell multi-omics integration. *Bioinformatics*, 36(Supplement_1):i48–i56, 2020.

[16] Y. Cao, L. Fu, J. Wu, Q. Peng, Q. Nie, J. Zhang, and X. Xie. Sailer: scalable and accurate invariant representation learning for single-cell atac-seq processing and integration. *Bioinformatics*, 37(Supplement_1):i317–i326, 2021.

[17] Y. Cao, L. Fu, J. Wu, Q. Peng, Q. Nie, J. Zhang, and X. Xie. Integrated analysis of multimodal single-cell data with structural similarity. *Nucleic acids research*, 50(21):e121–e121, 2022.

[18] C. Chen, Q. Meng, Y. Xia, C. Ding, L. Wang, R. Dai, L. Cheng, P. Gunaratne, R. A. Gibbs, S. Min, et al. The transcription factor pou3f2 regulates a gene coexpression network in brain tissue from patients with psychiatric disorders. *Science translational medicine*, 10(472):eaat8178, 2018.

[19] H. Chen, C. Lareau, T. Andreani, M. E. Vinyard, S. P. Garcia, K. Clement, M. A. Andrade-Navarro, J. D. Buenrostro, and L. Pinello. Assessment of computational methods for the analysis of single-cell atac-seq data. *Genome biology*, 20(1):1–25, 2019.

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

[21] X. Chen, R. Miragaia, K. Natarajan, and S. Teichmann. A rapid and robust method for single cell chromatin accessibility profiling. nat commun 9 (1): 5345, 2018.

[22] Z. Chen, J. Zhang, J. Liu, Z. Zhang, J. Zhu, D. Lee, M. Xu, and M. Gerstein. Scan-atac-sim: a scalable and efficient method for simulating single-cell atac-seq data from bulk-tissue experiments. *Bioinformatics*, 37(12):1756–1758, 2021.

[23] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[24] M. Collin and V. Bigley. Human dendritic cell subsets: an update. *Immunology*, 154(1):3–20, 2018.

[25] D. A. Cusanovich, R. Daza, A. Adey, H. A. Pliner, L. Christiansen, K. L. Gunderson, F. J. Steemers, C. Trapnell, and J. Shendure. Multiplex single-cell profiling of chromatin accessibility by combinatorial cellular indexing. *Science*, 348(6237):910–914, 2015.

[26] D. A. Cusanovich, A. J. Hill, D. Aghamirzaie, R. M. Daza, H. A. Pliner, J. B. Berletch, G. N. Filippova, X. Huang, L. Christiansen, W. S. DeWitt, et al. A single-cell atlas of in vivo mammalian chromatin accessibility. *Cell*, 174(5):1309–1324, 2018.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[28] D. v. Dijk, J. Nainys, R. Sharma, P. Kaithail, A. J. Carr, K. R. Moon, L. Mazutis, G. Wolf, S. Krishnaswamy, and D. Pe'er. Magic: A diffusion-based imputation method reveals gene-gene interactions in single-cell rna-sequencing data. *BioRxiv*, page 111591, 2017.

[29] J. Ding and A. Regev. Deep generative model embedding of single-cell rna-seq profiles on hyperspheres and hyperbolic spaces. *Nature communications*, 12(1):2554, 2021.

[30] S. Domcke, A. J. Hill, R. M. Daza, J. Cao, D. R. O'Day, H. A. Pliner, K. A. Aldinger, D. Pokholok, F. Zhang, J. H. Milbank, et al. A human cell atlas of fetal chromatin accessibility. *Science*, 370(6518):eaba7612, 2020.

[31] J. Du, P. Jia, Y. Dai, C. Tao, Z. Zhao, and D. Zhi. Gene2vec: distributed representation of genes based on co-expression. *BMC genomics*, 20:7–15, 2019.

[32] Z. Duren, X. Chen, M. Zamanighomi, W. Zeng, A. T. Satpathy, H. Y. Chang, Y. Wang, and W. H. Wong. Integrative analysis of single-cell genomics data by coupled nonnegative matrix factorizations. *Proceedings of the National Academy of Sciences*, 115(30):7723–7728, 2018.

[33] R. Fang, S. Preissl, Y. Li, X. Hou, J. Lucero, X. Wang, A. Motamedi, A. K. Shiau, X. Zhou, F. Xie, et al. Comprehensive analysis of single cell atac-seq data with snap-atac. *Nature communications*, 12(1):1337, 2021.

[34] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. *arXiv preprint arXiv:1903.10145*, 2019.

[35] L. Fu, L. Zhang, E. Dollinger, Q. Peng, Q. Nie, and X. Xie. Predicting transcription factor binding in single cells through deep learning. *Science advances*, 6(51):eaba9031, 2020.

[36] A. Gayoso, Z. Steier, R. Lopez, J. Regier, K. L. Nazor, A. Streets, and N. Yosef. Joint probabilistic modeling of single-cell multi-omic data with totalvi. *Nature methods*, 18(3):272–282, 2021.

[37] B. Gong, Y. Zhou, and E. Purdom. Cobolt: integrative analysis of multimodal single-cell sequencing data. *Genome biology*, 22(1):1–21, 2021.

[38] J. Granja, M. Corces, S. Pierce, S. Bagdatli, H. Choudhry, H. Chang, and W. Greenleaf. Archr: an integrative and scalable software package for single-cell chromatin accessibility analysis. biorxiv: 2020.2004. 2028.066498, 2020.

[39] L. Haghverdi, A. T. Lun, M. D. Morgan, and J. C. Marioni. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology*, 36(5):421–427, 2018.

[40] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[41] X. Han, Z. Zhou, L. Fei, H. Sun, R. Wang, Y. Chen, H. Chen, J. Wang, H. Tang, W. Ge, et al. Construction of a human cell landscape at single-cell level. *Nature*, 581(7808):303–309, 2020.

[42] Y. Hao, S. Hao, E. Andersen-Nissen, W. M. Mauck, S. Zheng, A. Butler, M. J. Lee, A. J. Wilk, C. Darby, M. Zager, et al. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587, 2021.

[43] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[44] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[45] B. Hie, B. Bryson, and B. Berger. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Nature biotechnology*, 37(6):685–691, 2019.

[46] W. Hou, Z. Ji, H. Ji, and S. C. Hicks. A systematic evaluation of single-cell rna-sequencing imputation methods. *Genome biology*, 21:1–30, 2020.

[47] M. Huang, J. Wang, E. Torre, H. Dueck, S. Shaffer, R. Bonasio, J. I. Murray, A. Raj, M. Li, and N. R. Zhang. Saver: gene expression recovery for single-cell rna sequencing. *Nature methods*, 15(7):539–542, 2018.

[48] Z. Huang, J. Wang, X. Lu, A. Mohd Zain, and G. Yu. scggan: single-cell rna-seq imputation by graph-based generative adversarial network. *Briefings in bioinformatics*, 24(2):bbad040, 2023.

[49] Y. Ji, Z. Zhou, H. Liu, and R. V. Davuluri. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.

[50] S. Jin, L. Zhang, and Q. Nie. scai: an unsupervised approach for the integrative analysis of parallel single-cell transcriptomic and epigenomic profiles. *Genome biology*, 21:1–19, 2020.

[51] K. Kanemaru, J. Cranley, D. Muraro, A. M. Miranda, S. Y. Ho, A. Wilbrey-Clark, J. Patrick Pett, K. Polanski, L. Richardson, M. Litvinukova, et al. Spatially resolved multiomics of human cardiac niches. *Nature*, pages 1–10, 2023.

[52] K. J. Karczewski, L. C. Francioli, G. Tiao, B. B. Cummings, J. Alföldi, Q. Wang, R. L. Collins, K. M. Laricchia, A. Ganna, D. P. Birnbaum, et al. The mutational constraint spectrum quantified from variation in 141,456 humans. *Nature*, 581(7809):434–443, 2020.

[53] I. D. Karemaker and M. Vermeulen. Single-cell dna methylation profiling: technologies and biological applications. *Trends in biotechnology*, 36(9):952–965, 2018.

[54] D. Kingma and J. Ba. Adam: A method for stochastic optimization in: Proceedings of the 3rd international conference for learning representations (iclr'15). *San Diego*, 500, 2015.

[55] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[56] S. L. Klemm, Z. Shipony, and W. J. Greenleaf. Chromatin accessibility and the regulatory epigenome. *Nature Reviews Genetics*, 20(4):207–220, 2019.

[57] I. Korsunsky, N. Millard, J. Fan, K. Slowikowski, F. Zhang, K. Wei, Y. Baglaenko, M. Brenner, P.-r. Loh, and S. Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature methods*, 16(12):1289–1296, 2019.

[58] A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[59] A. R. Kriebel and J. D. Welch. Uinmf performs mosaic integration of single-cell multi-omic datasets using nonnegative matrix factorization. *Nature communications*, 13(1):780, 2022.

[60] D. Lähnemann, J. Köster, E. Szczurek, D. J. McCarthy, S. C. Hicks, M. D. Robinson, C. A. Vallejos, K. R. Campbell, N. Beerenwinkel, A. Mahfouz, et al. Eleven grand challenges in single-cell data science. *Genome biology*, 21(1):1–35, 2020.

[61] M. Lawrence, W. Huber, H. Pagès, P. Aboyoun, M. Carlson, R. Gentleman, M. T. Morgan, and V. J. Carey. Software for computing and annotating genomic ranges. *PLoS computational biology*, 9(8):e1003118, 2013.

[62] C.-U. T. C. L. lcai@ caltech. edu 21 b Shendure Jay 9 Trapnell Cole 9 Lin Shin shinlin@ uw. edu 2 e Jackson Dana 9, U. T. Z. K. kzhang@ bioeng. ucsd. edu 15 b Sun Xin 15 Jain Sanjay 24 Hagood James 25 Pryhuber Gloria 26 Kharchenko Peter 8, C. I. of Technology TTD Cai Long lcai@ caltech. edu 21 b Yuan Guo-Cheng 35 Zhu Qian 35 Dries Ruben 35, H. T. Y. P. peng_yin@ hms. harvard. edu 36 37 b Saka Sinem K. 36 37 Kishi Jocelyn Y. 36 37 Wang Yu 36 37 Goldaracena Isabel 36 37, P. T. L. J. jlaskin@ purdue. edu 10 b Ye DongHye 10 38 Burnum-Johnson Kristin E. 39 Piehowski Paul D. 39 Ansong Charles 39 Zhu Ying 39, S. T. H. P. harbury@ stanford. edu 11 b Desai Tushar 40 Mulye Jay 11 Chou Peter 11 Nagendran Monica 40, et al. The human body at cellular resolution: the nih human biomolecular atlas program. *Nature*, 574(7777):187–192, 2019.

[63] W. V. Li and J. J. Li. An accurate and robust imputation method scimpute for single-cell rna-seq data. *Nature communications*, 9(1):997, 2018.

[64] G. C. Linderman, J. Zhao, M. Roulis, P. Bielecki, R. A. Flavell, B. Nadler, and Y. Kluger. Zero-preserving imputation of single-cell rna-seq data. *Nature communications*, 13(1):192, 2022.

[65] J. Liu, Y. Huang, R. Singh, J.-P. Vert, and W. S. Noble. Jointly embedding multiple single-cell omics measurements. In *Algorithms in bioinformatics:... International Workshop, WABI..., proceedings. WABI (Workshop)*, volume 143. NIH Public Access, 2019.

[66] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

[67] M. D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, M. Interlandi, M. F. Müller, D. C. Strobl, L. Zappia, M. Dugas, M. Colomé-Tatché, et al. Benchmarking atlas-level data integration in single-cell genomics. *Nature methods*, 19(1):41–50, 2022.

[68] M. D. Luecken and F. J. Theis. Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular systems biology*, 15(6):e8746, 2019.

[69] S. Ma, B. Zhang, L. M. LaFave, A. S. Earl, Z. Chiang, Y. Hu, J. Ding, A. Brack, V. K. Kartha, T. Tay, et al. Chromatin potential identified by shared single-cell profiling of rna and chromatin. *Cell*, 183(4):1103–1116, 2020.

[70] F. Mair and M. Prlic. Omip-044: 28-color immunophenotyping of the human dendritic cell compartment. *Cytometry Part A*, 93(4):402–405, 2018.

[71] C. Mayer, C. Hafemeister, R. C. Bandler, R. Machold, R. Batista Brito, X. Jaglin, K. Allaway, A. Butler, G. Fishell, and R. Satija. Developmental diversification of cortical inhibitory interneurons. *Nature*, 555(7697):457–462, 2018.

[72] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[73] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[74] K. Minoura, K. Abe, H. Nam, H. Nishikawa, and T. Shimamura. scmm: Mixture-of-experts multimodal deep generative model for single-cell multiomics data analysis. *bioRxiv*, pages 2021–02, 2021.

[75] K. R. Moon, J. S. Stanley III, D. Burkhardt, D. van Dijk, G. Wolf, and S. Krishnaswamy. Manifold learning-based methods for analyzing single-cell rna-sequencing data. *Current Opinion in Systems Biology*, 7:36–46, 2018.

[76] D. Moyer, S. Gao, R. Brekelmans, A. Galstyan, and G. Ver Steeg. Invariant representations without adversarial training. *Advances in Neural Information Processing Systems*, 31, 2018.

[77] T. Nagano, Y. Lubling, T. J. Stevens, S. Schoenfelder, E. Yaffe, W. Dean, E. D. Laue, A. Tanay, and P. Fraser. Single-cell hi-c reveals cell-to-cell variability in chromosome structure. *Nature*, 502(7469):59–64, 2013.

[78] Z. Ni, X.-Y. Zhou, S. Aslam, and D.-K. Niu. Characterization of human dosage-sensitive transcription factor genes. *Frontiers in genetics*, 10:1208, 2019.

[79] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Advances in neural information processing systems 32. *Curran Associates, Inc*, pages 8024–8035, 2019.

[80] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[81] H. A. Pliner, J. S. Packer, J. L. McFaline-Figueroa, D. A. Cusanovich, R. M. Daza, D. Aghamirzaie, S. Srivatsan, X. Qiu, D. Jackson, A. Minkina, et al. Cicero predicts cis-regulatory dna interactions from single-cell chromatin accessibility data. *Molecular cell*, 71(5):858–871, 2018.

[82] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[83] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[84] A. Regev, S. A. Teichmann, E. S. Lander, I. Amit, C. Benoist, E. Birney, B. Bodenmiller, P. Campbell, P. Carninci, M. Clatworthy, et al. The human cell atlas. *elife*, 6:e27041, 2017.

[85] J. W. Rhodes, O. Tong, A. N. Harman, and S. G. Turville. Human dendritic cell subsets, ontogeny, and impact on hiv infection. *Frontiers in immunology*, 10:1088, 2019.

[86] D. Risso, F. Perraudeau, S. Gribkova, S. Dudoit, and J.-P. Vert. A general and flexible method for signal extraction from single-cell rna-seq data. *Nature communications*, 9(1):284, 2018.

[87] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[88] D. Rogers, A. Sood, H. Wang, J. J. van Beek, T. J. Rademaker, P. Artusa, C. Schneider, C. Shen, D. C. Wong, A. Bhagrath, et al. Pre-existing chromatin accessibility and gene expression differences among naive cd4+ t cells influence effector potential. *Cell Reports*, 37(9), 2021.

[89] O. Rozenblatt-Rosen, M. J. Stubbington, A. Regev, and S. A. Teichmann. The human cell atlas: from vision to reality. *Nature*, 550(7677):451–453, 2017.

[90] S. Sakaguchi, M. Miyara, C. M. Costantino, and D. A. Hafler. Foxp3+ regulatory t cells in the human immune system. *Nature Reviews Immunology*, 10(7):490–500, 2010.

[91] A. T. Satpathy, J. M. Granja, K. E. Yost, Y. Qi, F. Meschi, G. P. McDermott, B. N. Olsen, M. R. Mumbach, S. E. Pierce, M. R. 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.

[92] A. N. Schep, B. Wu, J. D. Buenrostro, and W. J. Greenleaf. chromvar: inferring transcription-factor-associated accessibility from single-cell epigenomic data. *Nature methods*, 14(10):975–978, 2017.

[93] A. Schlitzer, W. Zhang, M. Song, and X. Ma. Recent advances in understanding dendritic cell development, classification, and phenotype. *F1000Research*, 7, 2018.

[94] R. Singh, B. L. Hie, A. Narayan, and B. Berger. Schema: metric learning enables interpretable synthesis of heterogeneous single-cell modalities. *Genome biology*, 22(1):1–24, 2021.

[95] S. G. Stark, J. Ficek, F. Locatello, X. Bonilla, S. Chevrier, F. Singer, G. Rätsch, and K.-V. Lehmann. Scim: universal single-cell matching with unpaired feature sets. *Bioinformatics*, 36(Supplement_2):i919–i927, 2020.

[96] M. Stoeckius, C. Hafemeister, W. Stephenson, B. Houck-Loomis, P. K. Chattopadhyay, H. Swerdlow, R. Satija, and P. Smibert. Simultaneous epitope and transcriptome measurement in single cells. *Nature methods*, 14(9):865–868, 2017.

[97] T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, W. M. Mauck, Y. Hao, M. Stoeckius, P. Smibert, and R. Satija. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902, 2019.

[98] T. Stuart, A. Srivastava, S. Madad, C. A. Lareau, and R. Satija. Single-cell chromatin state analysis with signac. *Nature methods*, 18(11):1333–1341, 2021.

[99] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[100] D. Szklarczyk, A. L. Gable, K. C. Nastou, D. Lyon, R. Kirsch, S. Pyysalo, N. T. Doncheva, M. Legeay, T. Fang, P. Bork, et al. The string database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic acids research*, 49(D1):D605–D612, 2021.

[101] F. Tang, C. Barbacioru, Y. Wang, E. Nordman, C. Lee, N. Xu, X. Wang, J. Bodeau, B. B. Tuch, A. Siddiqui, et al. mrna-seq whole-transcriptome analysis of a single cell. *Nature methods*, 6(5):377–382, 2009.

[102] A. Thawani, J. Pujara, and F. Ilievski. Numeracy enhances the literacy of language models. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 6960–6967, 2021.

[103] C. V. Theodoris, L. Xiao, A. Chopra, M. D. Chaffin, Z. R. Al Sayed, M. C. Hill, H. Mantineo, E. M. Brydon, Z. Zeng, X. S. Liu, et al. Transfer learning enables predictions in network biology. *Nature*, pages 1–9, 2023.

[104] V. A. Traag, L. Waltman, and N. J. Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019.

[105] H. T. N. Tran, K. S. Ang, M. Chevrier, X. Zhang, N. Y. S. Lee, M. Goh, and J. Chen. A benchmark of batch-effect correction methods for single-cell rna sequencing data. *Genome biology*, 21:1–32, 2020.

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

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

[108] D. Van Dijk, R. Sharma, J. Nainys, K. Yim, P. Kathail, A. J. Carr, C. Burdziak, K. R. Moon, C. L. Chaffer, D. Pattabiraman, et al. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729, 2018.

[109] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[110] C. Von Mering, L. J. Jensen, B. Snel, S. D. Hooper, M. Krupp, M. Foglierini, N. Jouffre, M. A. Huynen, and P. Bork. String: known and predicted protein–protein associations, integrated and transferred across organisms. *Nucleic acids research*, 33(suppl_1):D433–D437, 2005.

[111] J. D. Welch, A. J. Hartemink, and J. F. Prins. Matcher: manifold alignment reveals correspondence between single cell transcriptome and epigenome dynamics. *Genome biology*, 18(1):1–19, 2017.

[112] J. D. Welch, V. Kozareva, A. Ferreira, C. Vanderburg, C. Martin, and E. Z. Macosko. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*, 177(7):1873–1887, 2019.

[113] F. A. Wolf, P. Angerer, and F. J. Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19:1–5, 2018.

[114] K. E. Wu, K. E. Yost, H. Y. Chang, and J. Zou. Babel enables cross-modality translation between multiomic profiles at single-cell resolution. *Proceedings of the National Academy of Sciences*, 118(15):e2023070118, 2021.

[115] L. Xiong, K. Xu, K. Tian, Y. Shao, L. Tang, G. Gao, M. Zhang, T. Jiang, and Q. C. Zhang. Scale method for single-cell atac-seq analysis via latent feature extraction. *Nature communications*, 10(1):4576, 2019.

[116] C. Xu, R. Lopez, E. Mehlman, J. Regier, M. I. Jordan, and N. Yosef. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Molecular systems biology*, 17(1):e9620, 2021.

[117] F. Yang, W. Wang, F. Wang, Y. Fang, D. Tang, J. Huang, H. Lu, and J. Yao. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nature Machine Intelligence*, 4(10):852–866, 2022.

[118] L. Yu, Y. Cao, J. Y. Yang, and P. Yang. Benchmarking clustering algorithms on estimating the number of cell types from single-cell rna-sequencing data. *Genome biology*, 23(1):49, 2022.

[119] Y. Zhang, T. Liu, C. A. Meyer, J. Eeckhoute, D. S. Johnson, B. E. Bernstein, C. Nusbaum, R. M. Myers, M. Brown, W. Li, et al. Model-based analysis of chip-seq (macs). *Genome biology*, 9(9):1–9, 2008.

[120] C. Zuo and L. Chen. Deep-joint-learning analysis model of single cell transcriptome and open chromatin accessibility data. *Briefings in Bioinformatics*, 22(4):bbaa287, 2021.