University of Nevada, Reno

Machine Learning Techniques for Cancer Subtype Discovery and Single-cell RNA Sequencing Data Analysis

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in Computer Science and Engineering

by

Nho Trung Duc Tran

Dr. Tin Nguyen, Advisor

August, 2023

THE GRADUATE SCHOOL

We recommend that the dissertation
prepared under our supervision by

Nho Trung Duc Tran

entitled

Machine Learning Techniques for Cancer Subtype Discovery and Single-cell RNA Sequencing Data Analysis

be accepted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

Dr. Tin Nguyen
Advisor

Dr. Frederick C. Harris, Jr.
Committee Member

Dr. Lei Yang
Committee Member

Dr. Hung (Jim) La
Committee Member

Dr. David Cantu
Graduate School Representative

Markus Kemmelmeier, Ph.D., Dean
Graduate School

August, 2023

Abstract

Cancer is an umbrella term that includes a range of disorders, from those that are aggressive and life-threatening to indolent lesions with low or delayed potential for progression to death. After 20 years of cancer screening, the chance of a person being diagnosed with prostate or breast cancer has nearly doubled. However, this has only marginally reduced the number of patients with advanced disease, suggesting that screening has resulted in the substantial harm of excess detection and over-diagnosis. At the same time, 30 to 50% of patients with non-small cell lung cancer (NSCLC) develop recurrence and die after curative resection, suggesting that a subset of patients would have benefited from more aggressive treatments at early stages. Although not routinely recommended as the initial course of treatment, adjuvant and neoadjuvant chemotherapy have been shown to significantly improve the survival of patients with advanced early-stage disease. The ability to prognosticate outcomes would allow us to manage these diseases better: patients whose cancer is likely to advance quickly or recur would receive the necessary treatment. The important challenge is to discover the molecular subtypes of disease and subgroups of patients. To address this important challenge, we develop a novel approach named Subtyping via Consensus Factor Analysis (SCFA) that can efficiently remove noisy signals from consistent molecular patterns in order to reliably identify cancer subtypes and accurately predict risk scores of patients. In an extensive analysis of 7,973 samples related to 30 cancers that are available at The Cancer Genome Atlas (TCGA), we demonstrate that SCFA outperforms state-of-the-art approaches in discovering novel subtypes with significantly different survival profiles. We also demonstrate that SCFA accurately predicts risk scores that strongly correlate with patient survival and vital status. More importantly, the accuracy of subtype discovery and risk prediction improves when more data types are integrated into the analysis.

More recently, advancements in single-cell RNA sequencing (scRNA-seq) have revolutionized our ability to study biological systems at the single-cell level. The widespread utilization of scRNA-seq across various research domains, such as cancer, immunology, and virology, has resulted in the generation of massive amounts of scRNA-seq data each year. However, the analysis of scRNA-seq data poses significant computational challenges due to the increasing number of cells and technical noise. First, scRNA-seq data is high-dimensional, with thousands of genes representing each cell. This poses difficulties in visualizing and comprehending the data. Analyzing relationships between thousands of genes and millions of cells, as required

for applications such as trajectory inference or gene regulatory network inference, can be computationally demanding and time-consuming. Second, scRNA-seq data is characterized by noise and sparsity, with numerous missing values and outliers. This makes it challenging to identify consistent patterns and trends, potentially leading to false positives or false negatives in the results. Third, technical noise is often introduced during the sample preparation and sequencing process, stemming from low starting material and amplification procedures. Such noise introduces inconsistencies in the data, hindering comparisons across different experiments.

To address the challenges associated with scRNA-seq data mining, we establish four innovative computational methods that effectively extract biological information from the noisy and massive single-cell data. First, we introduce an analysis framework, named single-cell Decomposition using Hierarchical Autoencoder (scDHA), that reliably extracts representative information of each cell. In one joint framework, the scDHA software package conducts cell segregation through unsupervised learning, dimension reduction and visualization, cell classification, and time-trajectory inference. Second, we develop three novel imputation methods: single-cell Imputation via Subspace Regression (scISR), single-cell Imputation using Neural Network (scINN), and single-cell Imputation using Residual Network (scIRN). These methods effectively recover missing data caused by dropout events in scRNA-seq data. We validate the performance of the four methods using extensive real-world data, including 43 scRNA-seq datasets with over a million cells. We demonstrate that the proposed methods outperform state-of-the-art techniques in several research sub-fields of scRNA-seq analysis, including cell segregation through unsupervised learning, visualization of transcriptome landscape, cell classification, and pseudo-time inference.

The dissertation is divided into three parts. In the first part, I introduce the significance of molecular subtype discovery and then detail the proposed method, SCFA, for cancer subtyping and risk prediction. In the second part, I provide an overview of single-cell data (scRNA-seq), together with the opportunities and the computational challenges. Next, I describe the four methods we developed for single-cell analysis, scDHA, scISR, scINN, and scIRN. Each method is accompanied with extensive validation and extensive analyses. In the third part, I summarize the dissertation and discuss future research directions that I will potentially pursue.

Dedication

I would like to dedicate this Ph.D. dissertation to the individuals who have played instrumental roles in my life:

First and foremost, I am deeply indebted to my parents, Quy and Tu, for their unwavering love, unconditional support, and countless sacrifices they have made throughout my educational journey. Your faith in my abilities and constant encouragement have served as my pillars of strength. This dissertation stands as a testament to the unwavering belief you have always had in me.

I am also grateful to my sister, Nga, and my brother, Nhat, for their continuous love, support, and encouragement. Your presence in my life has been a consistent source of inspiration and motivation.

Furthermore, I would like to extend my dedication to all those who have touched my life in ways that words cannot fully express. Your kindness, encouragement, and unwavering belief in my capabilities have left a profound impact, and I am genuinely grateful for that.

This dissertation represents the culmination of the collective support, guidance, and encouragement I have received throughout my academic journey. To all those mentioned by name and those unmentioned, I extend my heartfelt appreciation for being integral parts of this endeavor.

Acknowledgments

I would like to extend my sincere gratitude to my advisor, Dr. Tin Nguyen, for his unwavering guidance, support, and encouragement throughout my doctoral studies. I am truly grateful for his mentorship, patience, and steadfast belief in my abilities. His passion for research and dedication to academic excellence have served as constant sources of inspiration, motivating me to strive for greatness. Additionally, I deeply appreciate his invaluable insights, thought-provoking discussions, and intellectual challenges that have expanded my horizons.

I would also like to express my gratitude to my esteemed committee members, Dr. Frederick Harris, Dr. Hung La, Dr. Lei Yang, and Dr. David Cantu, for their indispensable feedback and suggestions. I am thankful for the time and effort they dedicated to reviewing my dissertation and providing constructive input. Their expertise and guidance have played an instrumental role in the successful completion of this dissertation.

Finally, I want to extend my heartfelt thanks to my colleagues at the University of Nevada, Reno, especially Phi Hung Bya, Bang Sy Tran, and Ha Viet Nguyen, who are also my dear friends. Their camaraderie, friendship, and unwavering support have made this journey more enjoyable and memorable.

Contents

List of Tables

List of Figures

Part I

Machine Learning in Cancer Subtype Discovery and Risk Prediction of Patients

Chapter 1

Cancer Subtyping: Significance and Challenges

After 20 years of cancer screening, the chance of a person being diagnosed with prostate or breast cancer has nearly doubled [1–4]. However, this has only marginally reduced the number of patients with advanced disease, suggesting that screening has resulted in the substantial harm of excess detection and over-diagnosis. At the same time, 30-50% of patients with non-small cell lung cancer (NSCLC) develop recurrence and die after curative resection [5], suggesting that a subset of patients would have benefited from more aggressive treatments at early stages. Although not routinely recommended as the initial course of treatment, adjuvant and neoadjuvant chemotherapy have been shown to significantly improve the survival of patients with advanced early-stage disease [6–8]. The ability to prognosticate outcomes would allow us to manage these diseases better: patients whose cancer is likely to advance quickly or recur would receive the necessary treatment. The important challenge is to discover the molecular subtypes of disease and subgroups of patients [9–12].

Cluster analysis has been a basic tool for subtype discovery using gene expression data. These include hierarchical clustering (HC), neural networks [13–17], mixture

model [18–20], matrix factorization [21, 22], and graph-theoretical approaches [23–25]. Arguably, the state-of-the-art approach in this area is Consensus Clustering (CC) [26, 27], which is a resampling-based methodology of class discovery and cluster validation [28–30]. However, these approaches are not able to combine multiple data types. Although analyses on a single data type could reveal some distinct characteristics for different subtypes, it is not sufficient to explain the mechanism that happens across multiple biological levels.

With the advancement of multi-omics technologies, recent subtyping methods have shifted toward multi-omics data integration. The goal is to differentiate among subtypes from a holistic perspective, that can take into consideration phenomena at various levels (e.g., transcriptomics, proteomics, epigenetics). These methods can be grouped into three categories: simultaneous data decomposition methods, joint statistical models, and similarity-based approaches. Methods in the first category (data decomposition) include md-modules [31], intNMF [32], and LRAcluster [33]. These methods assume that there exist molecular patterns that are shared across multiple types of data. Therefore, these methods aim at finding a low dimensional representation of the high-dimensional multi-omics data that retains those patterns. For example, both md-modules and intNMF utilize a joint non-negative matrix factorization to simultaneously factorize the data matrices of multiple data types. In their design, the basis vectors are shared across all data types while the coefficient matrices vary from data type to data type. These two methods, md-modules and intNMF, only differ in the way they iteratively estimate the coefficient matrices. Another method is LRAcluster, which applies the low-rank approximation and singular vector decomposition to generate low dimensional representations of the data and then performs k-means clustering to identify the subtypes. These methods strongly rely on the assumption that all molecular signals can be linearly and simultaneously

reconstructed.

Methods in the second category (statistical modeling) include BCC [34], MDI [35], iClusterBayes [36], iClusterPlus [37], and iCluster [38, 39]. These methods assume that each data type follows a mixture of distributions and then integrate multiple types of data using a joint statistical model. The parameters of the mixture models are estimated by maximizing the likelihood of observed data. These methods strongly depend on the correctness of their statistical assumptions. Also, due to a large number of parameters and iterations involved, the computation complexity of statistical methods is usually extensive. Therefore, these methods often rely on pre-processing and gene filtering to ease the computational burden.

Methods in the third category (similarity-based) typically construct the pair-wise connectivity between patients (that represents how often the patients are grouped together) for each data type and then integrate multiple data types by fusing the individual connectivity matrices. As these methods perform data integration in the sample space, their computational complexity depends mostly on the number of patients, not the dimensions of features/genes. Therefore, these methods are capable of performing subtyping on a genomic scale. Methods in this category include SNF [40], rMKL-DR [41], NEMO [42], CIMLR [43], and PINS [44, 45]. SNF creates a patient-to-patient network by fusing connectivity matrices and then partitions the network using spectral clustering [46]. rMKL-DR projects samples into a lower-dimensional subspace and then partitions the patients using k-means. NEMO follows a similar strategy with the difference is that it incorporates only partial data into the integrative analysis. Though powerful, these methods do not account for the noise and unstable nature of quantitative assays. PINS and CIMLR follow two different strategies to address noise and instability. PINS introduces Gaussian noise to the data in order to obtain subtypes that are robust against data perturbation. CIMLR combines

multiple gaussian kernels per data type to measure the similarity between each pair of samples. The resulted similarity matrix is then subjected to dimension reduction and k-means to determine the subtypes. Though powerful, the similarity metrics used in these methods (i.e., Gaussian kernel, Euclidean distance) make them susceptible to noise and the "curse of dimensionality"[47] from the high-dimensional multi-omics data.

Chapter 2

SCFA: A Novel Method for Cancer Subtyping and Risk Prediction Using Consensus Factor Analysis

This chapter is based on the following publication: ***Duc Tran****, Hung Nguyen, Uyen Le, Hung N. Luu, and Tin Nguyen. A novel method for cancer subtyping and risk prediction using consensus factor analysis. Frontiers in Oncology, 2020. DOI: 10.3389/fonc.2020.01052*

To address the challenges in cancer subtyping, we develop a novel approach named Subtyping via Consensus Factor Analysis (SCFA) that can efficiently remove noisy signals from consistent molecular patterns in order to reliably identify cancer subtypes and accurately predict risk scores of patients. In an extensive analysis of 7,973 samples related to 30 cancers that are available at The Cancer Genome Atlas (TCGA), we demonstrate that SCFA outperforms state-of-the-art approaches in discovering novel subtypes with significantly different survival profiles. We also demonstrate that SCFA is able to predict risk scores that are highly correlated with true patient survival and

vital status. More importantly, the accuracy of subtype discovery and risk prediction improves when more data types are integrated into the analysis. The SCFA software and is publicly available on Bioconductor: `https://www.bioconductor.org/packages/SCFA/`.

2.1 Methodology

The high-level workflow of SCFA for subtyping is shown in Figure 2.1. The input of the subtyping module is a list of data matrices (e.g., mRNA, methylation, miRNA) in which rows represent patients while columns represent genes/features. For each matrix, the method first performs a filtering step using an autoencoder and then repeatedly performs factor analysis [48] to represent the data with different numbers of factors. By representing data with different numbers of factors, we can improve on situations where the projected data do not accurately represent the original data due to noise. Using an ensemble strategy, SCFA combines all of the factor representations to determine the final subtypes.

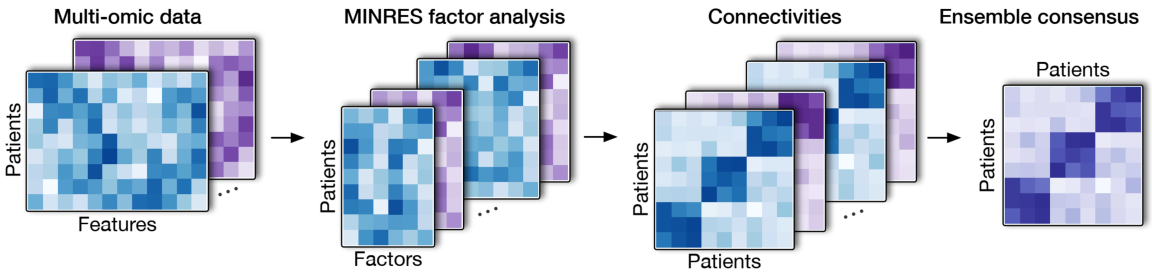

Figure 2.1: SCFA pipeline for cancer subtyping. For each of the data matrix, SCFA repeatedly performs factor analysis to generate multiple data representations with different numbers of factors. For each representation, SCFA clusters the data to construct a connectivity matrix. The method next merges all connectivity matrices using an ensemble strategy to obtain the final clustering.

In the following subsections, we will describe in detail the techniques used in the SCFA framework: (i) dimension reduction and factor analysis, and (ii) the ensemble

strategy for subtyping.

2.1.1 Dimension reduction and factor analysis

The method start with dimension reduction and factor analysis. The purpose of dimension reduction is to remove features/genes that play no role in differentiating between patients. Briefly, we utilize a non-negative kernel autoencoder which consists of two components: encoder and decoder. The encoder aims at representing the data in a low dimensional space whereas the decoder tries to reconstruct the original input from the compressed data. By forcing the weights of the network to be non-negative, we capture the positive correlation between the original features and the representative features. Selecting features with high variability in weights would result in a set of features that are informative, non-redundant, and capable of representing the original data.

After the filtering step using the non-negative autoencoder, we perform another dimension reduction step using Factor Analysis (FA) [48]. In general, factor analysis aims at minimizing the difference of feature-feature correlation matrix between the latent space and original data. Correlation is a standardized metric, where it takes into account the number of observations and variance of the features during the calculation process. This makes factor analysis robust against scaling and high number of dimensions compared to traditional decomposition such as principle component analysis (PCA), which uses Euclidean distance as the distance metric. To further improve the performance of factor analysis, we adjust the objective of FA to maintain the patient-patient correlation.

Starting with the original correlation matrix, FA finds k (number of factors) largest principle components and tries to reproduce the original matrix using those principal components (model matrix). FA iteratively fits the model matrix to the original ma-

trix using optimization algorithms. In our model, we employ the Minimum Residual (MINRES) optimization because it copes better with the small and medium sample size of the input data [49]. Also, instead of preserving the relationship between variables, we aim to maintain the overall patient-patient relationships by preserving their Pearson correlations in the representations. By changing the objective, the computational power required is significantly lower as the number of patients (in the scale of hundreds) is much lower than the number of features (in the scale of tens of thousands). Moreover, maintaining the distance between patients in the low dimensional representation would be more beneficial for our desired applications. To avoid overfitting, we repeatedly perform factor analysis with different numbers of factors, resulting in multiple representations of each input matrix. Clustering results using all factor representations of all data types (data matrices) are combined together using an ensemble strategy to determine the subtypes.

2.1.2 Subtyping using consensus ensemble

Given a collection of factor representations from all data types, we aim at finding patient subgroups that are consistently observed together in all representations (Figure 2.1). For each representation, we first determine the optimal number of clusters using two indices: (i) the ratio of *between sum of squares* over the *total sum of squares*, and (ii) the increase of *within sum of squares* when the number of cluster increases. After the optimal number of clusters is determined, we use k-means to cluster the underlying factor representation to build a connectivity matrix. To avoid the convergence to a local minimum, we perform k-means clustering using multiple starting points and choose the results with the smallest sum of square error. This process is repeated for all of the representations to obtain a collection of connectivity matrices for all data types.

Finally, we use the Weighted-based meta-clustering algorithm [50] to combine all clustering results from each data representation to determine the final subtyping. In short, the meta-clustering first calculates the weight for each pair of patients regarding their chance to be grouped together. Next, it assigns a weight for each patient by accumulating the weights of all pairs containing this patient. It then computes the weighted cluster-to-cluster similarity from all connectivity matrices. Finally, it partitions the cluster-to-cluster similarity matrix using hierarchical clustering to determine the final subtypes.

2.1.3 Risk score prediction

The goal of this module is to calculate the risk score of new patients using their molecular data. The high-level workflow for risk score prediction is shown in Figure 2.2. This is a supervised learning method that learns from a training set in order to predict the risk scores each patient in the testing set. More specifically, the training set consists of a set of patients with molecular data (e.g., mRNA, methylation, miRNA) and known survival information while the testing set consists of patients with only molecular data. By default, we provide TCGA datasets in our package as training data, but users are free to provide training data if necessary. Using the training data, this module will train the Cox regression model that can be used to predict the risk scores of new patients. Below is the description of the method for one data type and for multi-omics data.

Given a single data type as input, we merge the testing data with training data and then perform dimension reduction and factor analysis to generate multiple representations of this data. For each representation, we use the training data to train the Cox regression model. This model aims at estimating a coefficient β_i for each corresponding predictor x_i of the input data. After the model is trained, the risk

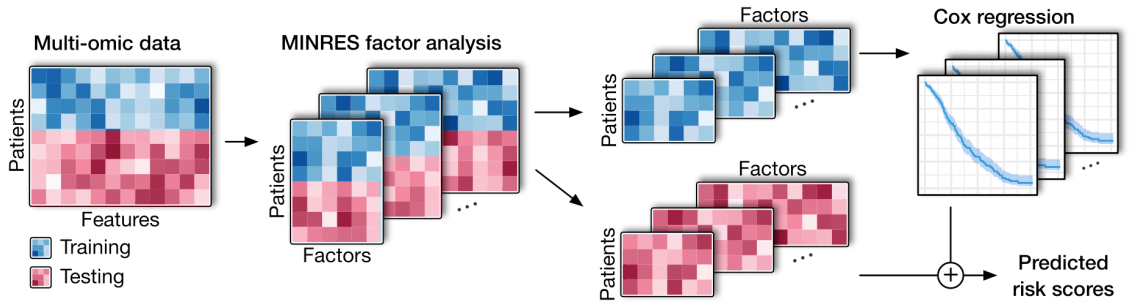

Figure 2.2: Overall pipeline for risk prediction using SCFA. The method will be able to learn from training data (patients with survival information) in order to predict risk scores of patients in testing data (patients without survival information). SCFA first merges training and testing sets together and then performs factor analysis. Using the factor representations of the training set, the method trains a Cox regression model, which will be utilized to predict risk factor of patients in the testing set

scores for new patients can be calculated as $\exp(\sum_{i=1}^{n} \beta_i x_i)$, where n is the number of features in the factor representation. In the Cox model, the risk score is defined as $\frac{h(t)}{h_0(t)}$, where $h(t)$ is the expected hazard at time t, and $h_0(t)$ is the baseline hazard when all the predictors are equal zero. Patients with a higher risk score are likely to suffer the event of interest (e.g., vital status or disease recurrence) earlier than the one with a lower risk score. Here we use elastic net [51] implemented in the R-package "glmnet" [52] to fit the model to better cope with the dynamic number of predictors. Elastic net linearly combines Lasso and Ridge penalty during the training process to select only the most relevant predictors that have important effects on the response (the risk scores in this case). We use five-fold cross-validation to select the parameters for the model. The final risk score for each patient is the geometric average of the risk scores resulted from all representations.

In the case of multi-omics data, we repeat the same process (described above) for each data type. We perform factor analysis to produce multiple representations, resulting in a collection of representations from all data types. For a new patient, each representation will produce an estimated risk score. The final risk score for the patient is calculated as the geometric average of all predictions from all representations.

2.2 Validation and Analysis Results

Here we assess the performance of SCFA using data obtained from 7,973 patients from 30 different cancer diseases downloaded from The Cancer Genome Atlas (TCGA). For each of the 30 cancer datasets, we downloaded mRNA, miRNA, and methylation data. Table 2.1 shows the details of each dataset. We also downloaded the clinical data for these patients, which includes vital status and survival information. Using clinical information, we comprehensively assess the ability of SCFA over existing methods in unsupervised subtyping, clinical variable association analysis, and supervised risk prediction.

2.2.1 Subtypting on 30 TCGA datasets

Here we compare the performance of SCFA with four state-of-the-art methods: Consensus Clustering (CC) [26, 27], Similarity Network Fusion (SNF) [40], Cancer Integration via Multikernel LeaRning (CIMLR) [43], and iClusterBayes (iCB) [36]. CC is a resampling-based approach, while SNF and CIMLR are graph-theoretical approaches. The fourth method, iClusterBayes is a model-based approach and is the enhanced version iClusterPlus. These methods were selected to represent three distinctively different subtyping strategies. Among these methods, CC is the only method that cannot integrate multiple data types. For CC, we concatenate the three data types for the integrative analysis. We demonstrate that SCFA outperforms these methods in identifying subtypes with significantly different survival profiles.

Note that here we focus on unsupervised learning, in which each dataset is partitioned independently without using any external information. For example, when analyzing the glioblastoma multiforme (GBM) dataset, we use only the molecular data (mRNA, miRNA, and methylation) of this dataset to determine the subtypes. For each cancer dataset, we first use each of the five methods (SCFA, CC, SNF,

Table 2.1: Description of 30 cancer datasets from The Cancer Genome Atlas (TCGA) that will be used for validation of the proposed method SCFA.

Dataset	#Samples	mRNA	Methylation	miRNA
ACC	79	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
BLCA	404	HiSeq RNASeq v2	Methylation450	GASeq miRNASeq
BRCA	622	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
CHOL	36	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
CESC	304	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
COAD	220	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
DBLC	47	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
ESCA	183	HiSeq RNASeq	Methylation450	HiSeq miRNASeq
GBM	273	HT HG-U133A	Methylation27	HiSeq miRNASeq
GBMLGG	510	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
HNSC	228	HiSeq RNASeq	Methylation450	HiSeq miRNASeq
KICH	65	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
KIPAN	654	HiSeq RNASeq	Methylation450	HiSeq miRNASeq
KIRC	124	HiSeq RNASeq	Methylation27	GASeq miRNASeq
KIRP	271	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
LAML	164	GASeq RNASeq	Methylation27	GASeq miRNASeq
LGG	510	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
LIHC	366	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
MESO	86	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
OV	286	HiSeq RNASeq v2	Methylation27	HiSeq miRNASeq
PAAD	178	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
SARC	257	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
SKCM	439	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
STES	545	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
TGCT	134	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
THCA	499	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
THYM	119	HiSeq RNASeq v2	Methylation450	GASeq miRNASeq
UCEC	234	GASeq RNASeq v2	Methylation450	HiSeq miRNASeq
UCS	56	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
UVM	80	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq

Table 2.2: Cox p-values of subtypes identified by SCFA, CC, SNF, iClusterBayes (iCB), and CIMLR for 30 TCGA datasets. The cells highlighted in yellow have Cox p-values smaller than 5%. In each row, cells highlighted in green have the most significant p-value. SCFA outperforms other methods by having significant p-values in most datasets (24 out of 30 datasets).

	SCFA	**CC**	**SNF**	**iCB**	**CIMLR**
ACC	3.4e-03	5.4e-04	4.3e-05	9.2e-04	3.4e-01
BLCA	7.2e-03	1.1e-01	1.1e-01	5.1e-01	4.7e-01
BRCA	3.2e-04	2.9e-02	1.2e-01	2.7e-02	4.9e-03
CESC	9.4e-03	5.8e-02	5.1e-01	2e-02	1.9e-01
DLBC	4.3e-06	5.1e-01	7.5e-01	2.9e-01	7.4e-01
ESCA	7.3e-05	7.7e-01	3.9e-01	7.9e-01	5.6e-01
GBM	2.3e-03	3.2e-01	2.1e-02	1.1e-01	8.1e-02
GBMLGG	5.8e-14	1.6e-04	4.8e-14	8e-02	6.4e-10
HNSC	4e-02	5e-01	3.7e-01	3.7e-01	4e-01
KICH	2.3e-13	8.7e-01	7e-01	6.9e-01	4.6e-01
KIPAN	1.4e-19	9.3e-08	2.1e-07	1.6e-09	9.8e-05
KIRP	1.7e-03	4.5e-01	5.3e-03	3e-03	1.9e-02
LAML	5.8e-04	3.9e-02	1.7e-03	9e-01	1.4e-04
LGG	6.5e-15	6.6e-07	1.6e-14	1.1e-01	8.3e-15
MESO	1.6e-04	3.1e-01	4.2e-04	3.7e-02	1.1e-02
PAAD	6.9e-04	1.1e-02	7.4e-04	2.3e-03	2e-03
SARC	3.3e-03	2.4e-01	4.4e-02	4.3e-02	5.6e-02
SKCM	1.6e-03	6.3e-01	4.8e-01	8.4e-03	7.4e-05
STES	3.9e-02	2e-01	1.6e-01	4.1e-03	3.4e-02
THCA	7.8e-03	7.9e-01	6.2e-01	7.8e-01	8.6e-03
THYM	8.1e-04	1.5e-01	9.7e-02	9e-03	1.2e-01
UCEC	6.5e-03	8.9e-02	1.8e-02	5.9e-02	4.6e-02
UCS	3.4e-02	1.6e-01	8.6e-01	9.6e-01	3.6e-01
UVM	1.3e-06	6.1e-04	1.7e-04	6.6e-02	5.8e-04
CHOL	3.1e-01	7.9e-02	5.7e-01	9.1e-01	3.4e-01
COAD	4.7e-01	5.8e-01	1.3e-01	2.2e-01	5.6e-01
KIRC	1e-01	8.3e-01	6.9e-01	8.3e-01	9.1e-02
LIHC	3.8e-01	8.8e-01	3.3e-01	9.3e-02	1.9e-01
OV	4.2e-01	6.1e-01	4.4e-01	4.6e-01	5.4e-01
TGCT	3.9e-01	7.4e-01	8.4e-01	7.1e-01	8.4e-01
#Significant	24	8	12	11	13

CIMLR, and iClusterBayes) to integrate the molecular data (mRNA, miRNA, and methylation) in order to determine patient subgroups. For each method, we calculate the Cox p-value that measures the statistical significance in survival differences between the discovered subtypes. The Cox p-values of subtypes discovered by the five methods for the 30 datasets are shown in Table 2.2. Among the 30 datasets, there are 6 datasets (CHOL, COAD, KIRC, LIHC, OV, and TGCT) for which no method is able to identify subtypes with significant survival differences. In the remaining 24 datasets, SCFA is able to obtain significant Cox p-values in all of them while CC, SNF, iClusterBayes, and CIMLR have significant p-values in only 8, 12, 11, and 13 datasets, respectively. Also, SCFA has the most significant p-values in 19 out of 24 datasets. Regarding time complexity, SCFA, CC, SNF, and CIMLR are able to analyze each dataset in minutes, whereas iClusterBayes can take up to hours to analyze a dataset.

To better understand the usefulness of data integration, we also calculated the Cox p-values obtained from individual data types and compared them to Cox p-values obtained from data integration (when mRNA, miRNA, and methylation are analyzed together). For each dataset, we perform subtyping using SCFA for each data type and report the Cox p-value of the discovered subtypes. The distributions of Cox p-values for data integration and for individual data types using SCFA are shown in Figure 2.3. Among 30 cancer datasets, the Cox p-values obtained from data integration has the median $-log10(p)$ of 2.6, compared to 1.7, 1.1, and 1.1 from gene expression, methylation and miRNA data. Interestingly, subtypes discovered using gene expression data have significantly different survival in 18 over 30 datasets, compared to 10 and 14 of methylation and miRNA data, respectively. The figure also shows that the Cox p-values obtained from gene expression data are more significant than those obtained from methylation and miRNA data ($p = 0.046$ using one-sided

Wilcoxon test). However, we note that miRNA and methylation also provide valuable information in data integration, when all data types are analyzed together. As shown in Figure 2.3, the Cox p-values obtained from data integration are more significant than those of any individual data type (including mRNA) with a one-sided Wilcoxon test p-value of 0.004. This means that each of the three data types provides meaningful contributions to the data integration.

To understand how other methods perform with respect to each data type, we also plot the distributions of Cox p-values obtained from each data type using CC, SNF, iClusterBayes, and CIMLR (Figure 2.4). CC is the only method that produces comparable Cox p-values across the three data types. SNF and CIMLR perform better using miRNA, while iClusterBayes favors mRNA and miRNA data.

2.2.2 Discovered subtypes and clinical variables

There are four important clinical variables that are available in more than 10 TCGA datasets: age (21 datasets), gender (25 datasets), cancer stages (24 datasets), and tumor grades (12 datasets). To understand the association between these variables and the discovered subtypes, we perform the following analyses: (1) Fisher's exact test to assess the association between gender (male and female) and the discovered subtypes; (2) ANOVA test to assess the age difference between the discovered subtypes; and finally (3) calculate the agreement between the discovered subtypes and known cancer stages/tumor grades using Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) (see Appendix A).

Figure 2.5 shows the p-value distribution for gender, age, and survival analysis (Cox p-value). Tables 2.3 and 2.4 show the p-values obtained for gender and age, respectively. The four methods, SCFA, CC, SNF, and CIMLR, are not biased toward gender with only some significant p-values. In contrast, iClusterBayes is subject to

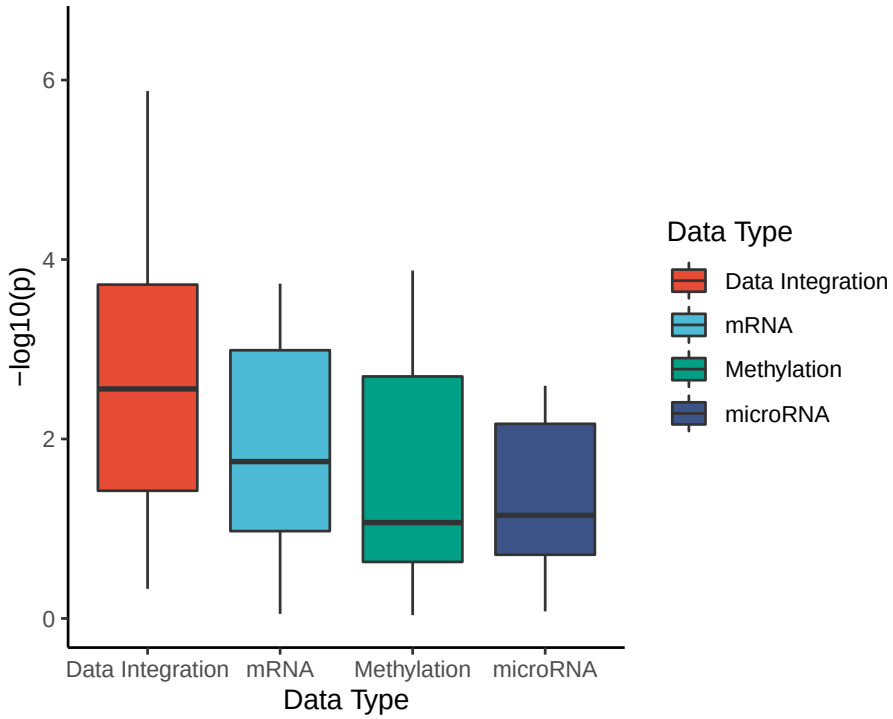

Figure 2.3: Cox p-values of subtypes identified by SCFA. To better understand the usefulness of data integration, we calculate the Cox p-values obtained from individual data types and compared them to Cox p-values obtained from data integration (when mRNA, miRNA, and methylation are analyzed together). The horizontal axis shows the data types while the vertical axis shows the minus $log10$ p-values. Overall the Cox p-values obtained from data integration are significantly smaller than those obtained from individual data types ($p = 0.004$ using one-sided Wilcoxon test).

gender bias with significant p-values in 12 out of 25 datasets (Table 2.3). Regarding age, all methods have comparable p-values (Table 2.4).

Figure 2.6 and Table 2.5 show the ARI values that represent the agreement between the discovered subtypes and known cancer stages and tumor grades. The median ARI of SCFA and SNF are comparable and they are higher than those of CC, iClusterBayes, and CIMLR. Regarding tumor grade, the ARI values of SCFA are higher than the rest. Figure 2.7 and Table 2.6 shows the NMI values. SCFA has higher NMI values in both comparisons. However, for both cancer stage and tumor grade, the ARI and NMI values of all methods are low, meaning that there is a low

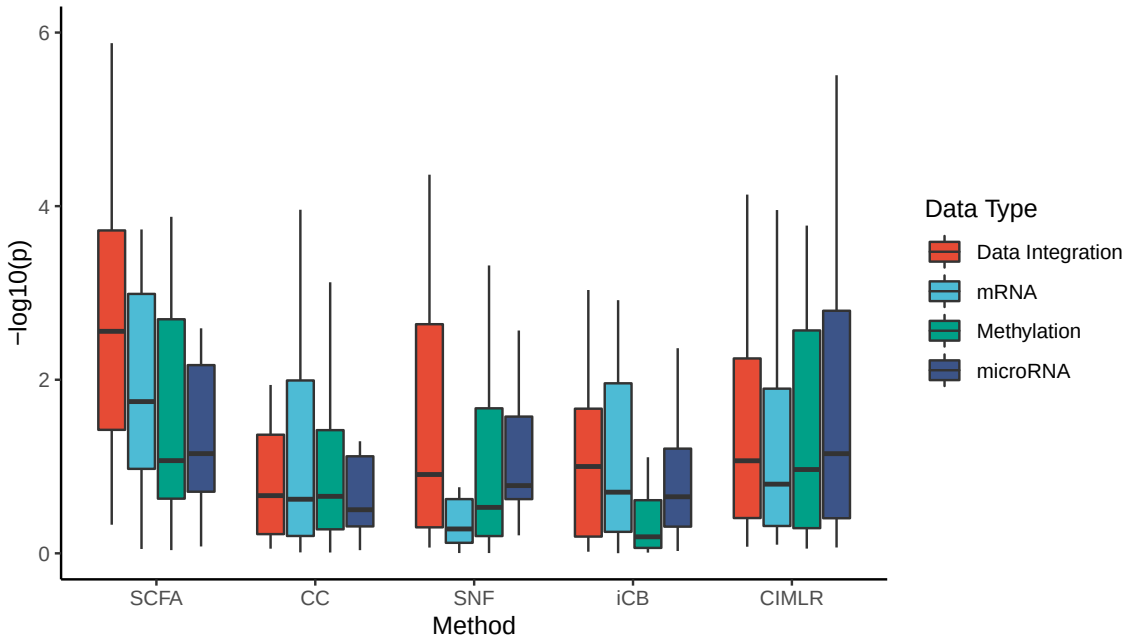

Figure 2.4: Cox p-values of subtypes identified by SCFA, SNF, iClusterBayes, CC, and CIMLR using different data types.

agreement between the known stages/grades and the discovered subtypes using any of the subtyping methods.

2.2.3 In-depth analysis of the Pan-Kidney (KIPAN) dataset

Figure 2.8 shows the Kaplan-Meier survival analysis [53] of the discovered subtypes using the KIPAN dataset. SCFA discovers five subtypes, each with a very different survival probability. Subtype 1 has the lowest survival rate while Subtype 5 has the highest survival rate. All patients of Subtype 1 die within three years whereas 85% of patients in Subtype 5 survive at the end of the study (after 15 years). Figure 2.9 shows the age distribution of each subtype, in which patients in Subtype 1 (low survival) are slightly older than patients in Subtype 5 (high survival) but there is no significant difference in age between the two groups. Patients in Subtypes 2, 3, and 4 are older than those of Subtype 1 (low survival) but they have higher survival probability.

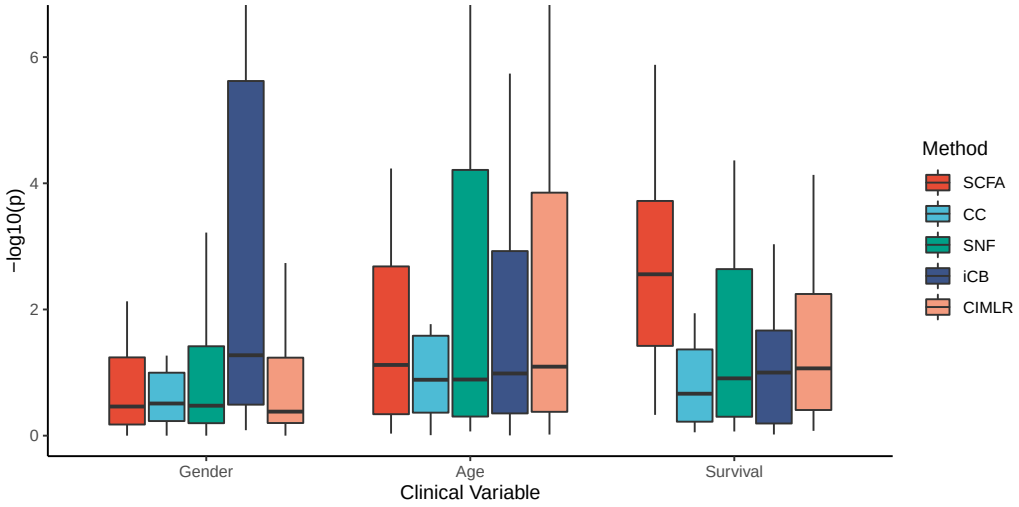

Figure 2.5: P-values obtained from comparing the discovered subtypes against gender, age, and survival information. Fisher's exact test was used to assess the statistical significance in the association between the discovered subtypes and gender while ANOVA was used to assess age difference. For survival analysis, Cox regression was used to assess the statistical difference in survival profiles. The horizontal axis shows the clinical variables while the vertical axis shows the minus $log10$ p-values.

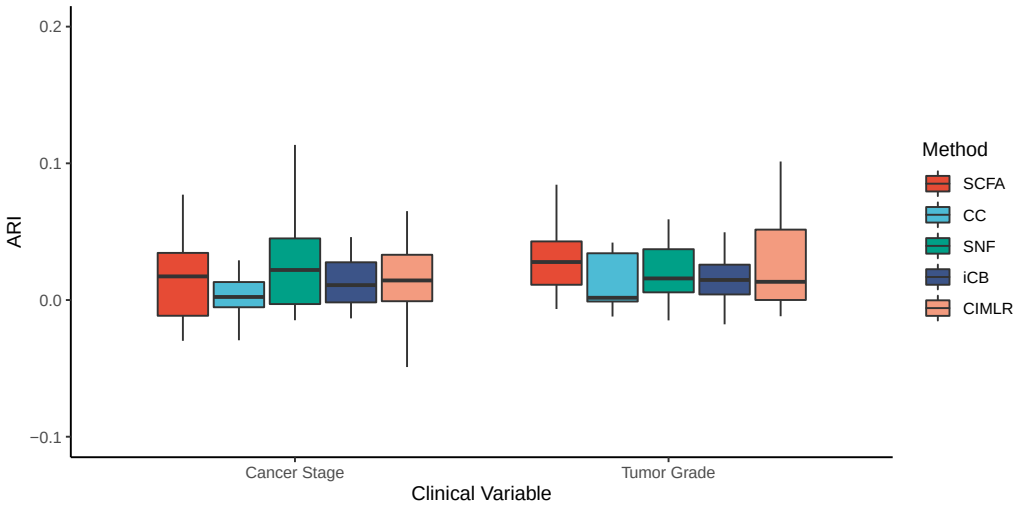

Figure 2.6: Adjusted Rand Index (ARI) values obtained from comparing the discovered subtypes against known cancer stages (left panel) and tumor grades (right panel).

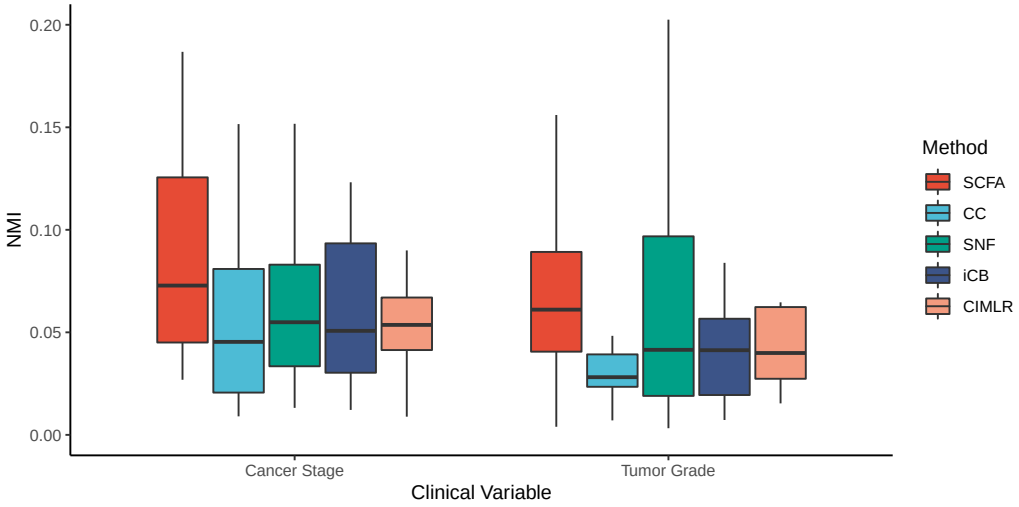

Figure 2.7: Normalized Mutual Information (NMI) values obtained from comparing the discovered subtypes against known cancer stages (left panel) and tumor grades (right panel).

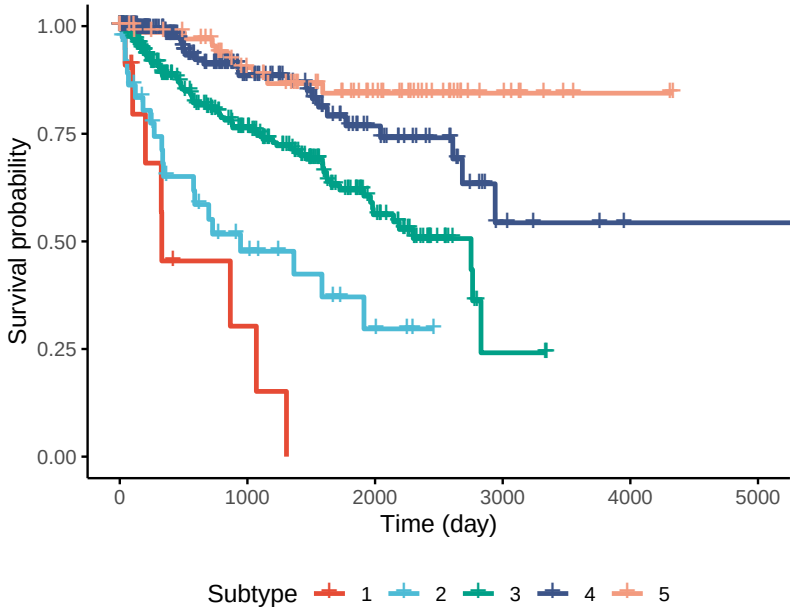

Figure 2.8: Kaplan-Meier survival analysis of the Pan-kidney (KIPAN) dataset. The horizontal axis represents the time (day) while the vertical axis represents the estimated survival probability.

Table 2.3: P-values obtained from Fisher's exact test that assesses the statistical significance of the association between the discovered subtypes and gender. NA indicates that there is not enough data to perform the test or all patients have the same gender. Cells highlighted in green have p-values smaller than the significance threshold of 0.05.

	SCFA	**CC**	**SNF**	**iCB**	**CIMLR**
ACC	4.5e-01	7.1e-01	1.1e-01	3.2e-01	4.9e-01
BLCA	2.4e-01	2.3e-01	2.6e-01	4.4e-01	4.1e-02
BRCA	7.4e-02	5.4e-02	2.1e-01	5.3e-02	9.0e-02
CESC	NA	NA	NA	NA	NA
CHOL	1.0e+00	1.1e-01	3.4e-01	1.6e-01	1.0e+00
COAD	5.7e-01	3.9e-01	7.5e-01	7.6e-45	4.2e-01
DLBC	1.6e-01	5.9e-01	1.0e+00	2.4e-06	3.1e-01
ESCA	8.8e-01	3.0e-01	1.0e+00	6.6e-01	8.3e-01
GBM	6.7e-01	4.6e-01	7.7e-01	8.1e-03	3.6e-01
GBMLGG	5.7e-06	4.9e-01	3.7e-01	6.2e-52	7.5e-01
HNSC	1.9e-01	1.9e-01	9.6e-03	8.2e-01	6.7e-01
KICH	6.5e-01	3.1e-01	2.0e-01	8.2e-03	1.0e+00
KIPAN	7.4e-03	7.0e-04	3.8e-02	7.1e-15	5.8e-02
KIRC	1.4e-02	2.3e-01	2.7e-01	5.8e-01	9.3e-15
KIRP	1.2e-02	1.0e+00	1.1e-03	5.5e-08	7.0e-05
LAML	8.0e-01	7.7e-01	4.3e-01	9.6e-02	6.3e-01
LGG	5.7e-02	1.0e-01	3.6e-01	3.9e-14	4.2e-01
LIHC	5.1e-01	9.0e-04	2.9e-05	3.1e-04	5.7e-06
MESO	2.2e-01	5.3e-01	7.6e-01	2.1e-01	5.8e-02
OV	NA	NA	NA	NA	NA
PAAD	1.0e+00	4.5e-03	6.7e-03	8.1e-01	1.4e-01
SARC	1.6e-05	5.4e-03	2.5e-05	5.6e-12	1.2e-03
SKCM	3.5e-01	6.7e-01	4.2e-01	1.2e-01	6.4e-01
STES	1.3e-02	5.9e-02	6.0e-04	9.9e-05	1.8e-03
TGCT	NA	NA	NA	NA	NA
THCA	8.7e-01	7.9e-01	3.7e-01	4.8e-01	4.6e-01
THYM	6.8e-01	5.7e-01	6.3e-01	5.5e-06	5.3e-01
UCEC	NA	NA	NA	NA	NA
UCS	NA	NA	NA	NA	NA
UVM	5.0e-01	1.0e+00	1.0e+00	6.2e-02	5.2e-01

Table 2.4: P-values obtained from ANOVA that assesses statistical significance in age difference between the discovered subtypes. Cells highlighted in green have p-values smaller than the significance threshold of 0.05.

	SCFA	**CC**	**SNF**	**iCB**	**CIMLR**
ACC	NA	NA	NA	NA	NA
BLCA	2.1e-03	1.5e-01	6.6e-03	4.6e-03	2.5e-02
BRCA	1.3e-02	6.2e-02	2.0e-01	6.7e-05	1.4e-04
CESC	4.6e-01	4.0e-02	3.8e-01	1.2e-07	1.3e-01
CHOL	NA	NA	NA	NA	NA
COAD	9.3e-01	4.3e-01	5.4e-01	6.4e-02	3.1e-01
DLBC	8.0e-01	2.4e-01	8.6e-01	4.9e-01	8.3e-01
ESCA	NA	NA	NA	NA	NA
GBM	5.8e-05	3.1e-02	1.4e-02	2.0e-05	2.9e-02
GBMLGG	9.5e-13	1.7e-02	1.2e-17	1.1e-01	2.8e-16
HNSC	1.5e-01	1.1e-01	5.3e-01	9.2e-01	4.2e-01
KICH	3.2e-01	1.3e-01	3.0e-01	4.4e-01	8.1e-02
KIPAN	3.0e-08	1.3e-06	1.2e-08	1.3e-01	8.8e-08
KIRC	1.9e-01	6.8e-01	6.1e-01	9.9e-01	6.4e-01
KIRP	3.7e-03	2.9e-01	2.3e-01	1.0e-01	9.6e-01
LAML	6.9e-03	2.4e-06	6.1e-05	7.9e-02	5.2e-06
LGG	4.3e-11	3.8e-04	1.9e-18	3.4e-01	3.9e-16
LIHC	6.4e-01	2.1e-05	3.3e-05	9.3e-04	1.9e-03
MESO	NA	NA	NA	NA	NA
OV	1.9e-02	4.7e-01	1.3e-01	1.8e-06	2.1e-01
PAAD	7.6e-02	9.8e-01	5.0e-01	1.7e-01	5.5e-01
SARC	NA	NA	NA	NA	NA
SKCM	1.5e-01	8.8e-01	6.1e-03	1.2e-03	1.1e-01
STES	6.1e-01	2.6e-02	5.1e-01	4.5e-01	8.8e-01
TGCT	NA	NA	NA	NA	NA
THCA	5.8e-01	2.7e-01	9.5e-02	6.0e-01	1.3e-02
THYM	NA	NA	NA	NA	NA
UCEC	1.6e-03	6.0e-01	1.3e-07	8.4e-03	1.4e-04
UCS	NA	NA	NA	NA	NA
UVM	NA	NA	NA	NA	NA

Table 2.5: Adjusted Rand Index (ARI) values obtained from comparing the discovered subtypes against known cancer stages and tumor grades.

	Cancer Stage					Tumor Grade				
	SCFA	**CC**	**SNF**	**iCB**	**CIMLR**	**SCFA**	**CC**	**SNF**	**iCB**	**CIMLR**
ACC	0.05	0.02	0.07	0.04	0.02	NA	NA	NA	NA	NA
BLCA	0.02	0	0.03	0.03	0.02	0.03	-0.01	0.04	0.01	0.02
BRCA	-0.02	0	-0.01	0	0.01	NA	NA	NA	NA	NA
CESC	0	0.01	-0.01	0	0.02	0	0	0.01	0.03	0
CHOL	-0.02	-0.02	-0.01	0	-0.02	NA	NA	NA	NA	NA
COAD	-0.03	-0.01	0	-0.01	0	NA	NA	NA	NA	NA
DLBC	-0.02	0.01	0	0.05	-0.05	NA	NA	NA	NA	NA
ESCA	0.08	0.08	0.07	0	0.07	NA	NA	NA	NA	NA
GBM	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
GBMLGG	NA	NA	NA	NA	NA	0.03	0.04	0.04	0.01	0.05
HNSC	-0.03	0	-0.01	0	0	-0.01	0	0.01	0	0
KICH	0.05	0	0.11	0.05	0.04	NA	NA	NA	NA	NA
KIPAN	0.07	0.01	0.04	0.03	0.06	0.04	0	0.01	0.01	0.02
KIRC	0.02	-0.03	-0.01	-0.01	-0.01	0.06	0.02	0.02	0.05	-0.01
KIRP	0.03	0.01	0.15	0.02	0.1	NA	NA	NA	NA	NA
LAML	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
LGG	NA	NA	NA	NA	NA	0.03	0.03	0.04	0.02	0.05
LIHC	0	0	0	0.02	0.03	0.02	0	0.01	0.03	0.01
MESO	-0.01	-0.01	0.03	0	-0.02	NA	NA	NA	NA	NA
OV	0	0.02	0	-0.01	0.01	0.02	0	-0.01	-0.02	0.01
PAAD	0.1	-0.01	0.04	0.12	0.05	0.06	-0.01	0.06	0.06	0.05
SARC	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
SKCM	0.02	0	0.02	0.01	0.01	NA	NA	NA	NA	NA
STES	0.01	0.02	0.01	0.01	0.01	0	0.04	0	-0.02	-0.01
TGCT	0.03	0.05	0.03	0.02	0.03	NA	NA	NA	NA	NA
THCA	-0.01	0.01	0.01	0.01	0.02	NA	NA	NA	NA	NA
THYM	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
UCEC	0.01	-0.03	0.04	0.01	0.01	0.08	0.12	0.06	0.02	0.1
UCS	0.03	0	0.05	0.02	-0.03	NA	NA	NA	NA	NA
UVM	0.04	0.03	0.05	0.09	0.03	NA	NA	NA	NA	NA

Table 2.6: Normalized Mutual Information (MNI) values obtained from comparing the discovered subtypes against known cancer stages and tumor grades.

	Cancer Stage					Tumor Grade				
	SCFA	**CC**	**SNF**	**iCB**	**CIMLR**	**SCFA**	**CC**	**SNF**	**iCB**	**CIMLR**
ACC	0.12	0.06	0.1	0.11	0.06	NA	NA	NA	NA	NA
BLCA	0.05	0.02	0.03	0.04	0.03	0.06	0.05	0.1	0.06	0.05
BRCA	0.03	0.01	0.02	0.02	0.04	NA	NA	NA	NA	NA
CESC	0.04	0.06	0.03	0.05	0.06	0.03	0.01	0.01	0.04	0.04
CHOL	0.15	0.13	0.08	0.2	0.16	NA	NA	NA	NA	NA
COAD	0.08	0.05	0.06	0.05	0.06	NA	NA	NA	NA	NA
DLBC	0.11	0.09	0.07	0.1	0.02	NA	NA	NA	NA	NA
ESCA	0.13	0.12	0.09	0.09	0.09	NA	NA	NA	NA	NA
GBM	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
GBMLGG	NA	NA	NA	NA	NA	0.06	0.03	0.06	0.01	0.06
HNSC	0.05	0.02	0.01	0.03	0.05	0.06	0.04	0.02	0.03	0.03
KICH	0.19	0.12	0.1	0.12	0.04	NA	NA	NA	NA	NA
KIPAN	0.07	0.05	0.06	0.03	0.05	0.04	0.03	0.02	0.06	0.02
KIRC	0.04	0.07	0.04	0.01	0.01	0.12	0.11	0.1	0.08	0.04
KIRP	0.08	0.02	0.1	0.02	0.07	NA	NA	NA	NA	NA
LAML	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
LGG	NA	NA	NA	NA	NA	0.09	0.03	0.06	0.01	0.06
LIHC	0.03	0.02	0.02	0.03	0.03	0.01	0.01	0.01	0.04	0.03
MESO	0.09	0.04	0.02	0.05	0.06	NA	NA	NA	NA	NA
OV	0.05	0.02	0.05	0.02	0.04	0.05	0.03	0.02	0.02	0.02
PAAD	0.13	0.04	0.08	0.11	0.06	0.09	0.03	0.1	0.08	0.06
SARC	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
SKCM	0.04	0.02	0.03	0.04	0.05	NA	NA	NA	NA	NA
STES	0.06	0.05	0.05	0.05	0.05	0	0.02	0	0.01	0.02
TGCT	0.12	0.12	0.09	0.09	0.08	NA	NA	NA	NA	NA
THCA	0.03	0.03	0.03	0.03	0.05	NA	NA	NA	NA	NA
THYM	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
UCEC	0.05	0.04	0.04	0.07	0.07	0.16	0.07	0.2	0.06	0.15
UCS	0.29	0.15	0.15	0.21	0.11	NA	NA	NA	NA	NA
UVM	0.13	0.08	0.08	0.08	0.11	NA	NA	NA	NA	NA

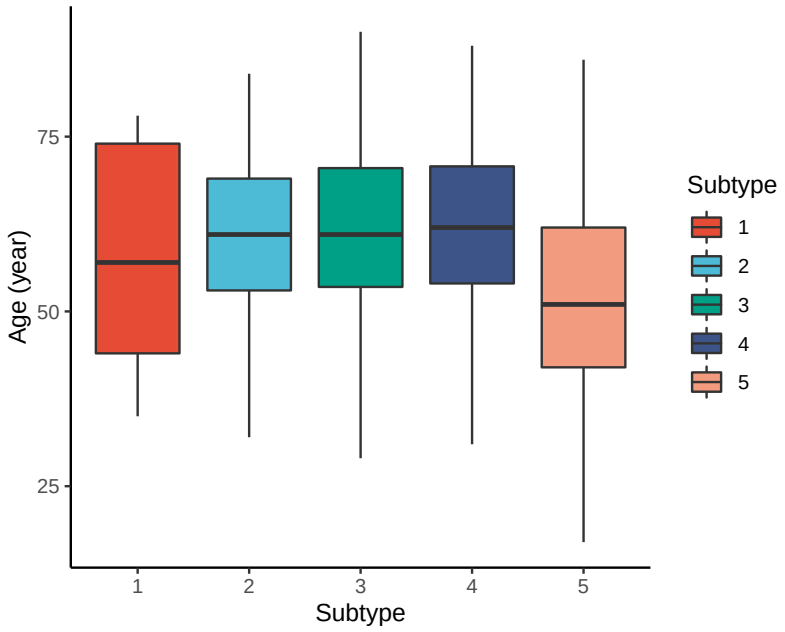

Figure 2.9: Age distribution for each subtype of the KIPAN dataset.

To show the molecular signature of each subtype, we also plot the heatmaps that visualize different subtypes of KIPAN patients on important genes/features. For each data type, we calculate the p-value for each feature using ANOVA and then choose 20 features/genes with the most significant p-value. Figure 2.10 shows the heatmap for mRNA (left panel), methylation (middle panel) and miRNA (right panel). The methylation data clearly differentiates Subtype 5 (highest survival probability) from the rest. In the listed probes (DNA regions), Subtype 5 has a consistently low level of methylation compared to other subtypes. However, methylation data alone cannot differentiate among the rest of the patients (Subtypes 1, 2, 3, and 4). Using information from mRNA and miRNA, SCFA can further divide the rest of the patients into four subtypes with very different survival profiles.

We also perform variant analysis to look for mutations that are highly abundant in the short-term survival groups but not in the long-term survival groups, as shown in Figure 2.11. In this figure, each point represents a gene and its coordinates are

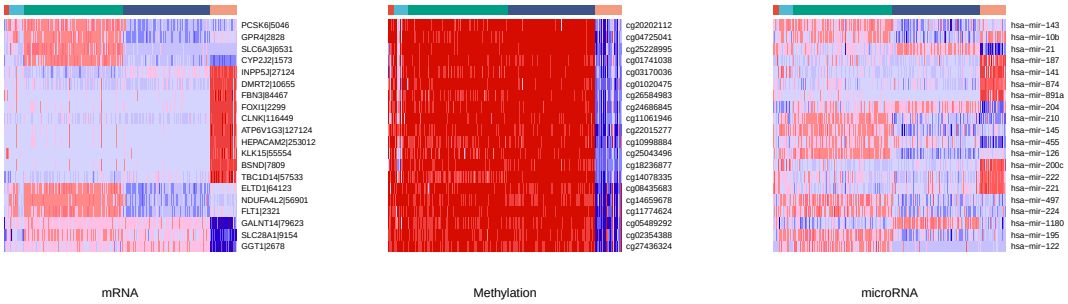

Figure 2.10: Heatmap of subtypes discovered by SCFA for the KIPAN dataset.

the number of patients having at least a variant in that gene in each group. In principle, we would look for mutated genes in the top left and the bottom right corners. From this figure, we can identify four notable markers: VHL, PBRM1, MUC4, and FRG1B. Among these, MUC4 is known to be associated with exophytic growth of clear cell renal cell carcinoma [54]. VHL has been reported to be linked to a primary oncogenic driver in kidney cancers [55]. Functional studies show that HIF is sufficient for transformation caused by loss of VHL, thereby establishing HIF as the primary oncogenic driver in kidney cancers. PBRM1 is also a major clear cell renal cell carcinoma (ccRCC) gene [56].

2.2.4 Risk score prediction using multi-omics data

We also use the same set of data to demonstrate the ability of SCFA in predicting risk score of each patient. For each of the TCGA datasets, we randomly split the data into two equal sets of patients: a training set and a testing set. We use the training set to train the model and then predict the risk for patients in the testing set. The predicted risk scores are then compared with the true vital status and survival information using Cox p-value and concordance index (C-index) [57]. Concordance index represents the probability that, for a pair of randomly chosen patients, the patient with higher predicted risk will experience death event before the other patient. On the other hand, Cox p-value measures how significant the difference in survival when correlating with

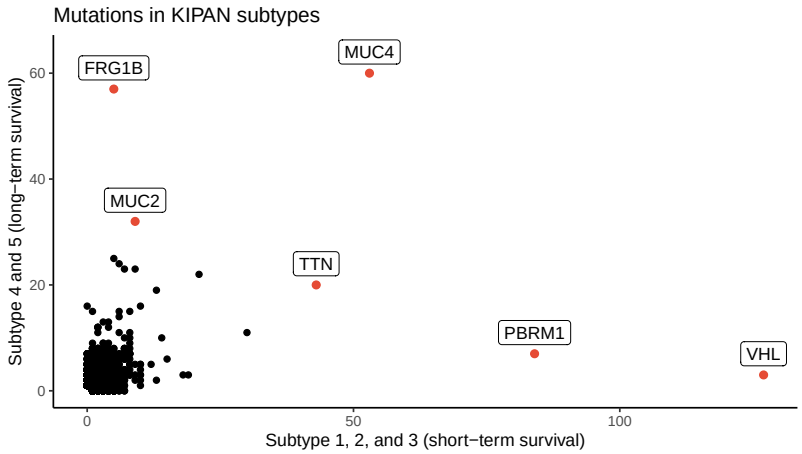

Figure 2.11: Number of patients in each group for each mutated gene for KIPAN. The horizontal axis represents the count in subtypes with low survival rate (subtype 1, 2, and 3), while the vertical axis shows the count for subtypes with high survival (subtype 4 and 5) rate.

predicted risk scores. This process is repeated 20 times for each dataset, and the average C-index and $-log10(p)$ for each dataset are calculated using results from these 20 runs. We note that some datasets do not have enough patients with either event (survive or death), which leads to errors for Cox regression. For that reason, we removed five datasets (DLBC, KIRP, TGCT, THYM, UCEC) from the analysis, and report survival prediction for only 25 datasets without errors.

Figure 2.12 shows the distributions of C-indices and Cox p-values (in minus log10 scale), while Table 2.7 shows the exact values calculated for each dataset. We calculate the C-index and Cox p-value obtained from individual data types and compared them to those obtained from data integration (when mRNA, miRNA, and methylation are analyzed together). As shown in Figure 2.12a, the accuracy of the prediction using data integration is generally higher than the accuracy obtained from individual data types. Predictions using data integration have a median C-index of 0.62, compared to 0.57, 0.54, and 0.57 when using mRNA, methylation, and miRNA, respectively. Similar results are also observed in the evaluation using Cox p-values (Figure 2.12b).

The Cox p-values obtained from data integration has the median $-log10(p)$ of 1.9, compared to 1.0, 0.7, and 0.9 for mRNA, methylation, and miRNA. The results demonstrate that we can potentially predict the risk score of each patient using only molecular data. More importantly, the prediction using multi-omics data is generally more accurate than using individual data types.

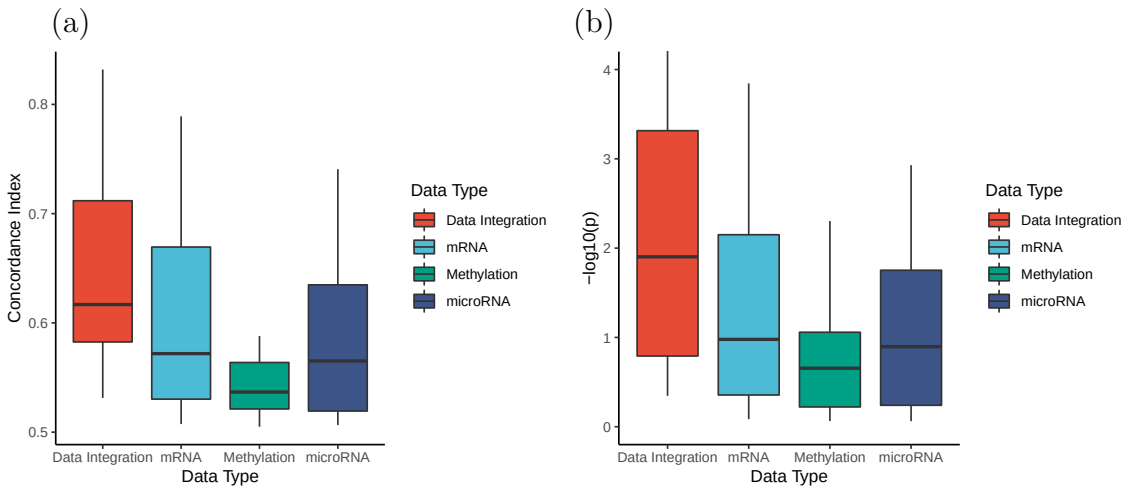

Figure 2.12: Evaluation of risk prediction using concordance index (C-index) and Cox p-values. For each dataset, we calculate the C-index and Cox p-values between predicted risk scores and known survival of patients. To better understand the usefulness of data integration, we calculate the C-index and Cox p-value obtained from individual data types and compared them to those obtained from data integration. (a) Distributions of C-indices for data integration and individual data types. (b) Distributions of Cox p-values for data integration and individual data types. SCFA is able to predict risk scores that are highly correlated to true survival with a median C-index of 0.62 and Cox p-value of 0.01. In addition, the prediction is more accurate when all data types are analyzed together. The C-indices are significantly higher and the p-values are significantly smaller when all data types are combined ($p = 0.0007$ and $p = 0.002$ using one-sided Wilcoxon test).

Table 2.7: Risk score prediction evaluated by concordance index (C-index) and Cox p-values.

Dataset	C-index				-log10(p)			
	Integration	**mRNA**	**Methylation**	**microRNA**	**Integration**	**mRNA**	**Methylation**	**microRNA**
ACC	0.78	0.79	0.59	0.72	3.32	3.84	0.66	2.73
BLCA	0.59	0.55	0.55	0.54	2.44	1.1	0.9	0.73
BRCA	0.62	0.55	0.52	0.51	1.38	0.77	0.28	0.14
CESC	0.68	0.63	0.54	0.64	3.42	2.15	1.4	2.02
CHOL	0.56	0.56	0.51	0.55	0.38	0.36	0.2	0.24
COAD	0.56	0.52	0.51	0.57	0.52	0.09	0.09	0.48
ESCA	0.53	0.52	0.5	0.51	0.35	0.09	0.18	0.06
GBM	0.55	0.51	0.53	0.53	2.44	0.3	1.04	1.12
GBMLGG	0.77	0.79	0.72	0.73	14.1	11.56	4.83	5.14
HNSC	0.59	0.59	0.51	0.55	1.41	1.81	0.22	0.48
KICH	0.68	0.6	0.63	0.57	1.35	0.62	2.3	1.31
KIPAN	0.79	0.77	0.73	0.74	24.42	14.53	11.65	20.54
KIRC	0.58	0.59	0.54	0.6	0.79	1.24	0.5	0.94
LAML	0.63	0.61	0.56	0.59	2.45	1.94	1.06	1.16
LGG	0.77	0.78	0.73	0.73	14.02	11.44	5.21	7.53
LIHC	0.62	0.53	0.55	0.57	1.9	0.36	0.86	0.9
MESO	0.72	0.69	0.53	0.63	4.46	3.72	0.22	2.93
OV	0.54	0.51	0.53	0.51	0.41	0.12	0.72	0.14
PAAD	0.71	0.67	0.56	0.59	3.35	2.58	0.79	1.75
SARC	0.62	0.57	0.53	0.53	1.19	0.98	0.19	0.26
SKCM	0.61	0.53	0.53	0.52	2.32	0.55	0.32	0.24
STES	0.54	0.51	0.52	0.51	0.4	0.11	0.29	0.16
THCA	0.66	0.53	0.54	0.51	1.26	0.44	0.33	0.57
UCS	0.58	0.53	0.51	0.51	0.68	0.15	0.06	0.08
UVM	0.83	0.67	0.69	0.72	2.62	1.14	2.87	1.33

2.3 Conclusion (SCFA)

We presented a novel method named SCFA for disease subtyping and risk assessment using multi-omics data. The contribution of SCFA is two-fold. First, it utilizes a robust dimension reduction procedure using autoencoder and factor analysis to retain only essential signals. Second, it allows researchers to predict risk scores of patients using multi-omics data – the attribute that is missing in current state-of-the-art subtyping methods.

To evaluate the developed method, we examined data obtained from 7,973 patients related to 30 cancer diseases downloaded from The Cancer Genome Atlas (TCGA). SCFA was compared against four state-of-the-art subtyping methods, CC, SNF, iClusterBayes, and CIMLR. We demonstrate that SCFA outperforms existing approaches in discovering novel subtypes with significantly different survival profiles. We also demonstrate that the method is capable of exploiting complementary signals available in different types of data in order to improve the subtypes. Indeed, the Cox p-values obtained from data integration are more significant than those obtained from individual data types.

To further demonstrate the usefulness of the developed method, we also performed a risk assessment using molecular data. We demonstrate that SCFA is able to predict risk scores that are highly correlated with vital status and survival probability. The correlation between predicted risk scores and survival information has a median of 0.62 and can be as high as 0.83. More importantly, we demonstrate that the risk prediction becomes more accurate when more data types are involved.

Part II

Single-cell RNA Sequencing (scRNA-seq): Data Mining of High-Dimensional, Large-scale Biological Data

Chapter 3

Mining scRNA-seq Data: Background, Significance, and Current Challenges

Bulk RNA sequencing (RNA-seq) has been the primary tool to study biological systems. Despite its popularity, bulk sequencing is unable to measure the heterogeneity inside complex tissues and cell-to-cell variability. This is due to the fact that the measurements from bulk sequencing technologies usually reflect the average gene expression across a cell population. Recent advances in microfluidic and sequencing technologies have allowed us to measure the expression profiles of individual cells [58, 59]. By allowing us to monitor the biological processes at the single-cell resolution, single-cell RNA sequencing technologies have enabled new research directions in genomics and transcriptomics research. These include a various atlas projects [60, 61] aiming at building the references of all cell types in model organisms, transcriptome landscape visualization in complex tissues [62, 63], inference of cell developmental trajectories [64], inferring gene regulatory network [65], *in silico*

cellular deconvolution [66, 67], and predicting cell spatial position [68, 69].

Such comprehensive decomposition of complex tissues holds enormous potential in both basic research and clinical applications [70–72]. By sequencing the RNA from individual cells, Single-cell RNA sequencing (scRNA-seq) is especially useful in fast-changing environments, such as tumor tissues or developing embryos. This allows researchers to see which genes are active in each cell, providing a more detailed and accurate picture of cellular function [58, 59]. scRNA-seq has also been used to identify new cell types, study the heterogeneity of cells in different tissues, and identify the mechanisms underlying diseases such as cancer. It has also been used to study environments with diverse composition, such as the microbiome.

The analysis of scRNA-seq data typically involves several steps. First, the raw data from the sequencing experiment must be processed and filtered to remove noise and low-quality data. This preprocessing can also include the removal of unwanted variation, such as batch effects or technical variation, or recovery of missing values due to the dropout phenomenon through data imputation. Next, the expression levels of genes in each cell must be normalized, allowing for comparison across cells. After the upstream analysis, scRNA-seq data usually is available as a table containing the gene expression levels for individual cells. Using this data, we can perform several downstream analyses, including clustering, visualization, classification, or pseudo-time inference, to extract useful biological insights.

Defining cell types through unsupervised learning, also known as cell segregation or clustering, is considered the most powerful application of scRNA-seq data analysis [73]. This has led to the creation of numerous atlas projects [60, 61], which aim to build the references for all cell types in various model organisms at multiple developmental stages. Widely-used methods in this category include SC3 [74], SEURAT [68], SINCERA [75], CIDR [76], and SCANPY [77]. Another fundamental

analysis of scRNA-seq data is the visualization of transcriptome landscape. Computational methods in this category aim at representing the high-dimensional scRNA-seq data in a low-dimensional space while preserving the relevant structure of the data. Non-linear methods [63], including Isomap [78], Diffusion Map [79], t-SNE [80], and UMAP [62], have been recognized as efficient techniques to avoid overcrowding due to the large number of cells, while preserving the local data structure. Among these, t-SNE is the most commonly used technique while UMAP and SCANPY are recent methods.

Once the cellular subpopulations have been determined and validated, classification techniques can be used to determine the composition of new datasets by classifying cells into discrete types. Dominant classification methods include XGBoost [81], Random Forest (RF) [82], Deep Learning (DL) [83], and Gradient Boosting Machine (GBM) [84]. Given the cell subpopulations information, researchers will be able to perform pseudo-time inference, which defines the biological progression of cells through their maturing stages [85]. This application, namely trajectory inference, computationally models multiple cellular processes, such as cell cycle, proliferation, differentiation, and activation [86, 87], by ordering the cells along developmental trajectories. Multiple trajectory inference tools have been developed, in which Monocle [88], TSCAN [89], Slingshot [64], and SCANPY [77] are considered state-of-the-art and are widely used for pseudo-temporal ordering.

Besides, scRNA-seq techniques have limitations including the high cost of the technology and the high rate of technical noise. This is because the process of isolating RNA from a single cell is very complex, and it is difficult to ensure that the RNA is not degraded during the process. This leads to a high rate of missing values in the data, which is commonly referred to as the dropout phenomenon. One way to address this issue is to use data imputation techniques to recover the missing values.

However, the imputation process can be challenging, and plausibly introduce bias into the data.

Despite its limitations, scRNA-seq remains a powerful tool with the potential to revolutionize our understanding of cellular biology. Its widespread utilization across various research domains, including cancer [90], immunology [91], or virology [92], has resulted in the massive amounts of scRNA-seq data being generated each year [93]. In addition, researchers have recently developed computational methods, known as cellular deconvolution, to obtain partial benefits of scRNA-seq analysis from existing bulk RNA-seq data. This innovative approach enables the inference of cell-type composition from bulk RNA-seq data, thereby facilitating the identification of cell types associated with diseases and other phenotypes.

Although scRNA-seq has gained wide popularity for studying the transcriptome of individual cells, several challenges persist in the analysis and interpretation of the data. Firstly, scRNA-seq data is high-dimensional, with thousands of genes representing each cell. This poses difficulties in visualizing and comprehending the data. Analyzing relationships between thousands of genes and millions of cells, as required for applications like trajectory inference or gene regulatory network inference, can be computationally demanding and time-consuming. Secondly, scRNA-seq data is characterized by noise and sparsity, with numerous missing values and outliers. This makes it challenging to identify consistent patterns and trends, potentially leading to false positives or false negatives in the results. Thirdly, technical noise is often introduced during the sample preparation and sequencing process, stemming from low starting material and amplification procedures. Such noise introduces inconsistencies in the data and hampers comparisons across different experiments. Lastly, scRNA-seq is expanding to measure additional modalities beyond gene expression, such as protein expression, chromatin accessibility, and DNA methylation. Integrat-

ing and interpreting these diverse modalities of data presents its own set of challenges. Despite these obstacles, scRNA-seq remains a powerful tool for studying cell biology. Ongoing technological advancements are expected to address these challenges, further enhancing the capabilities of scRNA-seq for researchers.

One of the main challenge for scRNA-seq computational approaches is the exponentially increasing in size of scRNA-seq dataset. Recently, it is becoming common for single-cell studies to generate and publish datasets with hundreds of thousands to millions of samples. Processing and analyzing this amount of data would prove to be a challenging problems. Moreover, due to the large number of genes in scRNA-seq datasets, the differences between cells in high dimensional space become more difficult to identify, this is known as the "curse of dimensionality"[47]. For researchers to be able to take full advantage of these rich datasets, efficient computational methods are required. Current computational approaches usually apply feature selection and/or dimension reduction techniques to reduce the noise and increase the scalability. Feature selection aims to indentify the most informative genes, for example ones with highest variance [94] or dispersion [95]. Dimension reduction methods, including PCA [74, 96], t-SNE [97], UMAP [98], random projection [50], and autoencoder [99, 100], are often used by scRNA-seq data analysis methods to project the data to lower dimensional space.

Another outstanding challenge is the "dropout" phenomenon where a gene is highly expressed in one cell but does not express at all in another cell [101]. These dropout events usually occur due to the limitation of sequencing technologies when only a small amount of starting mRNA in individual cells can be captured, leading to low sequencing depth and failed amplification [102, 103]. Since downstream analyses of scRNA-seq heavily rely on the accuracy of expression measurement, it is crucial to impute the zero expression values introduced by the dropout phenomenon and

sequencing errors.

To address the challenges mentioned above, we develop four novel methods for scRNA-seq data mining and interpretation. Chapter 4 describes a new analysis framework, called single-cell Decomposition using Hierarchical Autoencoder (scDHA), that can efficiently detach noise from informative biological signals. In one joint framework, the scDHA software package conducts cell segregation through unsupervised learning, dimension reduction and visualization, cell classification, and time-trajectory inference. We will show that scDHA outperforms state-of-the-art methods in all four sub-fields: cell segregation through unsupervised learning, transcriptome landscape visualization, cell classification, and pseudo-time inference. Chapter 5 describes a novel imputation method, named single-cell Imputation via Subspace Regression (scISR), that can reliably recover the dropout values of scRNA-seq data. We will show that scISR consistently improves the quality of cluster analysis regardless of dropout rates, normalization techniques, and quantification schemes. Chapter 6 describes a new approach, single-cell Imputation using Neural Network (scINN), that can reliably impute missing values from single-cell data. Chapter 7 describes another imputation approach, single-cell Imputation using Residual Network (scIRN), that can reliably impute missing values from single-cell data. We will demonstrate that scINN and scIRN outperform existing imputation methods (MAGIC [104], scImpute [105], SAVER [106], and DrImpute [107]) in improving the identification of cell subpopulations and the quality of biological landscape.

Chapter 4

scDHA: Fast and Precise Single-cell Data Analysis using a Hierarchical Autoencoder

This chapter is based on the following publication: ***Duc Tran****, Hung Nguyen, Bang Tran, Carlo La Vecchia, Hung N. Luu, and Tin Nguyen. Fast and precise single-cell data analysis using hierarchical autoencoder. Nature Communications. 2021. DOI: 10.1038/s41467-021-21312-2*

A primary challenge in single-cell RNA sequencing (scRNA-seq) studies comes from the massive amount of data and the excess noise level. To address this challenge, we introduce an analysis framework, named single-cell Decomposition using Hierarchical Autoencoder (scDHA), that reliably extracts representative information of each cell. The scDHA pipeline consists of two core modules. The first module is a non-negative kernel autoencoder able to remove genes or components that have insignificant contributions to the part-based representation of the data. The second module is a stacked Bayesian autoencoder that projects the data onto a low-dimensional space

(compressed). To diminish the tendency to overfit of neural networks, we repeatedly perturb the compressed space to learn a more generalized representation of the data. In an extensive analysis, we demonstrate that scDHA outperforms state-of-the-art techniques in many research sub-fields of scRNA-seq analysis, including cell segregation through unsupervised learning, visualization of transcriptome landscape, cell classification, and pseudo-time inference.

4.1 Introduction

Advances in microfluidics and sequencing technologies have allowed us to monitor biological systems at single-cell resolution [58, 59]. This comprehensive decomposition of complex tissues holds enormous potential in both developmental biology and clinical research [65, 108, 109]. Many computational methods have been developed to extract valuable information available in massive single-cell RNA sequencing data. These include methods for cell segregation, transcriptome landscape visualization, cell classification, and pseudo-time inference.

Defining cell types through unsupervised learning, also known as cell segregation or clustering, is considered the most powerful application of scRNA-seq data [73]. This has led to the creation of a number of atlas projects [60, 61], which aim to build the references of all cell types in model organisms at various developmental stages. Widely-used methods in this category include SC3 [74], SEURAT [68], SINCERA [75], CIDR [76], and SCANPY [77]. Another fundamental application of scRNA-seq is the visualization of transcriptome landscape. Computational methods in this category aim at representing the high-dimensional scRNA-seq data in a low-dimensional space while preserving the relevant structure of the data. Non-linear methods [63], including Isomap [78], Diffusion Map [79], t-SNE [80], and UMAP [62], have been recognized as efficient techniques to avoid overcrowding due to the large number of cells, while

preserving the local data structure. Among these, t-SNE is the most commonly used technique while UMAP and SCANPY are recent methods.

Visualizing transcriptome landscape and building comprehensive atlases are problems of unsupervised learning. Once the cellular subpopulations have been determined and validated, classification techniques can be used to determine the composition of new datasets by classifying cells into discrete types. Dominant classification methods include XGBoost [81], Random Forest (RF) [82], Deep Learning (DL) [83], and Gradient Boosting Machine (GBM) [84]. Another important down-stream analysis is pseudo-time inference. Cellular processes, such as cell cycle, proliferation, differentiation, and activation [86, 87], can be modeled computationally using trajectory inference methods. These methods aim at ordering the cells along developmental trajectories. Among a number of trajectory inference tools, Monocle [88], TSCAN [89], Slingshot [64], and SCANPY [77] are considered state-of-the-art and are widely used for pseudo-temporal ordering.

As the volume of scRNA-seq data increases exponentially each year [93], the above-mentioned methods methods have become primary investigation tools in many research fields, including cancer [90], immunology [91], or virology [92]. However, the ever-increasing number of cells, technical noise, and high dropout rate pose significant computational challenges in scRNA-seq analysis [73, 110, 111]. These challenges affect both analysis accuracy and scalability, and greatly hinder our capability to extract the wealth of information available in single-cell data.

4.2 Methodology

We develop a new analysis framework, called single-cell Decomposition using Hierarchical Autoencoder (scDHA), that can efficiently detach noise from informative biological signals. Figure 4.1 depicts the overall pipeline of scDHA. The first module is

a non-negative kernel autoencoder that provides a non-negative, part-based representation of the data. Based on the weight distribution of the encoder, scDHA removes genes or components that have insignificant contributions to the representation. The second module is a Stacked Bayesian Self-learning Network that is built upon the Variational Autoencoder [112] to project the data onto a low dimensional space (see Methods section). Using this informative and compact representation, many analyses can be performed with high accuracy and tractable time complexity (mostly linear or lower complexity). In one joint framework, the scDHA software package conducts cell segregation through unsupervised learning, dimension reduction and visualization, cell classification, and time-trajectory inference. We will show that scDHA outperforms state-of-the-art methods in all four sub-fields: cell segregation through unsupervised learning, transcriptome landscape visualization, cell classification, and pseudo-time inference. The details of each step are described below.

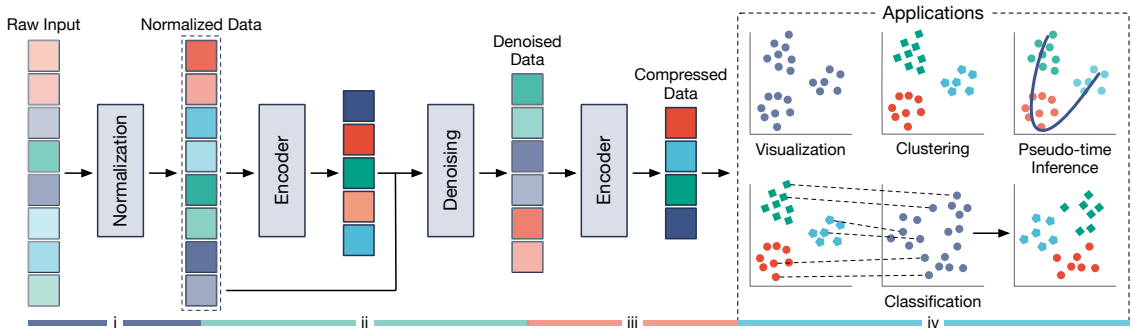

Figure 4.1: Overview of scDHA architecture. scDHA data processing and analyzing pipeline includes four steps: (i) Data input scaling, (ii) Data filtering using non-negative kernel autoencoder, (iii) Data compression using Stacked Bayesian Autoencoder, and (iv) Downstream applications.

4.2.1 Data filtering using non-negative kernel autoencoder

The method requires an expression matrix M as input, in which rows represents cells and columns represent genes or transcripts. Given the input M, scDHA first

automatically performs a log transformation (base 2) to rescale the data if the range of M is higher than 100. The goal is to prevent the domination of genes or features with high expression. To reduce the technical variability and heterogeneous calibration from sequencing technologies, the expression data is additionally rescaled to a range of 0 to 1 for each cell as follow:

$$X_{ij} = \frac{M_{ij} - min(M_{i.})}{max(M_{i.}) - min(M_{i.})} \tag{4.1}$$

where M is the input matrix and X is the log-based normalized matrix. This min-max scaling step is to reduce standard deviation and to suppress the effect of outliers, which is frequently used in deep learning models [113, 114]

After normalization, the data is then passed through a one-layer autoencoder to filter out insignificant genes/features. In short, autoencoder consists of two components: encoder and decoder. The formulation of autoencoder can be written as follows:

$$\begin{aligned} e &= f_E(x) \\ \bar{x} &= f_D(e) \end{aligned} \tag{4.2}$$

where $x \in R_+^n$ is the input of the model (x is simply a row/sample, i.e., $x = X_{i.}$), f_E and f_D represent the transformation by encoder and decoder layers, $\bar{x}$ is the reconstruction of x. The encoder and decoder transformations can be represented as $f_E(x) = xW_E + b_E$ and $f_D(e) = eW_D + b_D$, where W-s are the weight matrices and b-s are the bias vectors. Encoder aims at representing the data in a much lower dimensional space (compression) whereas decoder tries to reconstruct the original input from the compressed data. Optimizing this process can theoretically result in a compact representation of the original, high-dimensional data. The size of the bottleneck layer is set to 50 nodes (not user-provided parameter). Changing this number of nodes has no significant impact on the results of scDHA.

In our model, the weights of the encoder (W_E in $f_E(\cdot)$) are forced to be non-negative so that each latent variable is an additive combination of the original features. By doing so, the non-negative coefficients of the less important features will be shrunk toward zero. Based on the computed weights, the method only keeps genes or components with high weight variances. In principle, the set of these genes can be considered a "*sufficient and necessary*" set to represent the original data. These genes are *necessary* because removing them would greatly damage the reversibility of decoder, i.e., decoder cannot accurately reconstruct the original data. At the same time, they are *sufficient* because encoder automatically shrinks the weights of genes or gene groups that have similar but lesser impacts in the compression procedure. By default, scDHA selects $5,000$ genes but users can choose a different number based on the weight distribution.

4.2.2 Data compression using Stacked Bayesian Autoencoder

After the gene filtering step using non-negative kernel autoencoder, we obtain a data matrix in which each gene is considered critical to preserve cell heterogeneity. However, although the step has greatly reduced the number of features, the number of genes is still in the scale of hundreds or thousands. Therefore, it is necessary to perform dimension reduction before conducting any analysis or visualization. For this purpose, we developed a modified version of VAE (theorized by Kingma *et al.* [112]). We name it Stacked Bayesian Autoencoder (Figure 4.2) since the model is designed with multiple latent spaces, instead of only one latent space used in the original VAE or any other autoencoder model.

VAE has the same basic structure as a standard autoencoder, which is a self-learning model consisting of two components: encoder and decoder. Given the input matrix (the filtered matrix obtained from Non-negative kernel autoencoder), VAE's

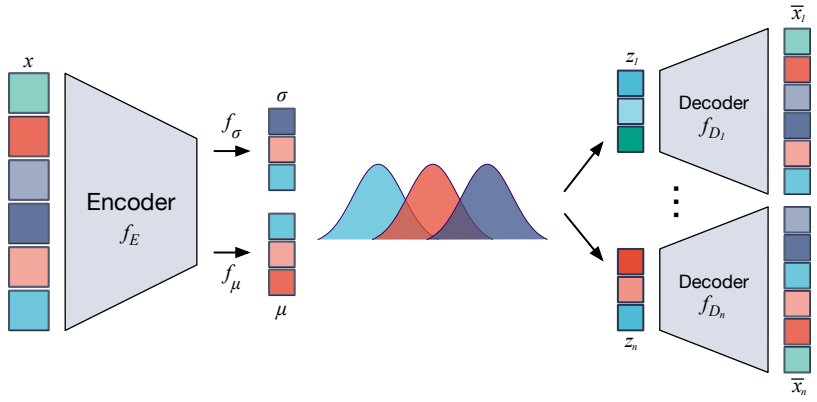

Figure 4.2: High-level representation of Stacked Bayesian Autoencoder. The encoder projects input data to multiple low-dimensional latent spaces (outputs of z_1 to z_n layers). The decoders infer original data from these latent data. Minimizing the difference between inferred data and original one leads to a high quality representation of the data at bottle neck layer (outputs of μ layer).

encoder constructs a low-dimensional representation of the input matrix while the decoder aims at inferring the original data. By minimizing the difference between the inferred and the input data, the middle bottleneck layer is considered as the "near lossless" projection of the input onto a latent space with a low number of dimensions ($m = 15$ by default). We keep the model size small to avoid over-fitting and force the neuron network to be as compressed as possible. Also, restricting the size of latent layer will converge cells from the same group into similar latent space manifold. At the same time, the size of the latent layer needs to be sufficient (15 dimensions) to keep the latent variables disentangled. Per our experience, varying m between 10 and 20 do not alter the analysis results.

Given an expression profile of a cell x, the formulation of this architecture can be

formulated as follows:

$$\begin{aligned} e &= f_E(x) \\ \mu &= f_\mu(e) \\ \sigma &= f_\sigma(e) \\ z &\sim N(\mu, \sigma^2) \\ \bar{x} &= f_D(z) \end{aligned} \tag{4.3}$$

where $x \in R_+^n$ is the input of the network, f_E and f_D represent the transformation by encoder and decoder layers. In addition to the standard autoencoder, two transformations f_μ and f_σ are added on the output e of encoder to generate the parameters μ and σ ($\mu, \sigma \in R^m$). The compress data z is now sampled from the distribution $N(\mu, \sigma^2)$. In contrast to the standard autoencoder, VAE uses z as the input of the decoder instead of e. By adding randomness in generating z, VAE prevents overfitting by avoiding mapping the original data to the compressed space without learning a generalized representation of data. The perturbation process was shown to be an effective method to increase data stability [44].

In our stacked model, to further diminish overfitting and increase the robustness, we generate multiple compressed spaces with multiple realizations of z. For that purpose, we use a re-parameterization trick to generate multiple realizations of z as follows: $z = \mu + \sigma * N(0, 1)$. This re-parameterization trick is introduced to ensure that the model can backpropagate [112].

To train our model, we use AdamW [115] as optimizer while adopting two stage training scheme [116]: (1) a *Warm-up* process which uses only reconstruction loss, and (2) the VAE stage, in which the Kullback-Leibler loss is also considered to ensure the normal distribution of latent variables z. The warm-up process prevents the model from ignoring reconstruction loss and only focuses on Kullback-Leibler loss. By doing this, we avoid the pitfall of making the model fail to learn generalized

representations of the data. This process also makes the model less sensitive to the weight initialization. For faster convergence and better accuracy, scaled exponential linear unit (SELU) [117] is used as the activation function.

After finishing the training stage, scDHA processes the input data through encoder to generate representative latent variables of original data. This compressed representation of the data will be used for single-cell applications: (1) cell segregation through unsupervised learning, (2) dimension reduction and visualization, (3) cell classification, and (4) pseudo-time trajectory inference.

4.2.3 Cell segregation via clustering

Predicting the number of cell types. The number of cell types is determined using two indices: (i) the ratio of *between sum of squares* over the *total sum of squares*, and (ii) the increase of the *within sum of squares* when number of clusters increases. The indices are formulated as follows:

$$Index\ 1 = \frac{SS_{between,j}}{SS_{total,j}} \tag{4.4}$$

$$Index\ 2 = \frac{SS_{within,j+1} - SS_{within,j}}{SS_{within,j}} \tag{4.5}$$

where j is number of clusters.

Larger *Index 1* means that members of one group are far from other groups, i.e., the clusters are well separated. *Index 2* is affected by the number of eigenvectors generated by spectral decomposition, which is also the number of clusters. We assume that the addition of an eigenvector that leads to the highest spike in the *within sum of squares* (which is undesirable) would be the correct number of clusters. These indices are calculated by performing k-nearest neighbor spectral clustering on a subset of samples over a range of cluster number. Mean of the predictions from these two

indices is set to be the final number of clusters.

Basic clustering algorithm. In order to improve the accuracy when clustering non-spherical data while ensuring the fast running time, we apply a k-nearest neighbor adaption of spectral clustering (k-nn SC) as the clustering method embedded in our package. Instead of using Euclidean distance to determine the similarity between two samples, Pearson correlation is used to improve the stability of cluster assignment. The difference between k-nn SC and normal SC is that the constructed affinity matrix of data points is sparse. For each data point, the distance is calculated for only its k nearest neighbors while the distance to the rest is left at zero. The clustering process of k-nn SC consists of 4 steps: (i) constructing affinity matrix A for all data points to use as input graph, (ii) generating a symmetric and normalized Laplacian matrix $L^{\text{sym}} = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ where D is the degree matrix of the graph, A is the constructed affinity matrix and I is the identity matrix, (iii) calculating eigenvalues for Laplacian matrix and select those with smallest values, generating eigenvectors corresponding to selected eigenvalues, (iv) performing final clustering using k-means on the obtained eigenvectors.

Consensus clustering. We use the basic clustering algorithm described above to cluster the compressed data. To achieve higher accuracy and to avoid local minima, an ensemble of data projection models is used. We first repeat the data projection and clustering process multiple times. We then combine the clustering results using the Weighted-based meta-clustering (wMetaC) implemented in SHARP [50]. wMetaC is conducted through 5 steps: (i) calculating cell-cell weighted similarity matrix W, $w_{i,j} = s_{i,j}(1 - s_{i,j})$ where $s_{i,j}$ is the chance that cell i and j are in the same cluster, (ii) calculating cell weight, which is the sum of all cell-cell weights related to this cell, (iii) generating cluster-cluster similarity matrix $|C|x|C|$, where C is the union of all the clusters obtained in each replicate, (iv) performing hierarchical clustering on

cluster-cluster similarity matrix, and (v) determining final results by voting scheme.

Voting procedure. For large datasets, we also provide an additional option in our package to reduce the time complexity without compromising the performance. Instead of clustering the whole dataset, which requires a large amount of memory and heavy computation, we can perform the clustering on a subset of the data points and then apply a vote-counting procedure to assign the rest of the data to each cluster. The voting process is based on the k-nearest neighbor classification. This approach still ensures the high clustering quality without compromising the speed of method.

4.2.4 Dimension reduction and visualization

Given the compressed data (10 to 15 dimensions), we compute the distance matrix for the cells and then perform log and z transformations as follows:

$$D_{ij} = \frac{\log(D_{ij}) - \mu_{\log(D_{i.})}}{\sigma_{\log(D_{i.})}} \tag{4.6}$$

where D is a distance matrix. The rationale of this transformation is to make the distribution of distances from one point to its neighbors more uniform. Next, we calculate the probabilities p_{ij} that are proportional to the similarity between sample i and j as follows:

$$p_{j|i} = \frac{\exp(D_{ij})}{\sum_{k \neq i} \exp(D_{ik})} \tag{4.7}$$

At the same time, using the compressed data, we build a neural network to project the data to 2-dimensional space. Using two formulas described above, we re-calculate the probabilities q_{ij} that are proportional to the similarity between sample i and j in the 2-dimensional space. Our goal is to learn a 2-dimensional projection of the data that retains the probabilities p as well as possible. We achieve this by minimizing the distance between Q and P. Here, we use the Kullback-Leibler divergence to represent

the distance between the two probability distributions, which can be formulated as:

$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{4.8}$$

By minimizing Kullback-Leibler divergence, we obtain the optimal representation of the data in the 2-dimensional space. The algorithm can be generalized to three or higher number of dimensions.

4.2.5 Cell classification

The problem can be described as follows. We are given two datasets of the same tissue: the training dataset and the testing dataset. For the training dataset, we have the cell labels. The goal is to determine the cell labels of the testing dataset.

Our classification procedure consists of the following steps: (i) concatenate the two matrices into a single matrix, in which the rows consist of all cells from the two datasets and columns are the common genes; (ii) normalize and compress the merged data using the hierarchical autoencoder described above; (iii) compute the similarity matrix for the cells using Pearson correlation; and finally (iv) determine the label of cells from testing data using k-nearest neighbor algorithm (k-nn).

The rationale for concatenating the two datasets is to exploit the robust denoising and dimension reduction procedure offered by the hierarchical autoencoder. Since we normalize the data per each cell, different scaling of the two datasets (training or testing) would not pose as a problem. At the same time, the hierarchical autoencoder efficiently diminishes batch effect and noise, moving cells of the same type closer to one another. We demonstrated that even with an unsophisticated classification technique as k-nn, scDHA is proven to be better than current state-of-the-art methods, including XGBoost, Random Forest, Deep Learning, and Gradient Boosted Machine.

4.2.6 Pseudo-time trajectory inference

We implement a pseudo-time inference method that allows users to infer non-branching trajectory that is correlated with the developmental stages of cells. This method requires a starting point as part of the input. We note that users can easily apply any other methods on the compressed data provided by scDHA (see Saelens *et al.* [85] for a comprehensive list of pseudo-time inference methods). Given the compressed data, our method computes the similarity distance for the cells using Pearson correlation. Using this similarity matrix as the affinity matrix, we construct a graph in which nodes represent cells and edges represent the distance between the cells. In order to construct the pseudo-time trajectory, we apply the minimum spanning tree (MST) algorithm on the graph to find the shortest path that goes through all cells. From the MST, pseudo time is determined by distance from one point to the designated starting point.

4.3 Validation and Analysis Results

To validate our method, we use real scRNA-seq data that are generated from human or mouse tissues using different protocols. By validating the methods with data from different tissue origins and protocols, we can ensure the stability of the proposed methods. Table 4.1 shows the details of 34 single-cell datasets that will be used in our validation. The datasets Montoro, Sanderson, Slyper, Zilionis, Karagiannis, Orozco, and Kozareva were downloaded from Broad Institute Single Cell Portal (`https://singlecell.broadinstitute.org/single_cell`). The datasets Puram, Hrvatin, and Darrah were downloaded from Gene Expression Omnibus. Tabula Muris was downloaded from Figshare. The remaining 23 datasets were downloaded from Hemberg Group's website (`https://hemberg-lab.github.io/scRNA.seq.dat`

asets). The only processing step we did was to perform log transformation (base 2) to rescale the data if the range of the data is larger than 100. These datasets include the ground truth (true cell type labels) for the each sample. This allows accurate validation of downstream analysis performance. We validate the quality of data using downstream analyses including clustering, visualization, classification, and time-trajectory inference.

For clustering analysis, we compare our clustering result with other state-of-the-arts including SC3 [74], SEURAT [68], SINCERA [75], CIDR [76], and SCANPY [77]. For visualization, we compare the transcriptome landscape generated by our methods with dominant methods including t-SNE [80], UMAP [62], SCANPY [77], and the classical principal component analysis (PCA). For classification, we compare the accuracy of our classifier with four methods that are dominant in machine learning: XGBoost [81], Random Forest (RF) [82], Deep Learning (DL) [83], and Gradient Boosting Machine (GBM) [84]. For time-trajectory inference, we compare our methods with state-of-the-art methods for time-trajectory inference: Monocle [88], TSCAN [89], Slingshot [64], and SCANPY [77].

4.3.1 Cell segregation

We assess the performance of scDHA in clustering using 34 scRNA-seq datasets with known cell types. The true class information of these datasets is only used *a posteriori* to assess the results. We compare scDHA with five methods that are widely used for single-cell clustering: SC3 [74], SEURAT [68], SINCERA [75], CIDR [76], and SCANPY [77]. Note that SCANPY is also an all-in-one pipeline that is able to perform three types of analysis: clustering, visualization and pseudo-time inference. We include k-means as the reference method in cluster analysis.

Since the true cell types are known in these datasets, we use adjusted Rand index

Table 4.1: Description of the 34 single-cell datasets used to assess the performance of computational methods. The first two columns describe the name and tissue while the next four columns show the number of cells, number of cell types, protocol, and accession ID.

Dataset	Tissue	Size	Class	Protocol	Accession ID	Reference
1. Yan	Human Embryo	90	6	Tang	GSE36552	Yan *et al.*, 2013 [118]
2. Goolam	Mouse Embryo	124	5	Smart-Seq2	E-MTAB-3321	Goolam *et al.*, 2016 [119]
3. Deng	Mouse Embryo	268	6	Smart-Seq2	GSE45719	Deng *et al.*, 2014 [120]
4. Pollen	Human Tissues	301	11	SMARTer	SRP041736	Pollen *et al.*, 2014 [121]
5. Patel	Human Tissues	430	5	Smart-Seq	GSE57872	Patel *et al.*, 2014 [109]
6. Wang	Human Pancreas	457	7	SMARTer	GSE83139	Wang *et al.*, 2016 [122]
7. Darmanis	Human Brain	466	9	SMARTer	GSE67835	Darmanis *et al.*, 2015 [123]
8. Camp (Brain)	Human Brain	553	5	SMARTer	GSE75140	Camp *et al.*, 2015 [124]
9. Usoskin	Mouse Brain	622	4	STRT-Seq	GSE59739	Usoskin *et al.*, 2015 [125]
10. Kolodziejczyk	Mouse Embryo Stem Cells	704	3	SMARTer	E-MTAB-2600	Kolodziejczyk *et al.*, 2015 [126]
11. Camp (Liver)	Human Liver	777	7	SMARTer	GSE81252	Camp *et al.*, 2017 [127]
12. Xin	Human Pancreas	1,600	8	SMARTer	GSE81608	Xin *et al.*, 2016 [128]
13. Baron (Mouse)	Mouse Pancreas	1,886	13	inDrop	GSE84133	Baron *et al.*, 2016 [129]
14. Muraro	Human Pancreas	2,126	10	CEL-Seq2	GSE85241	Muraro *et al.*, 2016 [130]
15. Segerstolpe	Human Pancreas	2,209	14	Smart-Seq2	E-MTAB-5061	Segerstolpe *et al.*, 2016 [131]
16. Klein	Mouse Embryo Stem Cells	2,717	4	inDrop	GSE65525	Klein *et al.*, 2015 [132]
17. Romanov	Mouse Brain	2,881	7	SMARTer	GSE74672	Romanov *et al.*, 2017 [133]
18. Zeisel	Mouse Brain	3,005	9	STRT-Seq	GSE60361	Zeisel *et al.*, 2015 [71]
19. Lake	Human Brain	3,042	16	Fluidigm C1	phs000833.v3.p1	Lake *et al.*, 2016 [134]
20. Puram	Human Tissues	5,902	10	Smart-Seq2	GSE103322	Puram *et al.*, 2017 [135]
21. Montoro	Human Pancreas	7,193	7	Smart-Seq2	GSE103354	Montoro *et al.*, 2018 [136]
22. Baron (Human)	Human Pancreas	8,569	14	inDrop	GSE84133	Baron *et al.*, 2016 [129]
23. Chen	Mouse Brain	12,089	46	Drop-seq	GSE87544	Chen *et al.*, 2017 [137]
24. Sanderson	Mouse Tissues	12,648	11	10X Genomics	SCP916	Sanderson *et al.*, 2020 [138]
25. Slyper	Human Blood	13,316	8	10X Genomics	SCP345	
26. Campbell	Mouse Brain	21,086	21	Drop-seq	GSE93374	Campbell *et al.*, 2017 [139]
27. Zilionis	Human Lung	34,558	9	inDrop	GSE127465	Zilionis *et al.*, 2019 [140]
28. Macosko	Mouse Retina	44,808	12	Drop-seq	GSE63473	Macosko *et al.*, 2015 [141]
29. Hrvatin	Mouse Visual Cortex	48,266	8	inDrop	GSE102827	Hrvatin *et al.*, 2018 [142]
30. Tabula Muris	Mouse Tissues	54,439	40	10X Genomics	GSE109774	Schaum *et al.*, 2018 [143]
31. Karagiannis	Human Blood	72,914	12	10X Genomics	GSE128879	Karagiannis *et al.*, 2020 [144]
32. Orozco	Human Eye	100,055	11	10X Genomics	GSE135133	Orozco *et al.*, 2020 [145]
33. Darrah	Human Blood	162,490	14	Drop-seq	GSE139598	Darrah *et al.*, 2020 [146]
34. Kozareva	Mouse Cerebellum	611,034	18	10X Genomics	SCP795	Kozareva *et al.*, 2020 [147]

(ARI) [148] to assess the performance of the six clustering methods. Figure 4.3 shows the ARI values obtained for each dataset, as well as the average ARIs and their variances. scDHA outperforms all other methods by not only having the highest average ARI, but also being the most consistent method. The average ARI of scDHA across all 34 datasets is 0.81 with very low variability. The second best method, CIDR, has an average ARI of only 0.5. The one-sided Wilcoxon test also indicates that the ARI values of scDHA are significantly higher than the rest with a p-value of 2.2×10^{-16}. To perform a more comprehensive analysis, we calculate the normalized mutual information (NMI) and Jaccard index (JI) for each method. Tables 4.2–4.4 show the detailed results of all methods on 34 single-cell datasets measured by the three metrics.

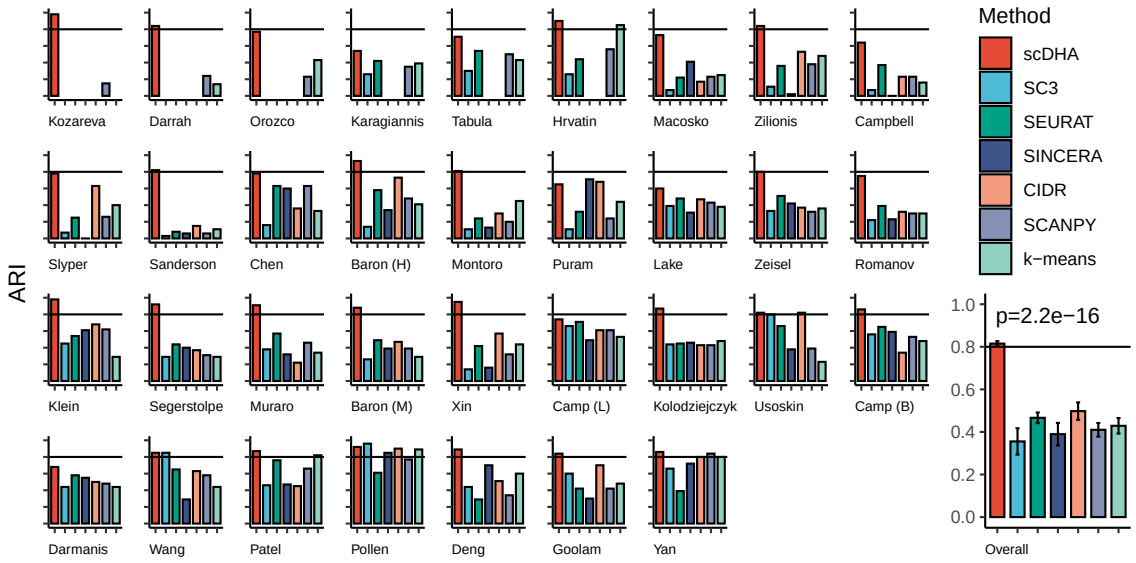

Figure 4.3: Clustering performance of scDHA, SC3, SEURAT, SINCERA, CIDR, SCANPY, and k-means measured by adjusted Rand index (ARI) on 34 scRNA-seq datasets. The first 34 panels show the ARI values obtained for individual datasets while the last panel shows the average ARIs and their variance (vertical segments). scDHA significantly outperforms other clustering methods by having the highest ARI values ($p = 2.2 \times 10^{-16}$ using one-sided Wilcoxon test). (b) Running time of the clustering methods, each using 10 cores. scDHA is the fastest among the six methods.

In 34 datasets analyzed, there are 19 plate-based datasets (Fluidigm C1, Tang,

Table 4.2: Performance of scDHA, SC3, SEURAT, SINCERA, CIDR, SCANPY, and k-means on 34 single-cell datasets measured by adjusted Rand index (ARI). Cells with NA values indicate that the method was not able to analyze the dataset (crashed or out-of-memory). Cells highlighted in green have the highest ARI values. The average ARI of scDHA is 0.81, which is much higher than the rest (CIDR is the second best with an average ARI of 0.5). In addition, scDHA has the highest ARI values in all but two datasets (Pollen and Puram).

Dataset	Size	Class	scDHA	SC3	SEURAT	SINCERA	CIDR	SCANPY	k-means
1. Yan	90	6	0.86	0.66	0.39	0.72	0.80	0.84	0.80
2. Goolam	124	5	0.84	0.60	0.42	0.30	0.70	0.42	0.48
3. Deng	268	6	0.89	0.44	0.29	0.70	0.51	0.34	0.60
4. Pollen	301	11	0.92	0.96	0.61	0.85	0.90	0.77	0.89
5. Patel	430	5	0.87	0.46	0.76	0.47	0.45	0.66	0.82
6. Wang	457	7	0.85	0.85	0.65	0.29	0.63	0.58	0.44
7. Darmanis	466	9	0.68	0.44	0.58	0.55	0.50	0.48	0.44
8. Camp (B)	553	5	0.86	0.56	0.65	0.59	0.34	0.53	0.48
9. Usoskin	622	4	0.82	0.80	0.66	0.38	0.82	0.39	0.23
10. Kolodziejczyk	704	3	0.87	0.44	0.45	0.46	0.43	0.43	0.48
11. Camp (L)	777	7	0.74	0.66	0.71	0.49	0.61	0.61	0.53
12. Xin	1,600	8	0.95	0.14	0.42	0.16	0.57	0.32	0.44
13. Baron (M)	1,886	13	0.88	0.26	0.49	0.39	0.47	0.39	0.29
14. Muraro	2,126	10	0.91	0.38	0.57	0.32	0.22	0.46	0.34
15. Segerstolpe	2,209	14	0.92	0.29	0.44	0.40	0.37	0.31	0.29
16. Klein	2,717	4	0.98	0.45	0.54	0.61	0.68	0.62	0.29
17. Romanov	2,881	7	0.75	0.22	0.39	0.23	0.32	0.30	0.30
18. Zeisel	3,005	9	0.80	0.33	0.51	0.42	0.37	0.32	0.36
19. Lake	3,042	16	0.60	0.39	0.48	0.31	0.47	0.43	0.38
20. Puram	5,902	10	0.65	0.11	0.32	0.71	0.68	0.24	0.44
21. Montoro	7,193	7	0.81	0.11	0.24	0.13	0.30	0.20	0.45
22. Baron (H)	8,569	14	0.93	0.14	0.58	0.34	0.73	0.48	0.41
23. Chen	12,089	46	0.78	0.16	0.63	0.60	0.36	0.63	0.33
24. Sanderson	12,648	11	0.82	0.03	0.08	0.06	0.15	0.06	0.11
25. Slyper	13,316	8	0.78	0.07	0.25	0.00	0.63	0.26	0.40
26. Campbell	21,086	21	0.64	0.07	0.37	0.00	0.23	0.23	0.16
27. Zilionis	34,558	9	0.84	0.11	0.36	0.02	0.53	0.38	0.48
28. Macosko	44,808	12	0.73	0.07	0.22	0.41	0.17	0.23	0.25
29. Hrvatin	48,266	8	0.90	0.26	0.44	NA	NA	0.56	0.85
30. Tabula Muris	54,439	40	0.71	0.30	0.54	NA	NA	0.50	0.43
31. Karagiannis	72,914	12	0.54	0.26	0.42	NA	NA	0.35	0.39
32. Orozco	100,055	11	0.77	NA	NA	NA	NA	0.23	0.43
33. Darrah	162,490	14	0.84	NA	NA	NA	NA	0.24	0.14
34. Kozareva	611,034	18	0.98	NA	NA	NA	NA	0.15	NA
Mean ARI			0.81	0.36	0.47	0.39	0.50	0.41	0.43

Table 4.3: Performance of scDHA, SC3, SEURAT, SINCERA, CIDR, SCANPY, and k-means on 34 single-cell datasets measured by normalized mutual information (NMI). Cells with NA values indicate that the method was not able to analyze the dataset (crashed or out-of-memory). Cells highlighted in green have the highest NMI values. scDHA outperforms other methods by having the highest average NMI value. In addition, scDHA has the highest NMI values in 31 out of 34 datasets.

Dataset	Size	Class	scDHA	SC3	SEURAT	SINCERA	CIDR	SCANPY	k-means
1. Yan	90	6	0.89	0.80	0.55	0.82	0.84	0.87	0.86
2. Goolam	124	5	0.82	0.80	0.61	0.61	0.78	0.71	0.63
3. Deng	268	6	0.89	0.73	0.53	0.73	0.74	0.70	0.78
4. Pollen	301	11	0.96	0.95	0.80	0.93	0.94	0.91	0.94
5. Patel	430	5	0.84	0.67	0.76	0.67	0.57	0.72	0.83
6. Wang	457	7	0.83	0.81	0.71	0.43	0.71	0.71	0.57
7. Darmanis	466	9	0.75	0.67	0.64	0.66	0.64	0.69	0.62
8. Camp (B)	553	5	0.82	0.68	0.70	0.62	0.49	0.69	0.55
9. Usoskin	622	4	0.81	0.79	0.74	0.54	0.80	0.65	0.31
10. Kolodziejczyk	704	3	0.90	0.68	0.68	0.54	0.57	0.67	0.51
11. Camp (L)	777	7	0.85	0.81	0.85	0.69	0.79	0.82	0.72
12. Xin	1,600	8	0.87	0.39	0.60	0.42	0.55	0.61	0.60
13. Baron (M)	1,886	13	0.85	0.65	0.75	0.61	0.51	0.74	0.59
14. Muraro	2,126	10	0.88	0.69	0.77	0.51	0.43	0.74	0.53
15. Segerstolpe	2,209	14	0.90	0.65	0.75	0.62	0.45	0.69	0.53
16. Klein	2,717	4	0.97	0.69	0.71	0.67	0.66	0.76	0.40
17. Romanov	2,881	7	0.69	0.43	0.60	0.31	0.34	0.58	0.35
18. Zeisel	3,005	9	0.78	0.62	0.67	0.47	0.47	0.63	0.55
19. Lake	3,042	16	0.67	0.68	0.73	0.47	0.54	0.73	0.62
20. Puram	5,902	10	0.79	0.45	0.66	0.68	0.63	0.62	0.63
21. Montoro	7,193	7	0.74	0.30	0.50	0.24	0.46	0.47	0.56
22. Baron (H)	8,569	14	0.88	0.50	0.80	0.46	0.72	0.77	0.63
23. Chen	12,089	46	0.77	0.53	0.79	0.53	0.42	0.77	0.63
24. Sanderson	12,648	11	0.71	0.21	0.43	0.29	0.12	0.40	0.40
25. Slyper	13,316	8	0.73	0.36	0.60	0.16	0.70	0.59	0.62
26. Campbell	21,086	21	0.68	0.49	0.74	0.15	0.38	0.69	0.48
27. Zilionis	34,558	9	0.83	0.41	0.70	0.08	0.58	0.66	0.62
28. Macosko	44,808	12	0.59	0.31	0.56	0.19	0.33	0.56	0.40
29. Hrvatin	48,266	8	0.92	0.59	0.74	NA	NA	0.77	0.88
30. Tabula Muris	54,439	40	0.80	0.65	0.77	NA	NA	0.77	0.68
31. Karagiannis	72,914	12	0.66	0.49	0.73	NA	NA	0.66	0.65
32. Orozco	100,055	11	0.76	NA	NA	NA	NA	0.60	0.70
33. Darrah	162,490	14	0.78	NA	NA	NA	NA	0.61	0.34
34. Kozareva	611,034	18	0.92	NA	NA	NA	NA	0.58	NA
Mean NMI			0.81	0.60	0.68	0.50	0.58	0.68	0.60

Table 4.4: Performance of scDHA, SC3, SEURAT, SINCERA, CIDR, SCANPY, and k-means on 34 single-cell datasets measured by Jaccard Index (JI). Cells with NA values indicate that the method was not able to analyze the dataset (crashed or out-of-memory). Cells highlighted in green have the highest JI values. scDHA outperforms other methods by having the highest average JI value. scDHA also has the highest JI values in 31 out of 34 datasets.

Dataset	Size	Class	scDHA	SC3	SEURAT	SINCERA	CIDR	SCANPY	k-means
1. Yan	90	6	0.80	0.57	0.38	0.64	0.73	0.77	0.73
2. Goolam	124	5	0.82	0.54	0.46	0.28	0.65	0.37	0.45
3. Deng	268	6	0.86	0.40	0.33	0.65	0.46	0.28	0.55
4. Pollen	301	11	0.87	0.93	0.50	0.76	0.83	0.66	0.82
5. Patel	430	5	0.81	0.35	0.67	0.36	0.38	0.55	0.75
6. Wang	457	7	0.81	0.81	0.58	0.31	0.56	0.50	0.39
7. Darmanis	466	9	0.59	0.34	0.48	0.46	0.42	0.37	0.35
8. Camp (B)	553	5	0.82	0.48	0.57	0.55	0.30	0.44	0.44
9. Usoskin	622	4	0.76	0.74	0.58	0.35	0.78	0.31	0.29
10. Kolodziejczyk	704	3	0.83	0.37	0.38	0.44	0.40	0.37	0.50
11. Camp (L)	777	7	0.64	0.54	0.59	0.40	0.49	0.48	0.44
12. Xin	1,600	8	0.94	0.13	0.39	0.15	0.58	0.29	0.41
13. Baron (M)	1,886	13	0.85	0.20	0.41	0.34	0.42	0.32	0.25
14. Muraro	2,126	10	0.87	0.29	0.46	0.32	0.23	0.36	0.30
15. Segerstolpe	2,209	14	0.88	0.21	0.34	0.35	0.35	0.22	0.24
16. Klein	2,717	4	0.97	0.36	0.46	0.58	0.61	0.54	0.33
17. Romanov	2,881	7	0.69	0.17	0.31	0.29	0.31	0.23	0.28
18. Zeisel	3,005	9	0.73	0.24	0.41	0.41	0.37	0.23	0.30
19. Lake	3,042	16	0.52	0.28	0.37	0.27	0.39	0.32	0.29
20. Puram	5,902	10	0.58	0.08	0.25	0.66	0.65	0.18	0.36
21. Montoro	7,193	7	0.80	0.11	0.23	0.13	0.29	0.19	0.43
22. Baron (H)	8,569	14	0.89	0.10	0.46	0.29	0.65	0.37	0.32
23. Chen	12,089	46	0.68	0.11	0.51	0.49	0.29	0.50	0.23
24. Sanderson	12,648	11	0.89	0.07	0.13	0.11	0.50	0.10	0.18
25. Slyper	13,316	8	0.77	0.07	0.23	0.02	0.62	0.24	0.39
26. Campbell	21,086	21	0.62	0.06	0.30	0.17	0.35	0.17	0.16
27. Zilionis	34,558	9	0.79	0.08	0.27	0.04	0.50	0.29	0.41
28. Macosko	44,808	12	0.76	0.08	0.22	0.50	0.24	0.22	0.28
29. Hrvatin	48,266	8	0.85	0.19	0.34	NA	NA	0.44	0.79
30. Tabula Muris	54,439	40	0.59	0.20	0.40	NA	NA	0.36	0.31
31. Karagiannis	72,914	12	0.51	0.21	0.33	NA	NA	0.29	0.33
32. Orozco	100,055	11	0.75	NA	NA	NA	NA	0.20	0.40
33. Darrah	162,490	14	0.79	NA	NA	NA	NA	0.18	0.19
34. Kozareva	611,034	18	0.99	NA	NA	NA	NA	0.18	NA
Mean Jaccard Index			0.77	0.30	0.40	0.37	0.48	0.34	0.39

SMARTer, Smart-Seq1/2, CEL-seq2, STRT-Seq) and 15 flow-cell-based datasets (inDrop, Drop-seq, 10X Genomics). There are four platforms that have more than five datasets per platform: Smart-Seq1/2, SMARTer, inDrop, and 10X Genomics. We compare scDHA with other methods for the six protocol groups: plate-based (19 datasets), flow-cell-based (15 datasets), Smart-Seq1/2 (six datasets), SMARTer (eight datasets), inDrop (five datasets), and 10X Genomics (six datasets). Figure 4.4 shows the performance of the clustering methods across the 6 platform groups. scDHA is the only method that performs consistently well across all six platform groups. The average ARI values of scDHA are close to 0.8 in all 6 groups. In contrast, the ARI values of other methods greatly differ across the platform groups. The average ARI of all methods drop when analyzing 10X Genomics data. This is partially due to the high dropout rate of 10X Genomics (the average dropout rates of Smart-Seq1/2, SMARTer, inDrop, and 10X Genomics datasets are 72.47, 76.61, 87.55, 91.50, respectively).

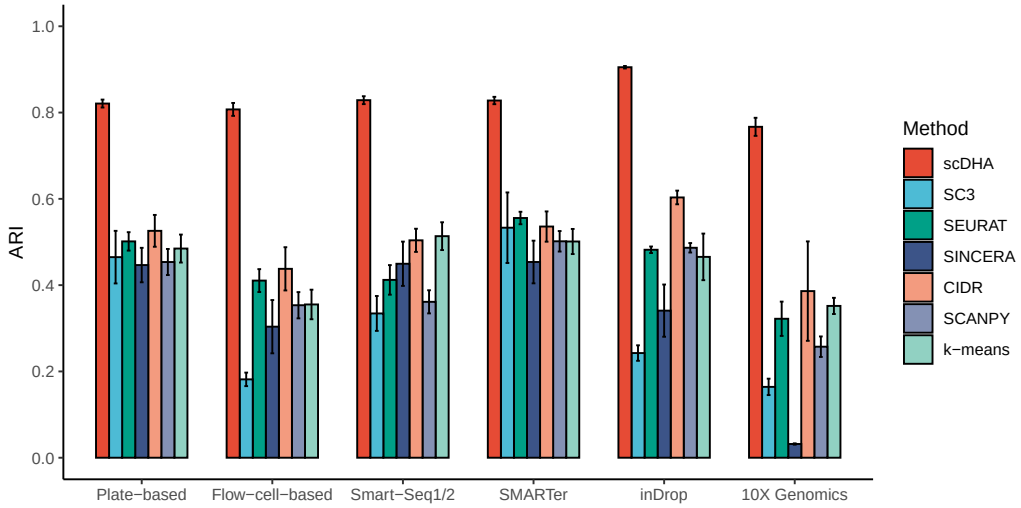

Figure 4.4: Clustering performance of scDHA, SC3, SEURAT, SINCERA, CIDR, SCANPY, and k-means across six data platforms. Data are presented as mean values +/- variance.

Figure 4.5 and Table 4.5 shows the running time of scDHA and other six clustering

methods on 34 scRNA-seq datasets. scDHA and SCANPY are the fastest among the seven methods. For the Macosko dataset with 44 thousand cells, scDHA finishes the analysis in less than five minutes. On the contrary, it takes CIDR more than two days (3,312 minutes) to finish the analysis of this dataset. In summary, scDHA outperforms other clustering methods in terms of both accuracy and scalability.

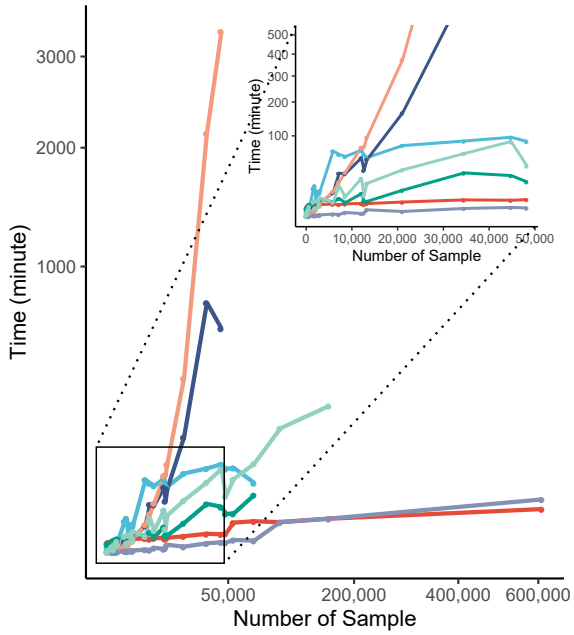

Figure 4.5: Running time of the scDHA, SC3, SEURAT, SINCERA, CIDR, SCANPY, and k-means on 34 scRNA-seq datasets. scDHA and SCANPY are the fastest among the seven methods.

Finally, we evaluate the consistency of scDHA with changing parameters. In the default setting of the denoising autoencoder, the bottleneck layer is set to a fixed size of 50 nodes. We test the model with different numbers of nodes and found that varying this number does not have a significant impact on the performance of the software. As shown in Figure 4.6, the average ARI value of the clustering results is consistently at 0.8 when we vary the number of nodes from 30 to 70.

Based on the computed weights of denoising module, we choose 5,000 genes with the highest weight variances (also the default setting). Figure 4.7a shows the nor-

Table 4.5: Running time of scDHA, SC3, SEURAT, SINCERA, CIDR, SCANPY, and k-means on 34 single-cell datasets. Overall, scDHA is the fastest and was able to analyze 611,034 cells within 24 minutes.

Dataset	**Size**	**scDHA**	**SC3**	**SEURAT**	**SINCERA**	**CIDR**	**SCANPY**	**k-means**
Yan	90	1.24	0.49	1.08	0.03	0.03	0.08	0.03
Goolam	124	1.52	0.46	0.92	0.05	0.04	0.03	0.13
Deng	268	1.51	0.50	0.94	0.05	0.05	0.03	0.37
Pollen	301	1.67	0.75	1.68	0.06	0.06	0.03	0.52
Patel	430	1.24	1.09	1.33	0.02	0.03	0.01	0.19
Wang	457	1.56	0.91	2.18	0.09	0.08	0.03	0.86
Darmanis	466	1.67	0.86	1.32	0.09	0.09	0.04	0.73
Camp (B)	553	1.57	1.10	2.04	0.11	0.08	0.03	0.80
Usoskin	622	1.67	1.44	1.97	0.17	0.17	0.03	1.09
Kolodziejczyk	704	1.89	1.71	2.59	0.29	0.23	0.05	1.79
Camp (L)	777	1.91	1.94	1.90	0.17	0.17	0.03	0.88
Xin	1,600	2.42	12.69	3.43	1.43	0.65	0.08	3.58
Baron (M)	1,886	2.33	15.01	1.43	0.71	0.54	0.04	1.20
Muraro	2,126	2.53	4.27	1.44	1.20	0.77	0.06	1.44
Segerstolpe	2,209	2.54	4.62	3.02	1.77	1.15	0.07	4.23
Klein	2,717	2.54	10.34	4.56	2.26	1.85	0.10	4.28
Romanov	2,881	2.56	8.78	3.08	2.57	2.09	0.07	3.05
Zeisel	3,005	2.50	9.00	3.04	2.51	1.96	0.08	1.90
Lake	3,042	2.53	10.44	4.94	3.11	2.85	0.10	5.78
Puram	5,902	2.72	66.35	3.69	9.39	10.16	0.18	3.62
Montoro	7,193	2.54	59.85	5.42	29.26	18.99	0.14	15.01
Baron (H)	8,569	2.81	55.79	3.36	28.93	30.73	0.37	6.49
Chen	12,089	3.00	67.84	8.51	53.33	73.57	0.24	22.98
Sanderson	12,648	2.57	59.44	3.96	33.39	74.31	0.31	7.20
Slyper	13,316	2.92	53.89	3.91	50.44	96.38	0.90	17.50
Campbell	21,086	3.60	77.56	11.19	164.04	372.83	0.56	34.05
Zilionis	34,558	4.68	87.73	29.85	764.05	2146.26	1.24	61.36
Macosko	44,808	4.49	96.58	26.40	614.65	3312.65	1.52	86.58
Hrvatin	48,266	4.81	86.76	19.11	NA	NA	1.39	40.00
Tabula Muris	54,439	11.52	89.16	19.23	NA	NA	2.21	66.90
Karagiannis	72,914	12.58	60.29	41.47	NA	NA	1.97	97.86
Orozco	100,055	11.80	NA	NA	NA	NA	12.06	189.99
Darrah	162,490	14.63	NA	NA	NA	NA	14.70	264.57
Kozareva	611,034	23.90	NA	NA	NA	NA	35.45	NA

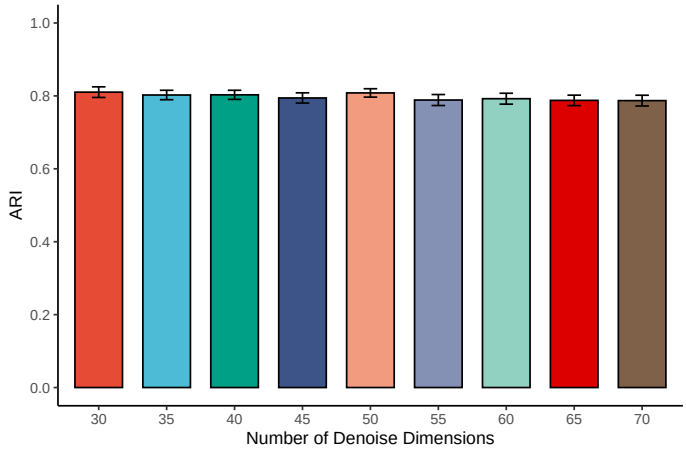

Figure 4.6: Clustering performance of scDHA on 34 single-cell datasets with varying size of bottleneck layer in the first module. Data are presented as mean values +/- variance.

malized weight variances in which each line represents a dataset. The figure shows that most lines are flattened at 5,000 genes. Another important note is that changing this threshold does not have a significant impact on the overall performance of scDHA. Figure 4.7b shows the clustering performance of scDHA with varying number of genes. The average ARI is consistently close to 0.8 when we change the number of genes from 3,000 to 10,000.

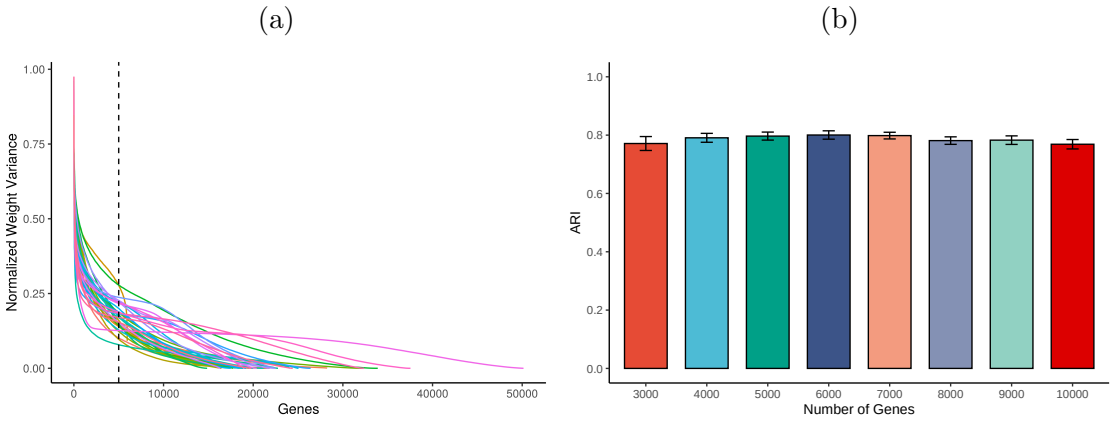

Figure 4.7: Effect of gene filtering cutoff on scDHA performance. (a) Normalized weight variance of genes. (b) Performance of scDHA on 34 single-cell datasets with varying number of selected genes. Data are presented as mean values +/- variance.

4.3.2 Dimension reduction and visualization

Here we demonstrate that scDHA is more efficient than t-SNE, UMAP, and SCANPY, as well as the classical principal component analysis (PCA) in visualizing single-cell data. We test the five techniques on the same 34 single-cell datasets described above. Again, cell type information is not given as input to any algorithm.

The top row of Figure 4.8a shows the color-coded representations of the Kolodziejczyk dataset, which consists of three mouse embryo stem cells: *2i*, *a2i*, and *lif*. The classical PCA simply rotates the orthogonal coordinates to place dissimilar data points far apart in the two-dimensional (2D) space. In contrast, t-SNE focuses on representing similar cells together in order to preserve the local structure. In this analysis, t-SNE splits each of the two classes *2i* and *a2i* into two smaller groups, and *lif* class into three groups. The transcriptome landscape represented by UMAP is similar to that of t-SNE, in which UMAP also splits cells of the same types into smaller groups. According to the authors of this dataset [126], embryonic stem cells were cultured in three different conditions: *lif* (serum media that has leukemia inhibitory factor), *2i* (basal media that has GSK3β and Mek1/2 inhibitor), and *a2i* (alternative *2i* that has GSK3β and Src inhibitor). The *lif* cells were measured in two batches and both t-SNE and UMAP split this cell type according to batches. Similarly, the *a2i* cells were measured by two batches and the cells were separated according to batches. The *2i* cells were measured by four batches (chip1 - 82 cells, chip2 - 59 cells, chip3 - 72 cells, and chip4 - 82 cells). Both t-SNE and UMAP split the cells into two groups: chip2, chip3 and chip4 were grouped together and were separated from chip1. SCANPY was able to mitigate batch effects in the *lif* cells but still split *2i* and *a2i* cells. In contrast, scDHA provides a clear representation of the data, in which cells of the same type are grouped together and cells of different types are well-separated.

The lower row of Figure 4.8a shows the visualization of the Sergerstolpe dataset

(human pancreas). The landscapes of SCANPY, UMAP and t-SNE are better than that of PCA. In these representations, the cell types are separable. However, the cells are overcrowded and many cells from different classes overlap. Also, the *alpha*, *beta* and *gamma* cells are split into smaller groups. According to the authors of this dataset [131], the data were collected from different donors, which is potentially the source of heterogeneity. For this dataset, scDHA again better represents the data by clearly showing the transcriptome landscape with separable cell types.

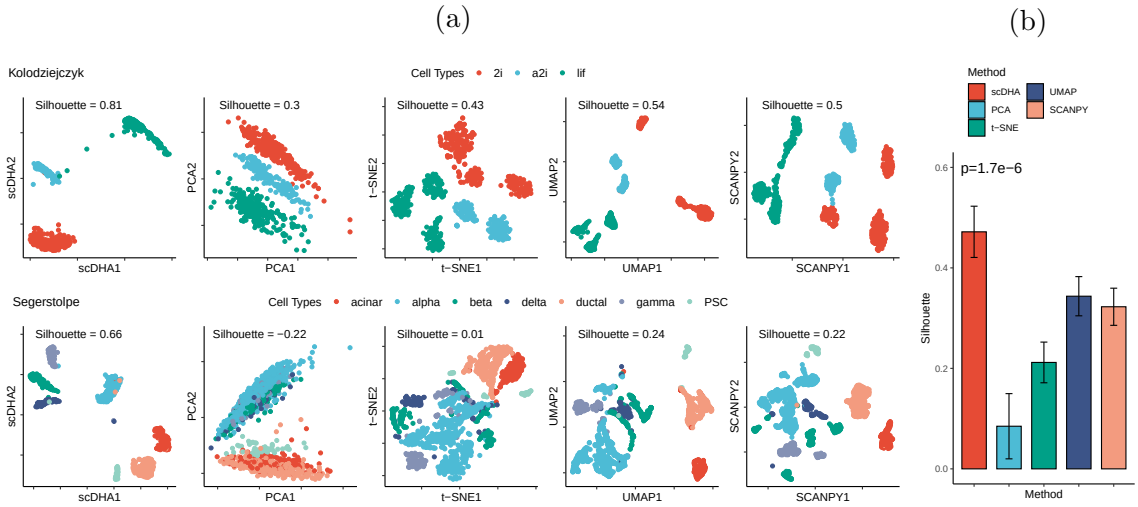

Figure 4.8: Transcriptome landscape visualization of Kolodziejczyk and Sergerstolpe datasets using scDHA, PCA, t-SNE, and UMAP. (a) Color-coded representation of the Kolodziejczyk and Segerstolpe datasets using scDHA, PCA, t-SNE, UMAP, and SCANPY (from left to right). For each representation, we report the silhouette index, which measures the cohesion among cells of the same type, as well as the separation between different cell types. (b) Average silhouette values (bar plot) and their variance (vertical lines). scDHA significantly outperforms other dimension reduction methods by having the highest silhouette values ($p = 1.7 \times 10^{-6}$ using one-sided Wilcoxon test).

To quantify the performance of each method, we calculate the silhouette index (SI) [149] of each representation using true cell labels. This metric measures the cohesion among the cells of the same type and the separation among different cell types. For both datasets shown in Figure 4.8a, the SI values of scDHA are much higher than those obtained for PCA, t-SNE, UMAP, and SCANPY. The average SI values

obtained across the 34 datasets are shown in Figure 4.8b. Overall, scDHA consistently and significantly outperforms other methods ($p = 1.7 \times 10^{-6}$). The visualization, and SI values of all datasets are shown in Figures 4.9–4.17 and Table 4.6.

We also compare the methods across different data platforms: plate-based, flow-cell-based, Smart-Seq1/2, SMARTer, inDrop, and 10X Genomics. Figure 4.18 shows the performance of the visualization methods. The silhouette values of all methods change across the platform groups. However, scDHA consistently outperforms other methods in each platform. Similar to clustering, the performance of all methods dropped when analyzing 10x Genomics.

4.3.3 Cell classification

We assess scDHA's classification capability by comparing it with four methods that are dominant in machine learning: XGBoost [81], Random Forest (RF) [82], Deep Learning (DL) [83], and Gradient Boosting Machine (GBM) [84].

We test these methods using five datasets: Baron (8,569 cells), Segerstolpe (2,209 cells), Muraro (2,126 cells), Xin (1,600 cells), and Wang (457 cells). All five datasets are related to human pancreas and thus have similar cell types. In each analysis scenario, we use one dataset as training and then classify the cells in the remaining four datasets. For example, we first train the models on Baron and then test them on Segerstolpe, Muraro, Xin, and Wang. Next, we train the models on Segerstolpe and test on the rest, etc. The accuracy of each method is shown in Figure 4.19 and Table 4.7.

Overall, scDHA is accurate across all 20 combinations with accuracy ranging from 0.88 to 1. scDHA outperforms other methods by having the highest accuracy. The average accuracy of scDHA is 0.96, compared to 0.77, 0.69, 0.43, and 0.72 for XGB, RF, DL, and GBM, respectively. In addition, scDHA is very consistent, while the

Table 4.6: Silhouette values calculated for representation using scDHA, PCA, t-SNE, UMAP, and SCANPY. Cells with NA values indicate that the method was not able to analyze the dataset (out-of-memory). Cells highlighted in green have the highest silhouette values. scDHA has the highest average silhouette value. It outperforms other methods in 25 out of 34 datasets.

Dataset	Size	scDHA	PCA	t-SNE	UMAP	SCANPY
Yan	90	0.52	0.54	0.45	0.47	0.73
Goolam	124	0.37	0.31	0.28	0.27	-0.02
Deng	268	0.50	0.60	0.49	0.67	0.45
Pollen	301	0.78	0.30	0.61	0.58	0.65
Patel	430	0.62	0.17	0.52	0.52	0.31
Wang	457	0.28	-0.07	0.13	0.21	0.27
Darmanis	466	0.47	0.01	0.31	0.34	0.25
Camp (B)	553	0.54	0.07	0.36	0.30	0.34
Usoskin	622	0.62	0.07	0.40	0.51	0.45
Kolodziejczyk	704	0.81	0.30	0.43	0.54	0.50
Camp (L)	777	0.67	0.17	0.42	0.50	0.41
Xin	1,600	0.67	0.08	0.25	0.17	0.36
Baron (M)	1,886	0.44	-0.23	0.05	0.10	0.43
Muraro	2,126	0.57	-0.20	0.24	0.46	0.24
Segerstolpe	2,209	0.66	-0.22	0.01	0.24	0.22
Klein	2,717	0.72	0.24	0.48	0.69	0.69
Romanov	2,881	0.37	0.03	0.24	0.34	0.27
Zeisel	3,005	0.67	0.03	0.31	0.55	0.34
Lake	3,042	0.35	-0.11	0.25	0.32	0.29
Puram	5,902	0.27	0.23	0.05	0.28	0.24
Montoro	7,193	0.16	0.24	0.09	0.29	0.22
Baron (H)	8,569	0.61	-0.14	0.20	0.46	0.50
Chen	12,089	0.49	-0.07	0.09	0.35	0.40
Sanderson	12,648	0.09	0.06	0.04	0.16	0.14
Slyper	13,316	0.44	0.22	0.16	0.44	0.45
Campbell	21,086	0.01	-0.31	-0.05	-0.08	0.03
Zilionis	34,558	0.44	0.00	0.22	0.42	0.29
Macosko	44,808	0.27	0.11	0.09	0.36	0.27
Hrvatin	48,266	0.73	0.36	0.26	0.59	0.46
Tabula Muris	54,439	0.11	-0.24	-0.14	-0.07	-0.17
Karagiannis	72,914	0.05	-0.08	0.05	0.17	0.11
Orozco	100,055	0.72	0.69	-0.15	0.02	0.22
Darrah	162,490	0.26	-0.36	-0.15	0.17	0.13
Kozareva	611,034	0.76	NA	NA	NA	0.50
Mean		0.47	0.08	0.21	0.33	0.32

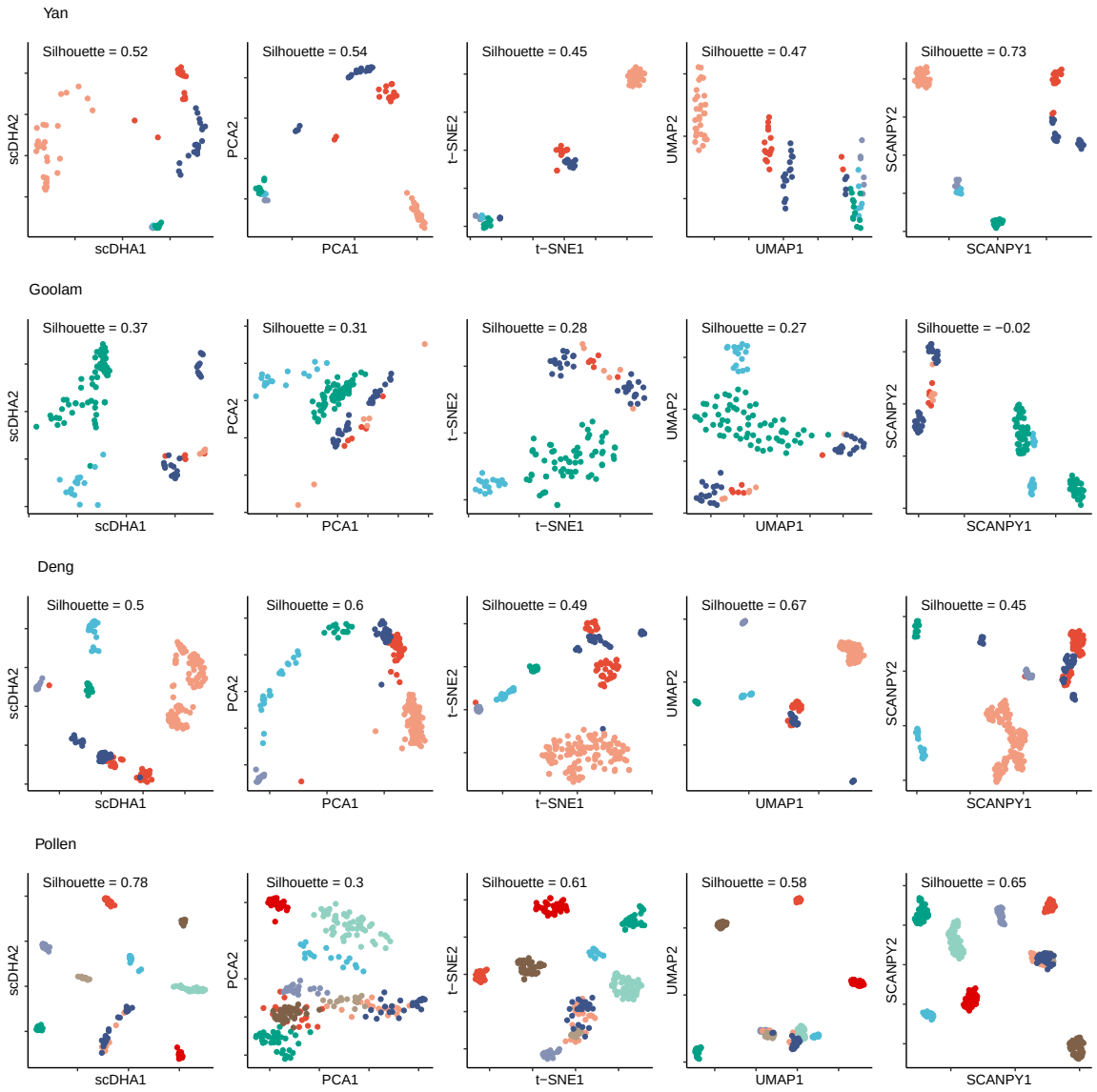

Figure 4.9: Representation of the Yan, Gollam, Deng, and Pollen datasets (top to bottom) using scDHA, PCA, t-SNE, UMAP, and SCANPY (left to right). Different colors code for different cell types.

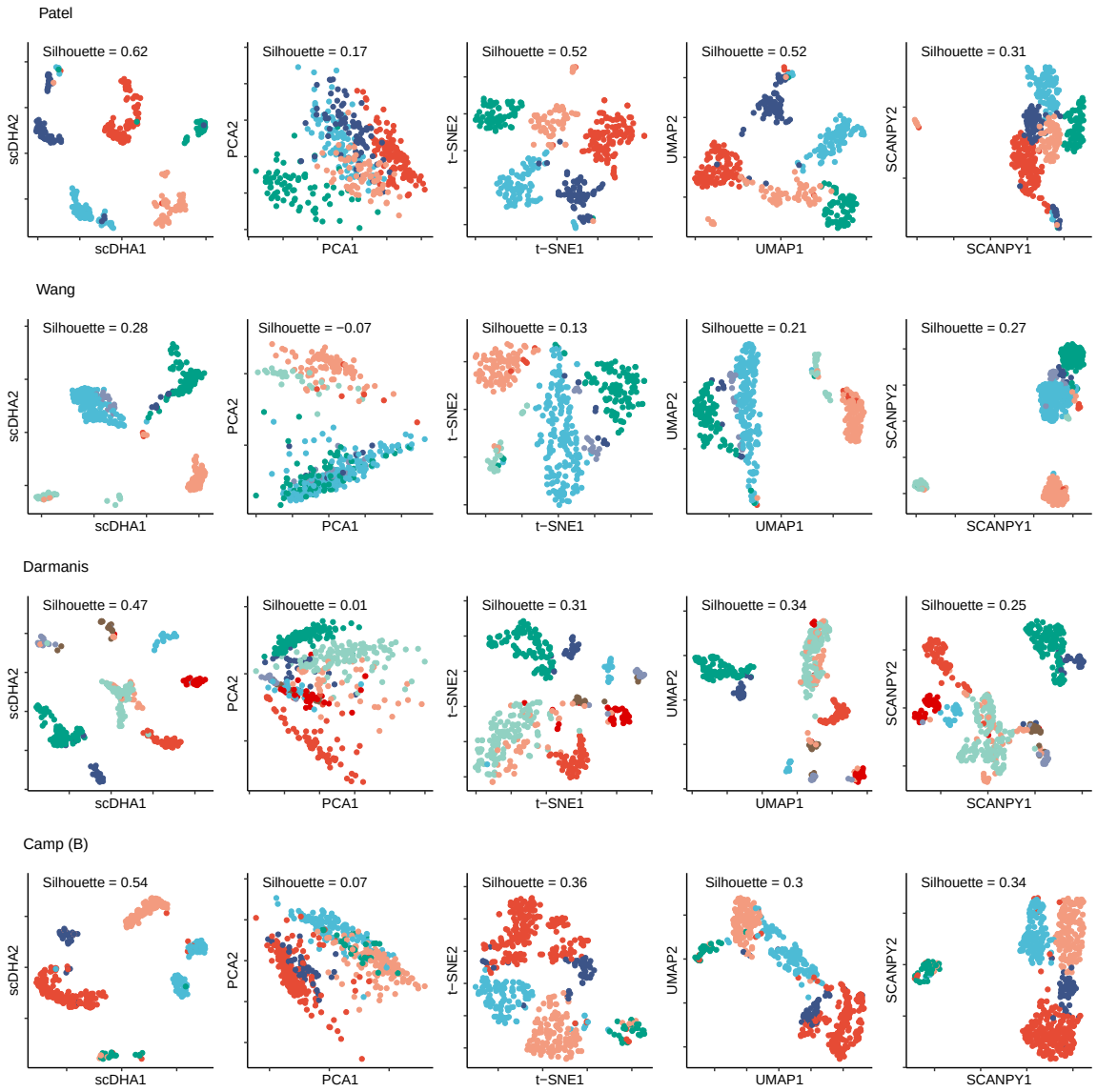

Figure 4.10: Representation of the Patel, Wang, Darmanis, and Camp (Brain) datasets (top to bottom) using scDHA, PCA, t-SNE, UMAP, and SCANPY (left to right). Different colors code for different cell types.

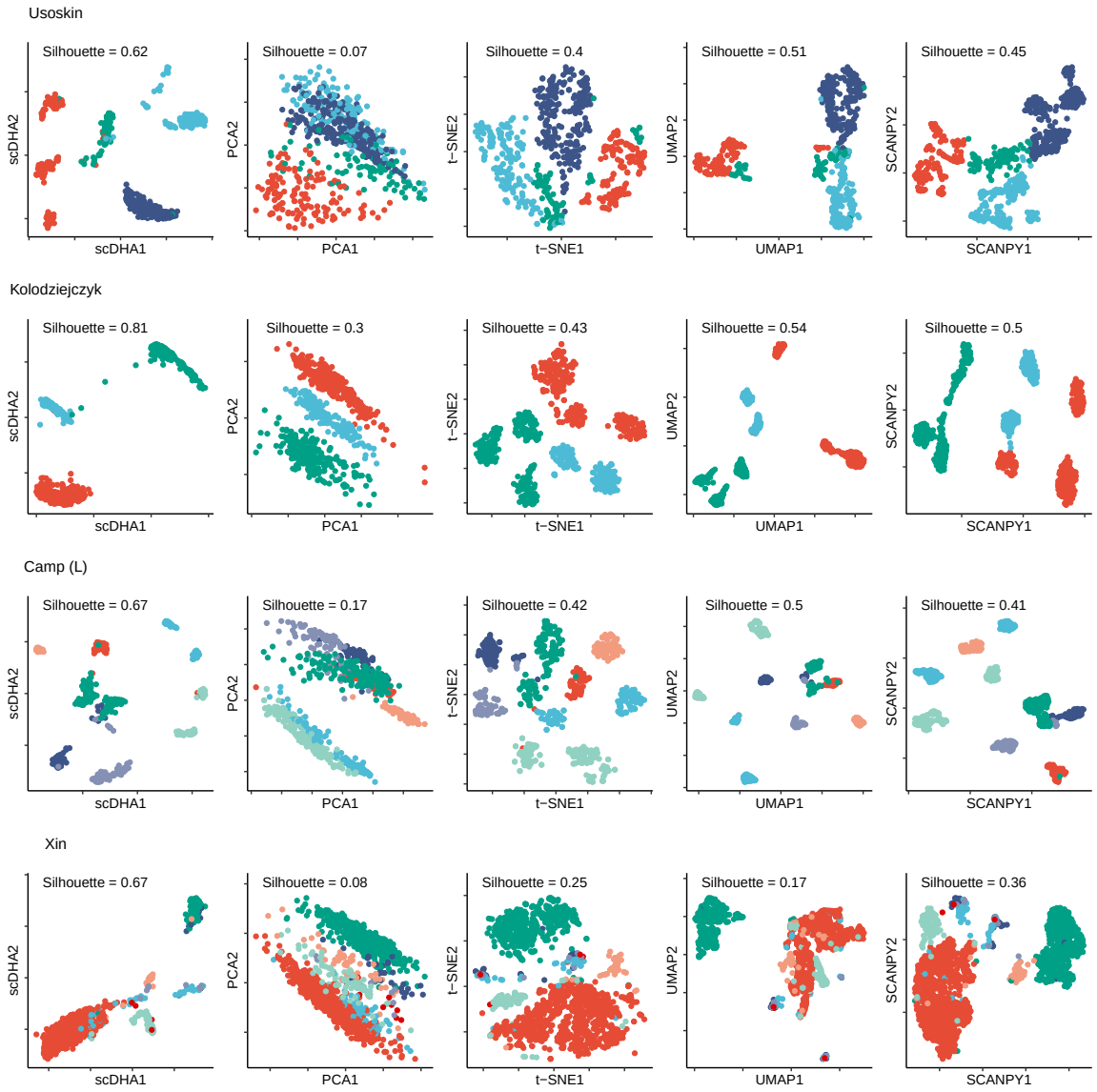

Figure 4.11: Representation of Usoskin, Kolodziejczyk, Camp (Liver), and Xin datasets (top to bottom) using scDHA, PCA, t-SNE, UMAP, and SCANPY (left to right). Different colors code for different cell types.

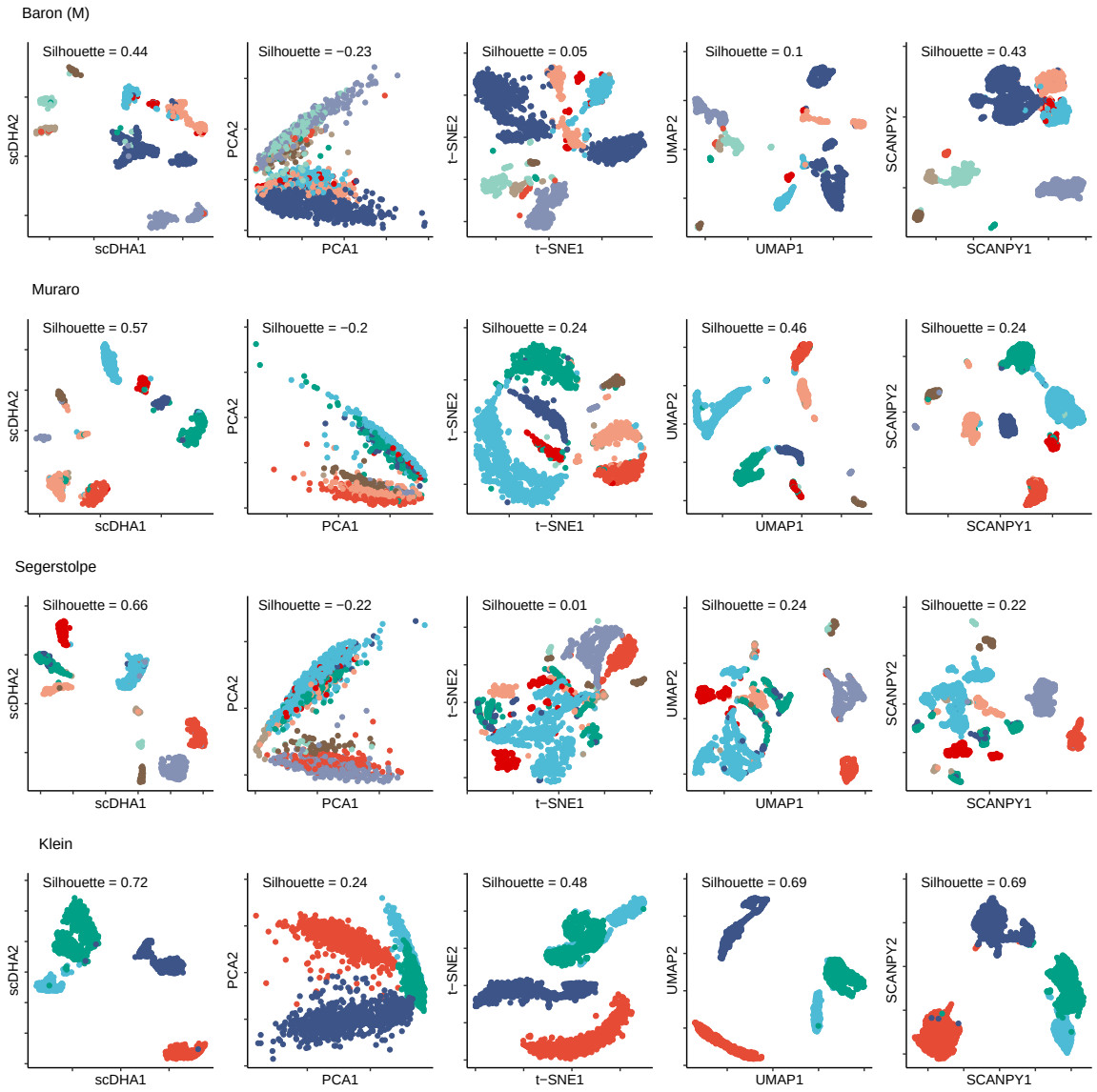

Figure 4.12: Representation of Baron (mouse), Muraro, Segerstolpe, and Klein datasets (top to bottom) using scDHA, PCA, t-SNE, UMAP, and SCANPY (left to right). Different colors code for different cell types.

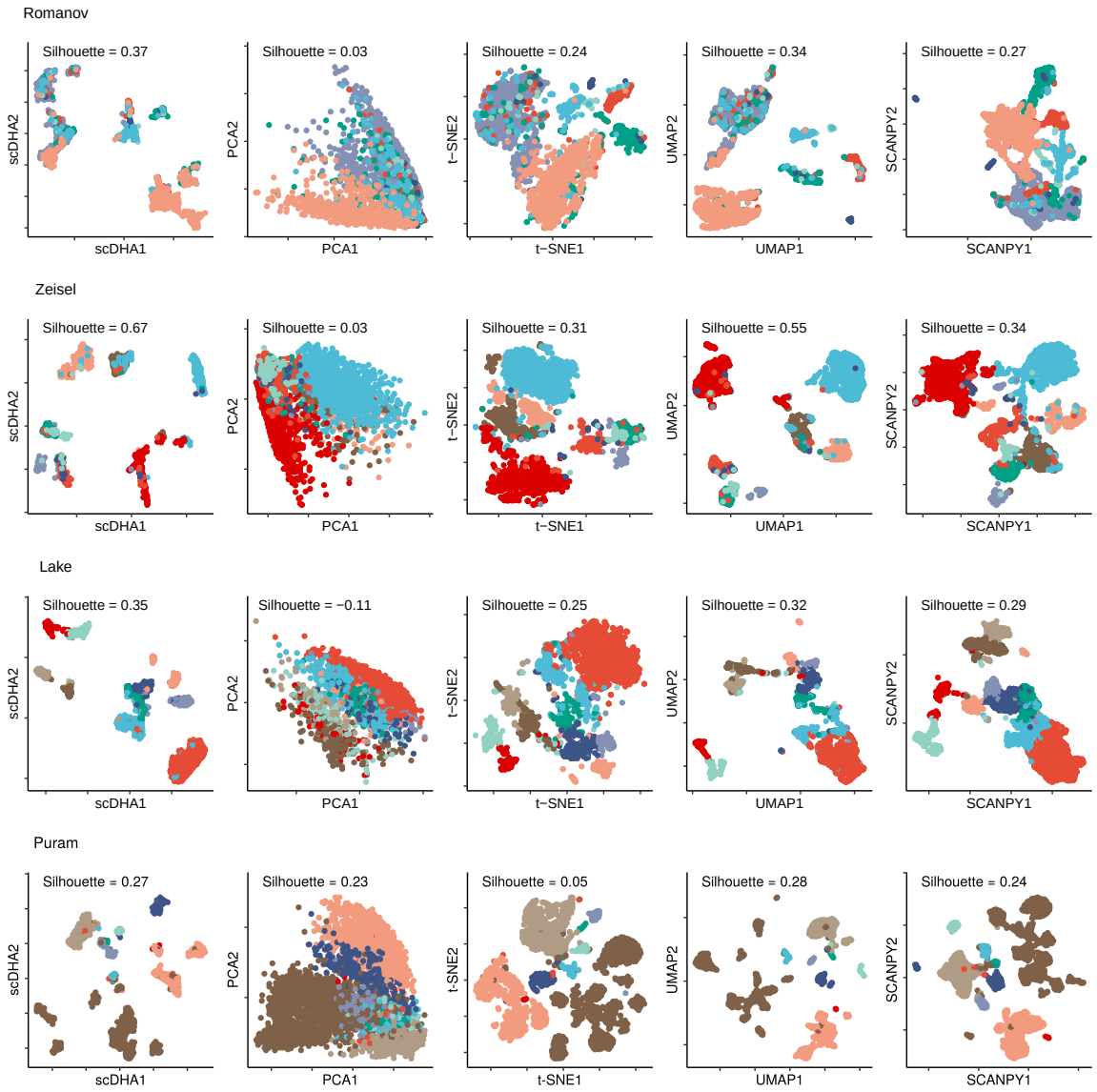

Figure 4.13: Representation of Romanov, Zeisel, Lake, and Puram datasets (top to bottom) using scDHA, PCA, t-SNE, UMAP, and SCANPY (left to right). Different colors code for different cell types.

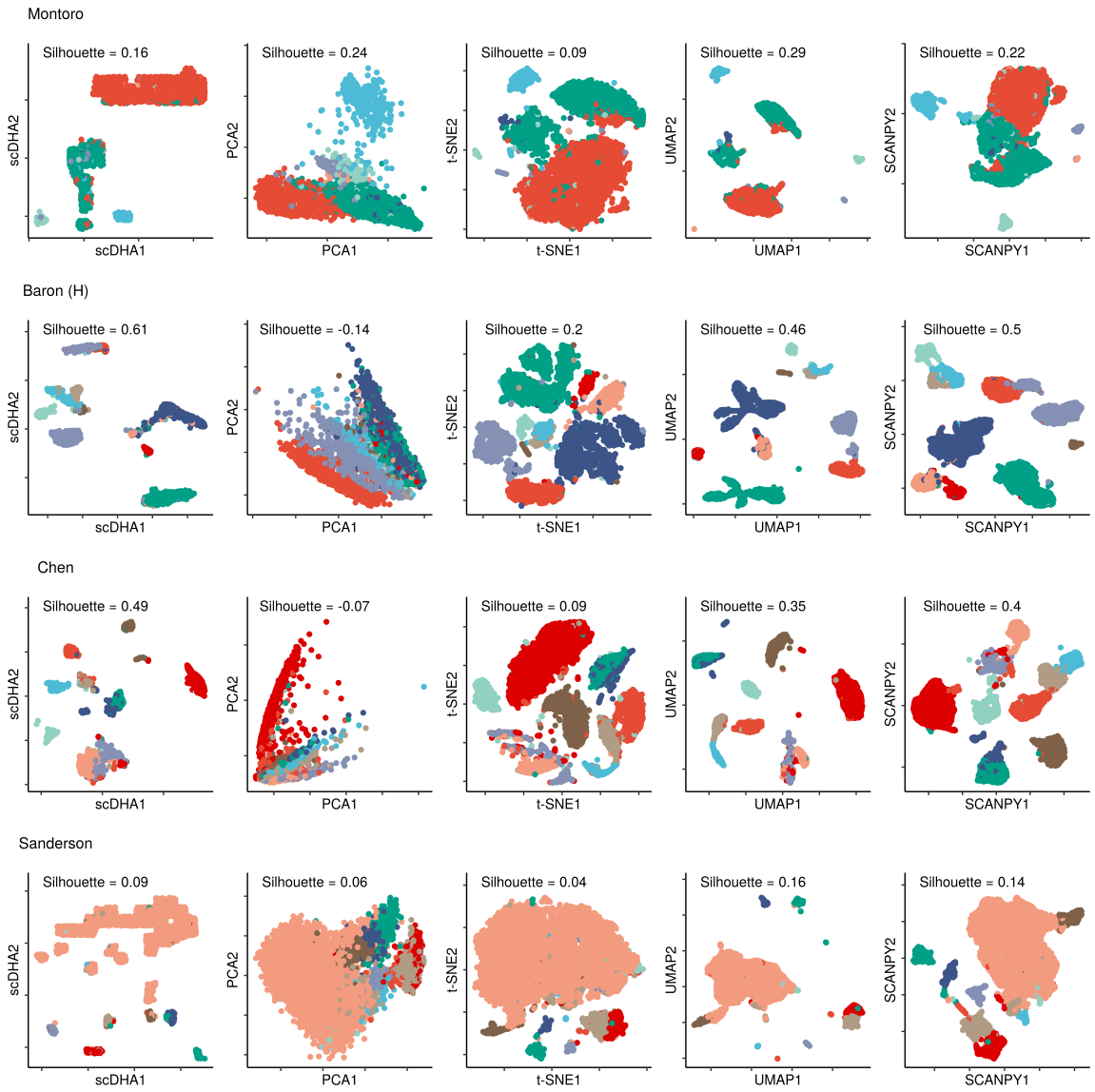

Figure 4.14: Representation of Montoro, Baron (Human), Chen, and Sanderson datasets (top to bottom) using scDHA, PCA, t-SNE, UMAP, and SCANPY (left to right). Different colors code for different cell types.

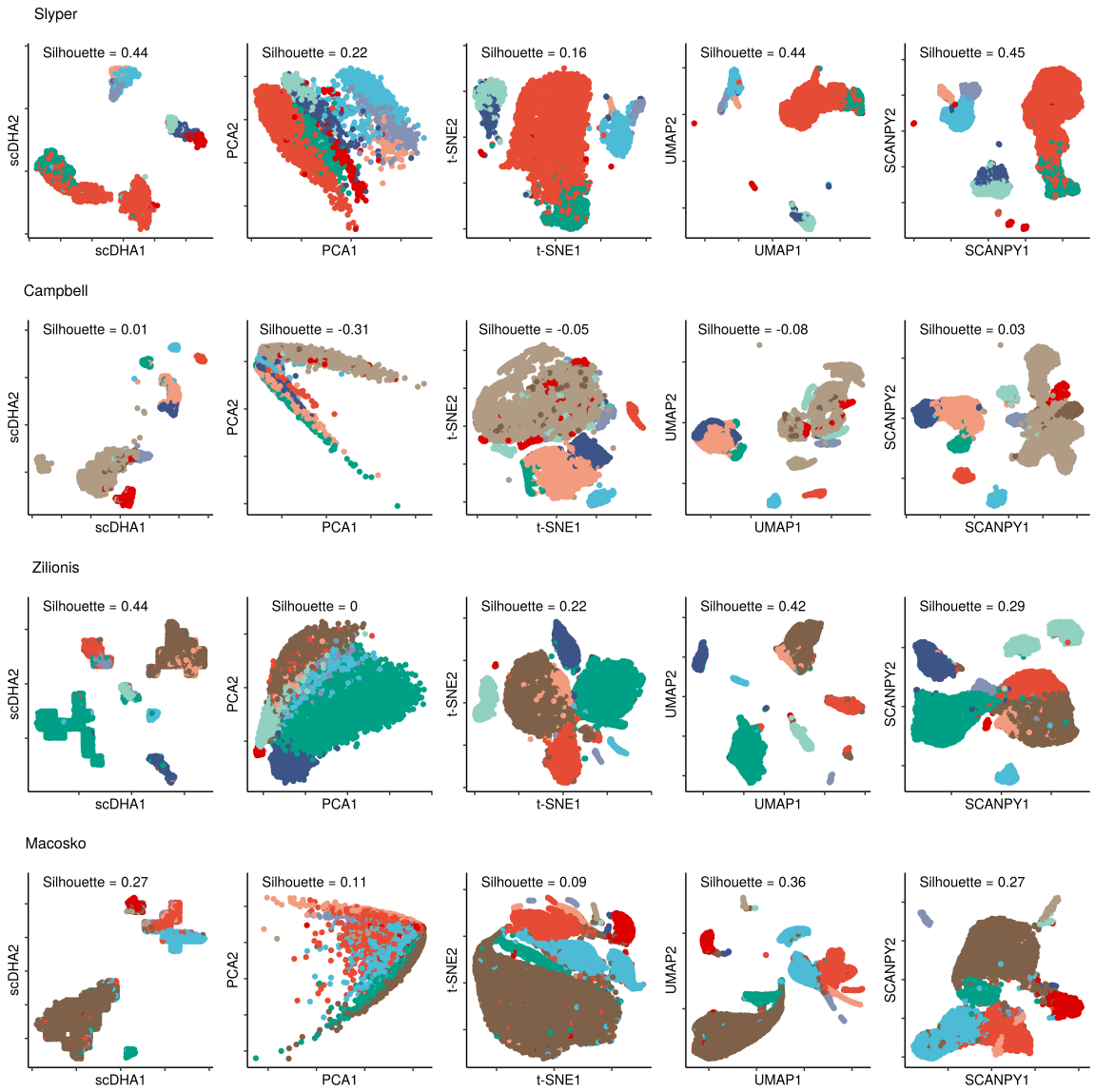

Figure 4.15: Representation of Slyper, Campbell, Zilionis, and Macosko datasets (top to bottom) using scDHA, PCA, t-SNE, UMAP, and SCANPY (left to right). Different colors code for different cell types.

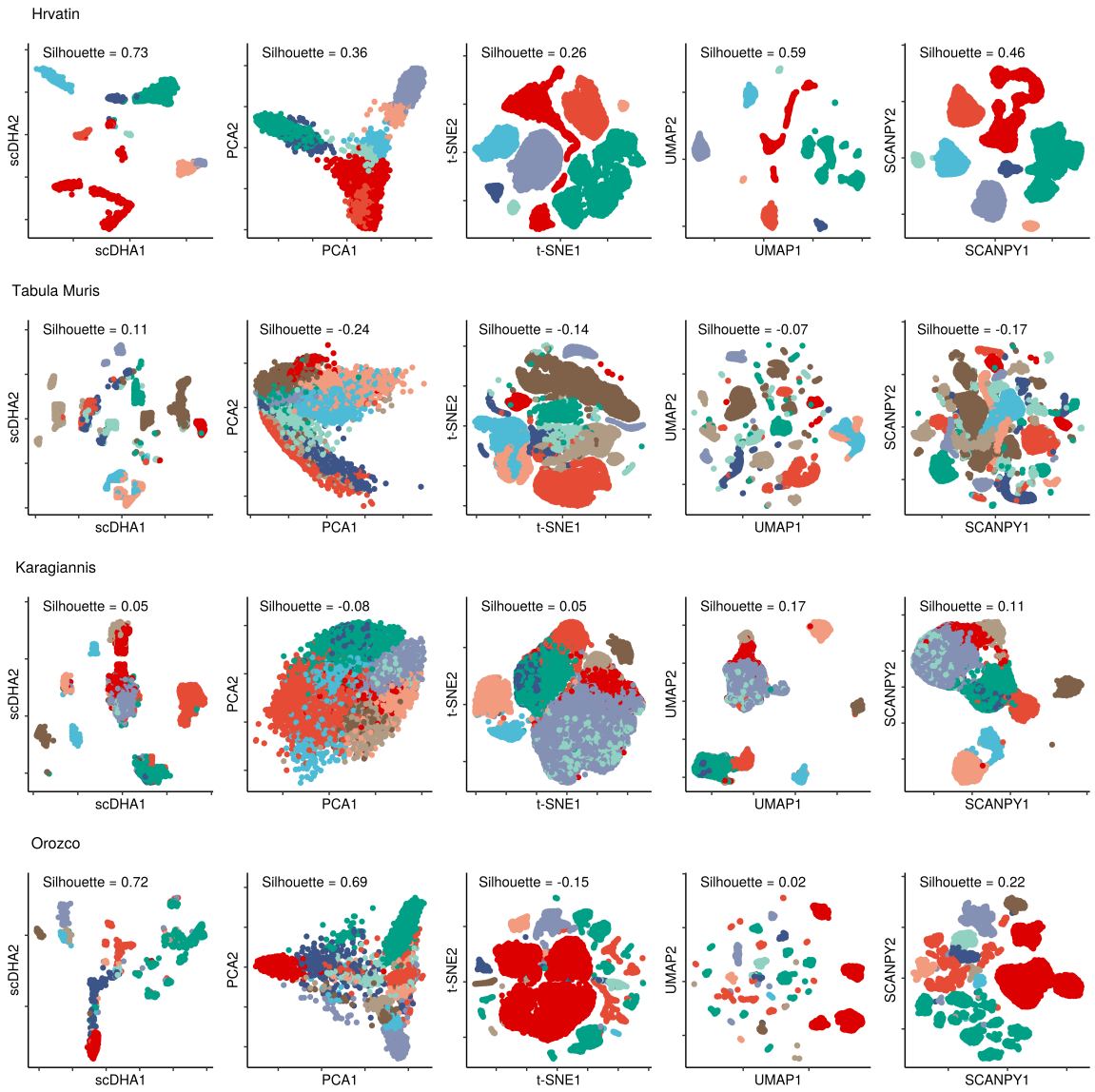

Figure 4.16: Representation of Hrvatin, Tabula Muris, Karagiannis, and Orozco datasets (top to bottom) using scDHA, PCA, t-SNE, UMAP, and SCANPY (left to right). Different colors code for different cell types.

Table 4.7: Classification performance measuring by accuracy of scDHA, XGBoost, Random Forest (RF), Deep Learning (DL), and Gradient Boosting Machine (GBM) approach on single cell evaluation pairs.

Training Dataset	Predicting Dataset	scDHA	XGBoost	RF	DL	GBM
Baron (Human)	**Segerstolpe**	0.93	0.82	0.32	0.60	0.39
Baron (Human)	**Muraro**	0.88	0.86	0.79	0.72	0.74
Baron (Human)	**Xin**	0.99	0.93	0.49	0.03	0.84
Baron (Human)	**Wang**	0.96	0.27	0.28	0.01	0.60
Segerstolpe	**Baron (Human)**	0.94	0.83	0.71	0.21	0.49
Segerstolpe	**Muraro**	0.96	0.81	0.88	0.73	0.74
Segerstolpe	**Xin**	0.99	1	0.97	0.46	0.99
Segerstolpe	**Wang**	0.99	0.98	0.93	0.22	0.97
Xin	**Baron (Human)**	0.99	0.55	0.60	0.77	0.46
Xin	**Segerstolpe**	0.99	0.98	0.91	0.78	0.92
Xin	**Muraro**	0.97	0.70	0.82	0.57	0.42
Xin	**Wang**	1	1	0.58	0.58	0.96
Muraro	**Baron (Human)**	0.93	0.86	0.78	0.16	0.85
Muraro	**Segerstolpe**	0.97	0.93	0.65	0.65	0.72
Muraro	**Xin**	0.99	0.88	0.89	0.06	0.84
Muraro	**Wang**	0.98	0.85	0.64	0.01	0.73
Wang	**Baron (Human)**	0.93	0.14	0.38	0.30	0.38
Wang	**Segerstolpe**	0.92	0.90	0.75	0.44	0.91
Wang	**Muraro**	0.89	0.13	0.55	0.46	0.52
Wang	**Xin**	0.97	1	0.90	0.76	0.96

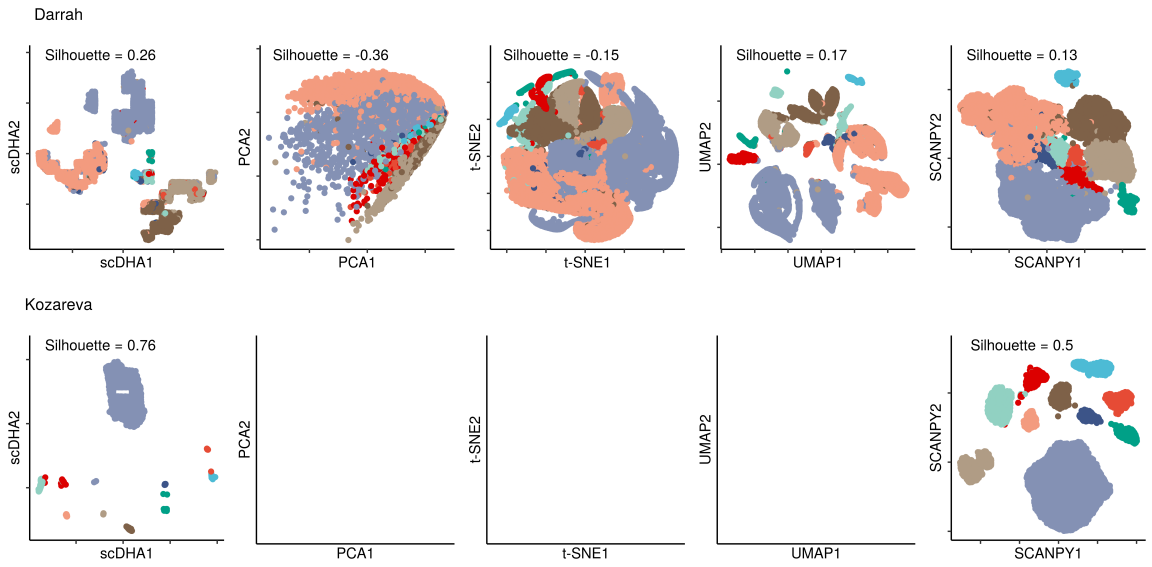

Figure 4.17: Representation of Darrah, and Kozareva datasets (top to bottom) using scDHA, PCA, t-SNE, UMAP, and SCANPY (left to right). Different colors code for different cell types. For Kozareva dataset, only scDHA and SCANPY can generate the 2D representation.

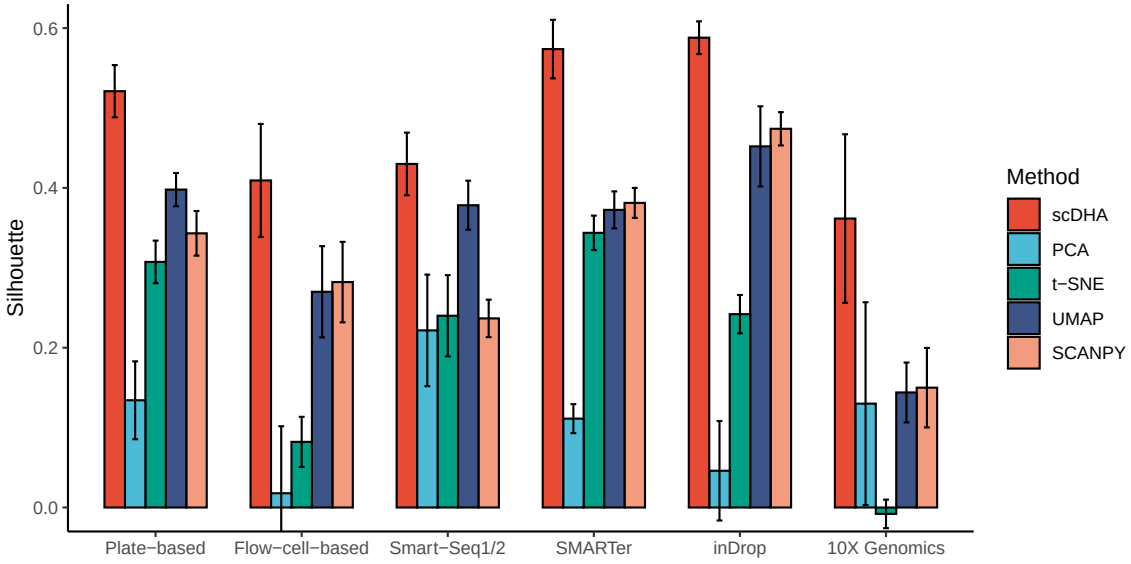

Figure 4.18: **Average silhouette values obtained from 2D representations across six data platforms.** Data are presented as mean values +/- variance

performance of existing methods fluctuates from one analysis to another, especially when the testing dataset is much larger than the training dataset. For example,

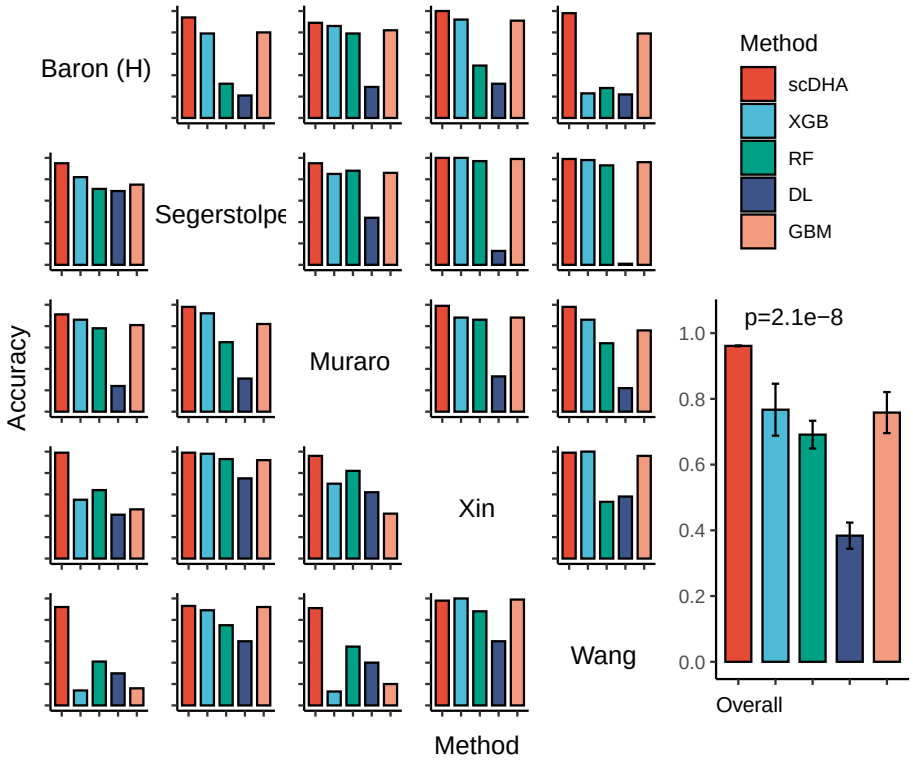

Figure 4.19: Classification accuracy of scDHA, XGBoost, Random Forest (RF), Deep Learning (DL), Gradient Boosted Machine (GBM) using five human pancreatic datasets. In each scenario (row), we use one dataset as training and the rest as testing, resulting in 20 train-predict pairs. The accuracy values of scDHA are significantly higher than those of other methods ($p = 2.1 \times 10^{-8}$ using Wilcoxon one-tailed test).

when the testing set (Baron) is 20 times larger than the training set (Wang), the accuracy of existing methods is close to 30%, while scDHA achieves an accuracy of 0.93. The one-sided Wilcoxon test also confirms that the accuracy values of scDHA are significantly higher than the rest ($p = 2.1 \times 10^{-8}$).

4.3.4 Time-trajectory inference

Here we compare the performance of scDHA with state-of-the-art methods for time-trajectory inference: Monocle [88], TSCAN [89], Slingshot [64], and SCANPY [77]. We test scDHA and these methods using three mouse embryo development datasets:

Yan, Goolam, and Deng. The true developmental stages of these datasets are only used *a posteriori* to assess the performance of the methods.

Figure 4.20a shows the Yan dataset in the first two t-SNE components. The smoothed lines shown in each panel indicate the time-trajectory of scDHA (left) and Monocle (right). The trajectory inferred by scDHA accurately follows the true developmental stages: it starts from zygote, going through 2cell, 4cell, 8cell, 16cell, and then stops at the blast class. On the contrary, the trajectory of Monocle goes directly from zygote to 8cell before coming back to 2cell. Figure 4.20b shows the cells ordered by pseudo-time. The time inferred by scDHA is strongly correlated with the true developmental stages. On the other hand, Monocle fails to differentiate between zygote, 2cell, and 4cell. To quantify how well the inferred trajectory explains the developmental stages, we also calculate the R-squared value. scDHA outperforms Monocle by having a higher R-squared value (0.93 compared to 0.84).

Figure 4.20c,d show the results of the Goolam dataset. scDHA correctly reconstructs the time-trajectory whereas Monocle fails to estimate pseudo-time for 8cell, 16cell, and blast cells (colored in gray). Monocle assigns an "infinity" value for these cell classes. Figure 4.20e,f show the results obtained for the Deng dataset. Similarly, the time-trajectory inferred by scDHA accurately follows the developmental stages whereas Monocle cannot estimate the time for half of the cells. The results of TSCAN, Slingshot, and SCANPY are shown in Figures 4.20 and 4.20. scDHA outperforms all three methods by having the highest R-squared values in every single analysis.

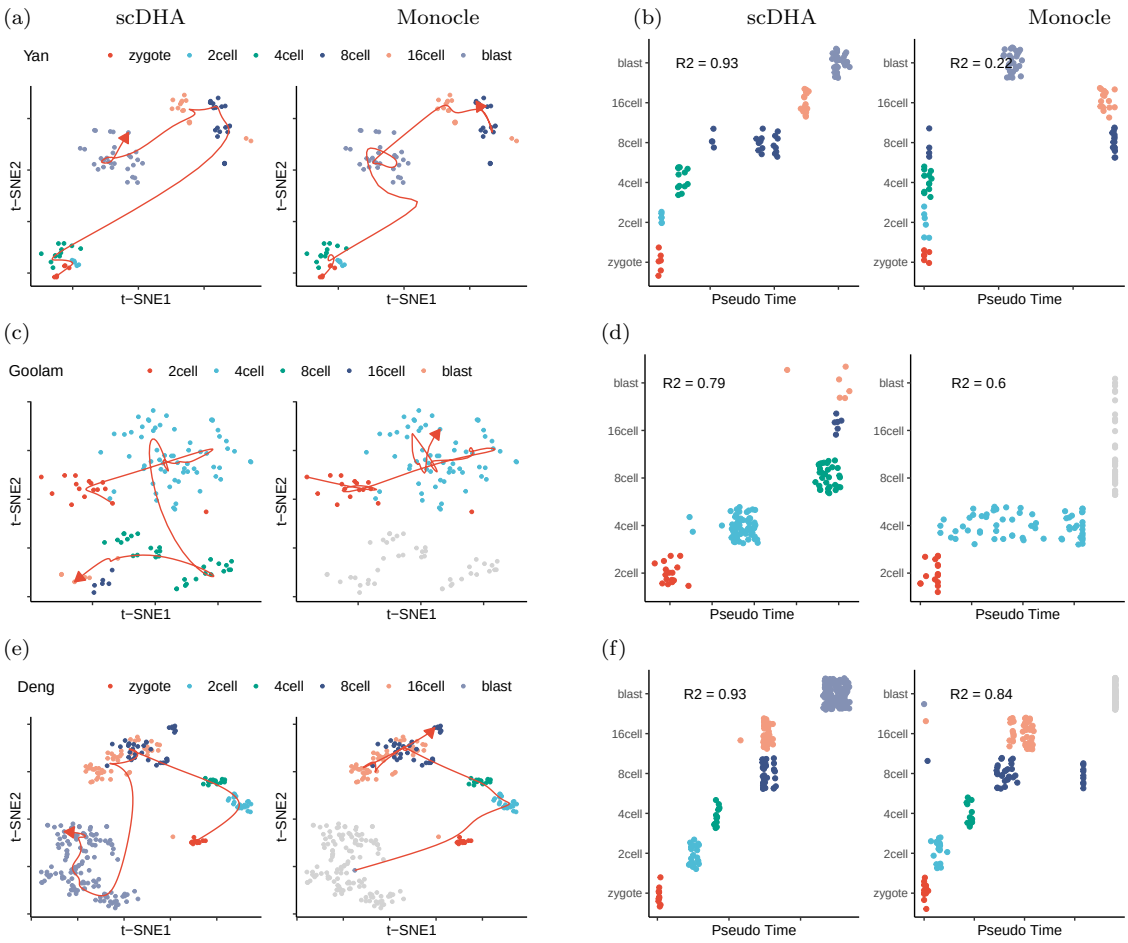

Figure 4.20: Pseudo-time inference of three mouse embryo development datasets (Yan, Goolam, and Deng) using scDHA and Monocle. (a) Visualized time-trajectory of the Yan dataset in the first two t-SNE dimensions using scDHA (left) and Monocle (right). (b) Pseudo-temporal ordering of the cells in the Yan dataset. The horizontal axis shows the inferred time for each cell while the vertical axis shows the true developmental stages. (c,d) Time-trajectory of the Goolam dataset. Monocle is unable to estimate the time for most cells in 8-cell, 16-cell, and blast (colored in gray). (e,f) Time-trajectory of the Deng dataset. Monocle is unable to estimate the pseudo time for most blast cells.

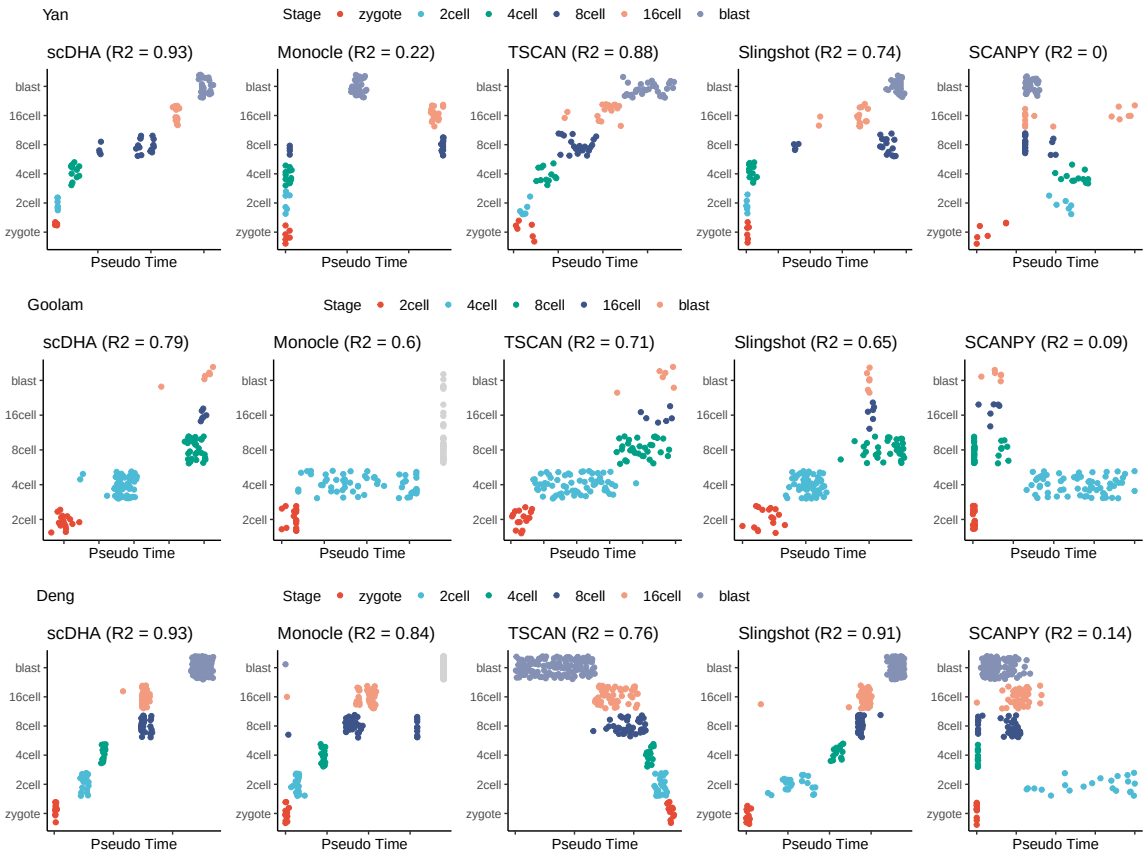

Figure 4.21: Pseudo-time inferred by scDHA, Monocle, TSCAN, Slingshot, and SCANPY for the Yan, Goolam, and Deng datasets. R-squared values shown in each panel represent the correlation between the true developmental stages and inferred pseudo-time. Points with gray color indicate cells with infinite pseudo-time.

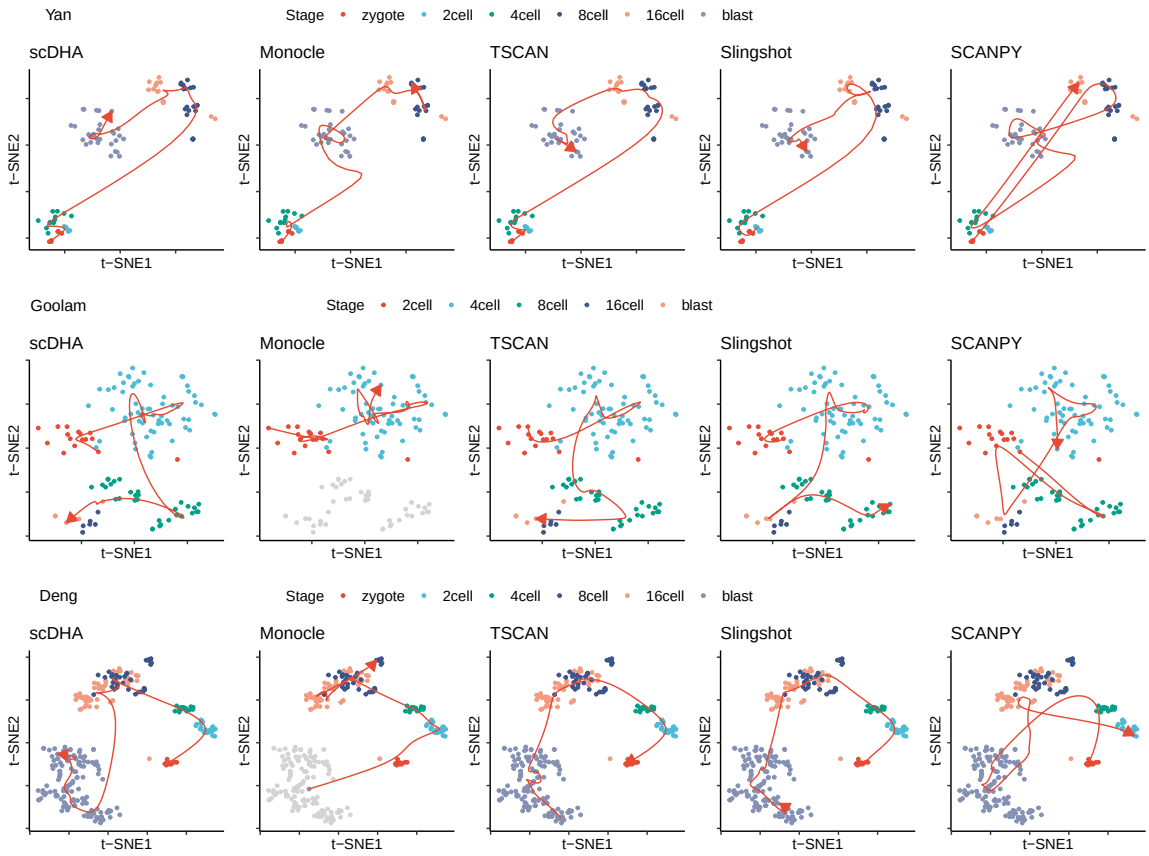

Figure 4.22: Visualized trajectory inferred from Yan, Goolam, and Deng dataset using scDHA, Monocle, TSCAN, Slingshot, and SCANPY. Points with gray color mean cells with infinity pseudo time from Monocle.

4.4 Conclusion (scDHA)

The ever-increasing number of cells, technical noise, and high dropout rate pose significant computational challenges in scRNA-seq analysis. These challenges affect both analysis accuracy and scalability, and greatly hinder our capability to extract the wealth of information available in single-cell data. To detach noise from informative biological signals, we have introduced scDHA, a powerful framework for scRNA-seq data analysis. We have shown that the framework can be utilized for both upstream and downstream analyses, including *de novo* clustering of cells, visualizing the transcriptome landscape, classifying cells, and inferring pseudo-time. We demonstrate that scDHA outperforms state-of-the-art techniques in each research sub-field. Although we focus on single-cell as an example, scDHA is flexible enough to be adopted in a range of research areas, from cancer to obesity to aging to any other area that employs high-throughput data.

In contrast to existing autoencoders, such as scVI [150] that was developed for data imputation, scDHA provides a complete analysis pipeline from feature selection (first module) to dimension reduction (second module) and downstream analyses (visualization, clustering, classification, and pseudo-time inference). The scVI package itself is not capable of clustering, visualization, classification, and pseudo-time inference. Even for the implementation of autoencoder, there are two key differences between scDHA and scVI. First, scDHA implements a hierarchical autoencoder that consists of two modules: the first autoencoder to remove noise (denoising), and the second autoencoder to compress data. The added denoising module (first module) filters out the noisy features and thus improves the quality of the data. Second, we modify the standard variational autoencoder (VAE, second module) to generate multiple realizations of the input. This step makes the VAE more robust. Indeed, our analysis results show that scDHA and its second module consistently outperform scVI

when scVI is used in conjunction with downstream analysis methods implemented in scDHA and other packages (see Supplementary Section 6 and Fig. 25–32).

In summary, scDHA is user-friendly and is expected to be more accurate than existing autoencoders. Users can apply scDHA to perform downstream analyses without installing additional packages for the four analysis applications (clustering, visualization, classification, and pseudo-time trajectory inference). At the same time, the hierarchical autoencoder and the modified VAE (second module of scDHA) are expected to be more efficient than other autoencoders in single-cell data analysis.

Chapter 5

scISR: A Novel Method for Single-cell Data Imputation using Subspace Regression

This chapter is based on the following publication: ***Duc Tran****, Bang Tran, Hung Nguyen, and Tin Nguyen. A novel method for single-cell data imputation using subspace regression. Scientific Reports, 2022. DOI: 10.1038/s41598-022-06500-4*

Recent advances in biochemistry and single-cell RNA sequencing (scRNA-seq) have allowed us to monitor the biological systems at the single-cell resolution. However, the low capture of mRNA material within individual cells often leads to inaccurate quantification of genetic material. Consequently, a significant amount of expression values are reported as missing, which are often referred to as dropouts. To overcome this challenge, we develop a novel imputation method, named single-cell Imputation via Subspace Regression (scISR), that can reliably recover the dropout values of scRNA-seq data. The scISR method first uses a hypothesis-testing technique to identify zero-valued entries that are most likely affected by dropout events

and then estimates the dropout values using a subspace regression model. Our comprehensive evaluation using 25 publicly available scRNA-seq datasets and various simulation scenarios against five state-of-the-art methods demonstrates that scISR is better than other imputation methods in recovering scRNA-seq expression profiles via imputation. scISR consistently improves the quality of cluster analysis regardless of dropout rates, normalization techniques, and quantification schemes. The source code of scISR can be found on CRAN at `https://cran.r-project.org/package=scISR`.

5.1 Introduction

Bulk RNA sequencing (RNA-seq) has been the primary tool to study biological systems. Despite its popularity, bulk sequencing is unable to measure the heterogeneity inside complex tissues and cell-to-cell variability. Recently, advances in microfluidics and sequencing technologies have allowed us to measure the expression profiles of individual cells [58, 59]. By allowing us to monitor the biological processes at the single-cell resolution, single-cell technologies (scRNA-seq) have enabled new research directions in genomics and transcriptomics research. However, scRNA-seq data also comes with additional challenges [73]. One of the challenges is that sequencing mRNA within individual cells requires artificial amplification of DNA materials, leading to disproportionate distortions of relative transcript abundance and gene expression. Another outstanding challenge is the "dropout" phenomenon where a gene is highly expressed in one cell but does not express at all in another cell [101]. These dropout events usually occur due to the limitation of sequencing technologies when only a small amount of starting mRNA in individual cells can be captured, leading to low sequencing depth and failed amplification [102, 103]. Since downstream analyses of scRNA-seq heavily rely on the accuracy of expression measurement, it is crucial to impute the zero expression values introduced by the dropout phenomenon and se-

quencing errors.

There have been a number of computational methods developed to impute single-cell data. These imputation methods can be classified into two categories: i) model-based methods and ii) model-free methods. Methods in the first category model the data using a mixture of two different distributions: one distribution represents the actual gene expression while the other accounts for the dropout events. Next, they estimate the model parameters and true expression values using the Expectation-Maximization (EM) algorithm [151]. Methods in this category include scImpute [105], SAVER [106], and BISCUIT [152]. scImpute uses a Gaussian distribution to model the actual expression and a Gamma distribution to model the dropout events. It estimates the model parameters and dropout values using the EM algorithm. Similarly, SAVER [106] models read counts as a mixture of Poisson-Gamma distribution and then uses a Bayesian approach to estimate the true expression values. BISCUIT [152] uses the Dirichlet process mixture model [153] to perform data normalization, cells clustering, and dropouts imputation by simultaneously inferring clustering parameters, estimating technical variations (e.g., library size), and learning co-expression structures of each cluster.

Methods in the second category typically assume that expression values from the same dataset follow a certain data structure (manifold), whereas dropout events move the values away from the underlying structure. These methods use regression techniques to infer missing values from genes or cells that have similar expression patterns. Methods in this category include MAGIC [104], DrImpute [107], scScope [154], DCA [155], and DeepImpute [156]. MAGIC imputes zero values using heat diffusion [157]. The method first computes the affinity matrix between cells using a Gaussian kernel and then constructs the Markov transition matrix by normalizing and smoothing the computed affinity matrix. Finally, the method multiplies the

exponentiated Markov matrix with the original data to obtain the imputed data. DrImpute [107] uses a cluster ensemble strategy and consensus clustering to separate data into groups of similar cells and then imputes missing data by averaging expression values of similar cells. The other three methods (scScope, DeepImpute, and DCA) rely on deep neural networks to denoise the data and to impute the missing values. scScope uses a recurrent network layer to iteratively impute the zero-valued entries while DeepImpute randomly splits genes into subsets and builds sub-neural networks to estimate the missing values. DCA, on the other hand, extends the standard autoencoder to account for sparse count data by incorporating a noise model into their loss function.

The quality of data imputed by methods in the first category (statistical methods) is determined by the validity of the assumption of the distribution models. In addition, these methods usually require excessive computational power, which makes them slow in processing big datasets. Therefore, these statistical methods often rely on gene filtering steps to ease the computational burden. For methods in the second category (regression approaches), their major drawbacks include i) relying on many parameters to fine-tune their models, which can lead to overfitting, and ii) tending to over-smoothen and remove the cell-to-cell stochasticity that represents meaningful biological variations in gene expression. More importantly, in addition to the limitations mentioned above, methods in both categories attempt to alter the expression of all zero-valued entries, including those not affected by dropout events. This may introduce false signals and further weaken their reliability.

5.2 Methodology

Here we propose a new approach, scISR, that can reliably impute missing values from single-cell data. Our method consists of three modules. The first module performs

hypothesis testing to identify the values that are likely to be impacted by the dropout events. By not altering the true zero values, we can avoid false imputations. The second module utilizes a data perturbation technique [44] to automatically group genes with similar patterns into smaller groups. The third module imputes missing values affected by dropout events (identified in the first module) by learning the gene patterns in each gene group (identified in the second module). This strategy ensures that the true missing values are imputed by using only highly relevant information. In an extensive analysis using simulation and 25 real scRNA-seq datasets, we demonstrate that scISR improves the quality of clustering analysis of single-cell data while preserving the transcriptome landscape.

The schematic pipeline of scISR is shown in Figure 5.1. Our method consists of three modules. The first module performs hypothesis testing to identify the values that are likely to be impacted by the dropout events. By not altering the true zero values, we can avoid false imputations. The second module utilizes a data perturbation technique [44] to automatically group genes with similar patterns into smaller groups. The third module imputes missing values affected by dropout events (identified in the first module) by learning the gene patterns in each gene group (identified in the second module). This strategy ensures that the true missing values are imputed by using only highly relevant information. The details of each module are provided in the the following subsections.

5.2.1 Hyper-geometric testing (Module 1)

The first module aims at determining whether each zero value observed is the result of dropouts. Our hypothesis is that dropout events happen randomly for a gene affected by this phenomenon. By treating each cell as an instance of the population, we also assume that the ratio of zero values (dropout probability) reported for each

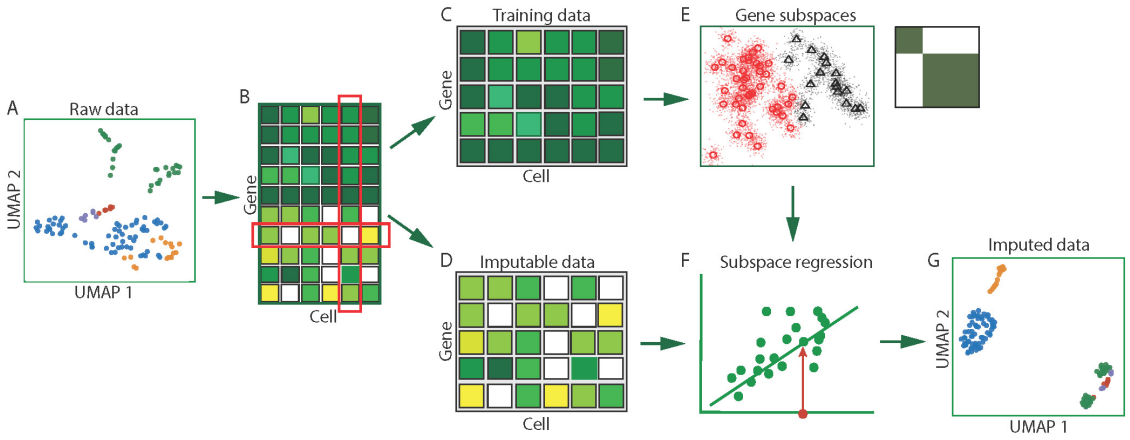

Figure 5.1: Single-cell Imputation using Subspace Regression (scISR). (A) Input data visualized in cell/sample space. (B) Hypergeometric test to determine whether each zero value is induced by dropout. Based on the computed p-values for each entry, we separate the original data into two sets of data: training data and imputable data. (C) Training data in which none of the values is induced by dropout events. (D) Imputable data in which each gene has at least one entry that is likely to be induced by dropout events. (E) Gene subspaces determined by perturbation clustering. We perturb the training data to discover the natural structure of the genes. Based on the pair-wise similarity between genes, we separate genes into groups that share similar patterns. (F) Subspace regression. We assign each gene in the imputable data to the closest subspace and then perform a generalized linear regression on the subspace to estimate the zero-valued entries that are impacted by dropouts. (G) Output expression matrix obtained by concatenating the training data and imputed data.

cell differ from each other. Using dropout probabilities from both genes and cells, we can calculate how likely each zero values is affected by dropout. If zero values caused by dropout are over-represented in a gene, we conclude that this gene is affected by dropout events.

Given a zero-valued entry, let us denote p_1 and p_2 as the probability of observing a zero value in the corresponding gene and cell, respectively. It follows that the chances of having zero values in a gene and in a cell follow binomial distributions denoted by $X \sim Bin(n, p_1)$ and $Y \sim Bin(m, p_2)$, respectively. n is the number of measured values for a gene, and m is the number of measured values for a cell. Under the null, we have $p = p_1 = p_2$. If X and Y are independent, we have $X + Y \sim Bin(n+m, p)$. Therefore, the conditional distribution of X, $P(X = x|X + Y = r)$, is a hyper-geometric where x is the number of observed zero values in the gene and r is the total number of observed zero values in the selected pair of gene and cell. The probability mass function of the hyper-geometric distribution can be written as follows:

$$P(X = x - 1|X + Y = r - 1) = \frac{\binom{n-1}{x-1}\binom{m}{r-x}}{\binom{n+m-1}{r-1}} \tag{5.1}$$

Note that X and Y have an overlapping entry for each gene and cell pair. Therefore, we remove the overlapping entry from the hypergeometic formula by using: i) $n+m-1$ (instead of $n+m$) as the total number of of observed values in the selected pair of gene and cell, ii) $n-1$ (instead of n) as the number of measured values for the gene, and iii) $x-1$ (instead of x) as the number of zero values observed in the gene.

Applying Equation (5.1), we calculate the p-value for every zero-valued. We perform two different kinds of tests: an under-representation and over-representation analysis with a significance threshold set to 0.01 for both analyses. An entry with a significant p-value in the over-representation analysis is considered untrustworthy

and should be imputed (imputable). An entry with a significant p-value in the under-representation analysis is considered trustworthy. An entry that is neither trustworthy nor untrustworthy should be left alone. These values will not be imputed, nor be used to impute other values. A gene is trustworthy if all of its entries are trustworthy. A gene is imputable when at least one of its values is imputable. Based on this hypothesis testing procedure, we obtain a set of genes that can be used for training (training data), and a set of genes that needed to be imputed (imputable data).

5.2.2 Identifying gene subspaces (Module 2)

It is crucial that the missing values of a gene are inferred using related genes that share similar expression patterns. Therefore, this module aims at identifying gene groups of the training data, i.e., gene subspaces that share similar patterns. For this purpose, we utilize the perturbation clustering [44, 45] that we recently developed. The method is based on the observation that small changes in quantitative assays will be inherently presented even when there is no significant difference between genes. If distinct gene groups do exist, they must be stable with respect to small degrees of data perturbation. This is indeed the case, as we have demonstrated in our previous work that the pair-wise connectivity between data points of the same group is preserved when the data are perturbed.

We will describe this approach using an illustrative example shown in Figure 5.2. In this simulated dataset, we have three distinct classes of genes in which the expressions of genes in each class are generated using a standard normal distribution. This distribution for the first class is $\mathcal{N}(0,1)$, for the second class is $\mathcal{N}(1,1)$ to simulate up-regulated genes, and for the third class is $\mathcal{N}(-1,1)$ to simulate down-regulated genes.

Assuming that we do not know the number of classes in this dataset, we set $k = 2$

(number of clusters) and then partition the genes. The upper panel in Figure 5.2B shows the connectivity between genes after clustering: green when they belong to the same cluster, and white otherwise. Note that two of the three true classes are wrongfully grouped together due to the wrong number of clusters. Now we repeatedly perturb the molecular measurements (by adding Gaussian noise) and partition the genes again (still with $k = 2$). The lower panel in Figure 5.2B shows the average connectivity between genes when the data is perturbed. The perturbed connectivity matrix suggests that the larger cluster is not stable. Similarly, the discordant connectivity in Figure 5.2C states that the partitioning using $k = 5$ is not correct either. The perturbed connectivity matrices (Figure 5.2B, C) suggest that there are three distinct classes of genes. Finally, when we set $k = 3$, the perturbed and original connectivity matrices are identical (Figure 5.2D).

The perturbed connectivity matrices suggest that there are three distinct classes of genes. This demonstrates that for truly distinct gene groups the true connectivity between genes within each class is recovered when the data is perturbed, no matter how we set the value of k. This resilience of pair-wise connectivity occurs consistently regardless of the clustering algorithm being used (e.g., k-means, hierarchical clustering, or partitioning around medoids), or the distribution of the data. When there are no truly distinct subgroups, the connectivity is randomly distributed. When the number of true classes changes, the perturbed connectivity always reflects the true structure of the data.

To identify the optimal partitioning, we calculate the absolute difference between the original and the perturbed connectivity matrices and compute the empirical cumulative distribution functions of the entries of the difference matrix (CDF-DM). In the ideal case of perfectly stable clusters, the original and perturbed connectivity matrices are identical, yielding a difference matrix of 0s, a CDF-DM that jumps from

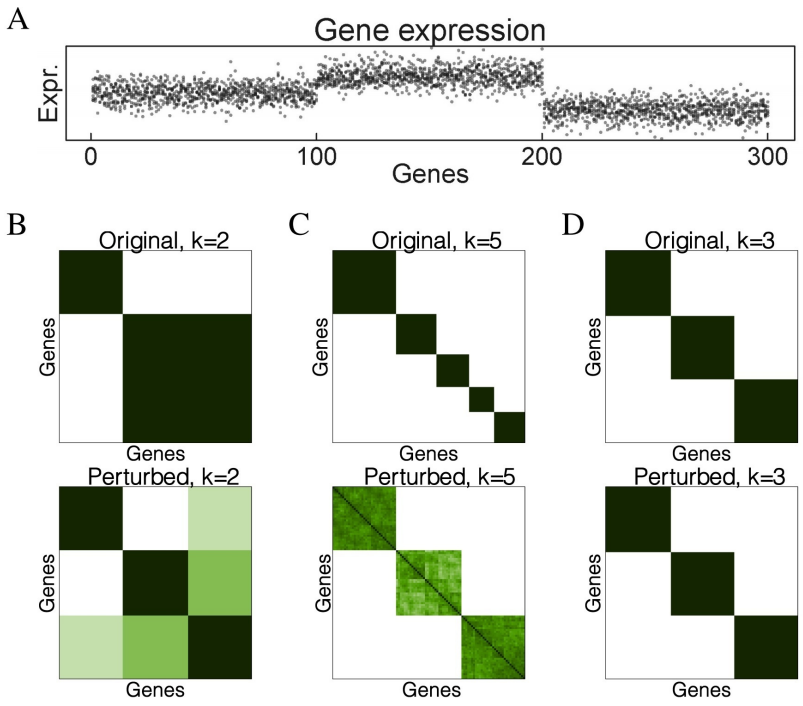

Figure 5.2: The resilience of pair-wise connectivity. (A) The dataset consists of three classes of genes: the first class has expression values of $\mathcal{N}(0,1)$, the second has expression values of $\mathcal{N}(1,1)$, and the third class has expression values of $\mathcal{N}(-1,1)$. (B) The original connectivity matrix (upper panel) and perturbed connectivity matrix (lower panel) for $k=2$. (C) The connectivity matrices for $k=5$. (D) The connectivity matrices for $k=3$. The perturbed connectivity matrices clearly reveal the true structure of the data.

0 to 1 at the origin, and an area under the curve (AUC) of 1 [44, 45]. We choose the partitioning with the highest AUC and then partition the genes into subgroups that are strongly connected in those perturbation scenarios. We note that the idea of determining subspaces can be realized for both genes and cells simultaneously. We do not focus on such simultaneous clustering in this manuscript, but it is of great interest.

5.2.3 Subspace regression (Module 3)

In the first module, we divide the genes into two sets: i) a set I in which all of the genes are likely to be affected by dropouts (imputable set), and ii) a set T that have

accurate gene expression that does not need to impute (training set). In the second module, we segregate T into smaller groups of genes (gene subspaces) that share similar expression patterns. In this third module, we will impute dropout values in group I using a generalized linear regression model on gene subspaces.

Given a gene in the imputable set $g \in I$, we calculate the Euclidean distance between the gene to the centroid of each gene subspaces. Based on the calculated distances, we assign the gene to the closest subspace (with the smallest Euclidean distance). In order to impute dropout values in g, we train a generalized linear model using only highly-correlated genes within the assigned subspace in T. The linear regression process consists of two steps. The first step is to select genes from the training set that are highly correlated with the gene we need to impute. In the second step, we train the linear model using these highly correlated genes and then estimate the missing values [158].

Denoting $y \subset g$ as the non-zero part of g, S as the gene subspace in T that g was assigned to, $\{s_i \in S\}$ are expression vectors of genes in S; and $\{x_i \subset t_i\}$ are the parts of $\{t_i\}$ that correspond with y. We calculate the Pearson correlation between y and x_i and then select the 10 genes $\{t_1, \ldots, t_{10}\}$ in T with the highest correlation coefficients. We train a linear model in which $\{x_1, \ldots, x_{10}\}$ are the predictor variables and y is the outcome variable. In our implementation, we adopt the *lm* function that is available in the *stats* R package. Next, we use the trained linear model to estimate the missing values in $g \setminus y$, using $\{t_1 \backslash x_1, \ldots, t_{10} \backslash x_{10}\}$ as the predictors, where $t_i \backslash x_i$ is the part of t_i that does not belong to x_i. To avoid adding excessive weight to genes with high expression values, we always rescale the data to an acceptable range (default is [0,100]) using log transformation (base 2).

5.3 Validation and Analysis Results

To assess the performance of scISR, we use both real scRNA-seq data and simulation. We compare scISR with five popular methods, MAGIC [104], scImpute [105], SAVER [106], scScope [154], and scGNN [159]. SAVER and scImpute are statistical approaches that impute the missing values using mixture models; MAGIC is a mathematical approach that relies on Markov transition to estimate the missing values. scScope uses a recurrent network layer to iteratively perform imputations on zero-valued entries of input scRNA-seq data. scGNN formulates and aggregates cell–cell relationships with graph neural networks and models heterogeneous gene expression patterns using a left-truncated mixture Gaussian model. scGNN uses the cell-cell relationships to impute the dropouts.

First, we apply the six methods on 25 real scRNA-seq datasets with known cell types. Table 5.1 shows the details of 25 single-cell datasets that will be used in our validation. The cell labels are only used *a posteriori* to assess whether the imputation enhances the cell segregation, i.e., making the cell types more separable without drastically altering the transcriptome landscape. Second, we simulate 46 single-cell expression datasets whose values follow different distributions and dropout rates. We then apply the six imputation methods, scISR, MAGIC, scImpute, SAVER, scScope, and scGNN on the masked dataset to recover the missing values. Since we know exactly the missing entries and values, we can accurately assess the reliability of each method in terms of both sensitivity and specificity.

Table 5.1: Description of the 25 single-cell datasets used to assess the performance of imputation methods. The first three columns describe the name, accession ID, and tissue, while the following seven columns show the sequencing protocol, cell isolation technique, quantification scheme, normalized unit, dropout rate, number of cell types, and number of cells.

Dataset	Accession ID	Tissue	Sequencing Protocol	Cell Isolation	Quant. Scheme	Norm. Unit	Drop. Rate	Class	Size
1. Fan [160]	GSE53386	Mouse Embryo	SUPeR-seq	Plate	Reads	FPKM	0.584	6	69
2. Treutlein [161]	GSE52583	Mouse Tissues	SMARTer	Plate	Reads	FPKM	0.902	5	80
3. Yan [118]	GSE36552	Human Embryo	Tang	Plate	Reads	RPKM	0.456	6	90
4. Goolam [119]	E-MTAB-3321	Mouse Embryo	Smart-Seq2	Plate	Reads	CPM	0.685	5	124
5. Deng [120]	GSE45719	Mouse Embryo	Smart-Seq	Plate	Reads	RPKM	0.605	6	268
6. Pollen [121]	SRP041736	Human Tissues	SMARTer	Plate	Reads	TPM	0.671	4	301
7. Darmanis [123]	GSE67835	Human Brain	SMARTer	Plate	Reads	CPM	0.808	9	466
8. Usoskin [125]	GSE59739	Mouse Brain	STRT-Seq	Plate	Reads	RPM	0.846	3	622
9. Camp [124]	GSE75140	Human Brain	SMARTer	Plate	Reads	FPKM	0.801	7	734
10. Klein [132]	GSE65525	Mouse Embryo	inDrop	Droplet	UMI	RPM	0.658	4	2,717
11. Romanov [133]	GSE74672	Human Brain	SMARTer	Plate	UMI	-	0.878	7	2,881
12. Segerstolpe [131]	E-MTAB-5061	Human Pancreas	Smart-Seq2	Plate	Reads	RPKM	0.823	15	3,514
13. Manno [162]	GSE76381	Human Brain	STRT-Seq	Plate	UMI	-	0.86	56	4,029
14. Marques [163]	GSE75330	Mouse Brain	Fluidigm C1	Plate	Reads	FPKM	0.891	13	5,053
15. Baron [129]	GSE84133	Human Pancreas	inDrop	Droplet	UMI	TPM	0.906	14	8,569
16. Sanderson [138]	SCP916	Mouse Tissues	10X Genomics	Droplet	Reads	-	0.764	11	12,648
17. Slyper	SCP345	Human Blood	10X Genomics	Droplet	UMI	-	0.956	8	13,316
18. Zilionis (Mouse) [140]	GSE127465	Mouse Lung	inDrop	Droplet	UMI	RPM	0.976	7	15,939
19. Tasic [164]	GSE115746	Mouse Visual Cortex	SMART-Seq	Plate	Reads	CPM	0.798	6	23,178
20. Zyl (Human) [165]	SCP780	Human Eye	inDrop	Droplet	UMI	-	0.913	19	24,023
21. Zilionis (Human) [140]	GSE127465	Human Lung	inDrop	Droplet	UMI	RPM	0.982	9	34,558
22. Wei [166]	SCP469	Human Synovium	10x Genomics	Droplet	UMI	TPM	0.915	9	41,565
23. Cao [167]	SCP454	Sea Squirt Embryos	10x Genomics	Droplet	UMI	-	0.821	7	90,579
24. Orozco [145]	GSE135133	Human Eye	10X Genomics	Droplet	UMI	RPKM	0.964	12	100,055
25. Darrah [146]	GSE139598	Human Blood	Drop-seq	Droplet	UMI	-	0.947	14	162,490

[1] UMI: Unique Molecular Identifier; CPM: Counts Per Million; RPM: Reads Per Million; RPKM: Reads Per Kilobase of transcript, per Million mapped reads; FPKM: Fragments Per Kilobase of transcript, per Million mapped reads.

5.3.1 Cluster analysis of 25 scRNA-seq datasets using k-means

We use the known cell types of the 25 scRNA-seq datasets to assess whether the imputation helps separate cells of different types in cluster analysis. We compare scISR against MAGIC, scImpute, SAVER, scScope, and scGNN using three assessment metrics: Adjusted Rand Index (ARI) [148], Jaccard Index (JI) [168], and Purity Index (PI) [169].

Given a dataset (raw data), we use k-means to cluster the cells using the true number of cell types k as the number of clusters. We calculate the Adjusted Rand Index (ARI) [148] to compare k-means partitioning against the known cell labels. Rand Index (RI) measures the agreement between a given clustering and the ground truth. The ARI is the corrected-for-chance version of the RI. The ARI takes values from -1 to 1, with the ARI expected to be 1 for a perfect agreement, and 0 for random partitionings. Next, we apply each of the six imputation methods to the raw data to obtain the imputed data. Again, we use k-means to partition the imputed data and calculate the ARI values using the true cell labels. We expect that by imputing the raw data, we obtain better data in which the cells of different types are more separable. Therefore, we assess the performance of each method by comparing the ARI of the imputed data against the ARI obtained from the raw data. We repeat the whole procedure for all 25 datasets to assess how well each imputation method performs.

Table 5.2 and Figure 5.3 show the ARI values obtained for the 25 datasets. For each row, a cell of a method is highlighted in green if the imputed ARI is higher than the raw ARI. The maximum memory permitted for each analysis was set to 100 GB of RAM. scISR and MAGIC are the only methods able to analyze all datasets. scImpute runs out of memory when analyzing datasets with 23,178 cells (Tasic) or larger.

SAVER crashes when analyzing the Tasic dataset, and it runs out of memory when analyzing datasets with 90,579 cells (Cao) or larger. scScope runs out of memory when analyze the biggest dataset (Darrah). scGNN ran out of memory when analyzing the datasets Cao, Orozco, and Darrah. For 25 real datasets, scISR is able to improve the ARI values 21 out of 25. The average ARI value of scISR is 0.571, which is the highest compared to those of raw data and data imputed by MAGIC, scImpute, SAVER, scScope, and scGNN (0.504, 0.461, 0.286, 0.423, 0.165, and 0.279, respectively). Overall, scISR increases the ARI values by 13.3% across all datasets. For the two datasets Zyl (Human) (24,023 cells) and Zilionis (Human) (34,558 cells), scISR increases the ARI values significantly (11.3% and 14.5%, respectively). For Orozco and Darrah datasets with more than 100,000 cells, scISR increases the ARI values by 13.6% and 77.2%, respectively. A one-sided Wilcoxon test also confirms that the ARI values of scISR are significantly higher than those of raw data ($p = 3.2 \times 10^{-5}$) and of other imputation methods ($p = 9.8 \times 10^{-6}$).

To perform a more comprehensive analysis, we also compare the methods using two other metrics: Jaccard Index (JI) [168] and Purity Index (PI) [169]. The detailed results for each dataset and method are reported in Table 5.2–5.4. Overall, scISR is the only method that has better clustering accuracy on average when comparing with using the raw data. The results are similar for analyses using JI and PI. Among all methods, scISR has the highest average JI values (Table 5.3). Its average JI value is 0.531, compare to 0.468, 0.453, 0.276, 0.403, 0.243 and 0.273 of the raw data, MAGIC's, scImpute's, SAVER's, scScope's, and scGNN's. A one-sided Wilcoxon test also confirms that the JI values of scISR are significantly higher than those of raw data ($p = 3.2 \times 10^{-5}$) and of all other methods ($p = 4.8 \times 10^{-5}$). Table 5.4 shows the PI values obtained from raw and imputed data. It is the only method that has the average PI value higher than that of the raw data. All other methods have an

average PI less than that of the raw data. scISR improves cluster analysis by having PI values higher than those of the raw data in 15 out of 25 datasets. A one-sided Wilcoxon test also confirms that the PI values of scISR are significantly higher than those of raw data ($p = 0.007$) and of all other methods ($p = 9.9 \times 10^{-5}$).

Next, to assess the performance of each method with respect to different cell isolation techniques, quantitative schemes, and normalized units, we divide the datasets into multiple overlapping groups: (1) 14 plate-based and 11 droplet-based datasets; (2) 12 with UMI and 13 with read count; and (3) 7 without normalization, 11 with transcript length-normalization (RPKM/FPKM/TPM), and 7 with transcript-depth normalization (CPM/RPM). Figure 5.3 shows the ARI values obtained for raw data and data imputed by four imputation methods. The ARI values of scISR are consistently higher than those of raw data and of other methods in each grouping. Interestingly, the ARI values of raw data are comparable across quantification schemes (UMI/read) but differ greatly across different normalization units (Figure 5.4A). Well-known normalization techniques developed for bulk RNA-seq (RPKM/FPKM/TPM) improve raw data's cluster analysis (better than no normalization), but they have apparent disadvantages compared to CPM/RPM. The ARI values of scISR follow the same trend but are always higher than those of raw data. Similarly, Figures 5.4B and Figure 5.4C show the JI and PI values obtained for the cluster analysis. Regardless of the assessment metrics, cluster analysis in conjunction with scISR has a notable advantage over other imputation methods.

To understand the impact of data scaling on the performance of the imputation methods, we also perform the same analysis without log transformation applied to the input data. We repeat the same process as the previous analysis, the only difference is we do not perform log transformation on the raw data before applying imputation method on it. The clustering results are also assessed using three different metrics

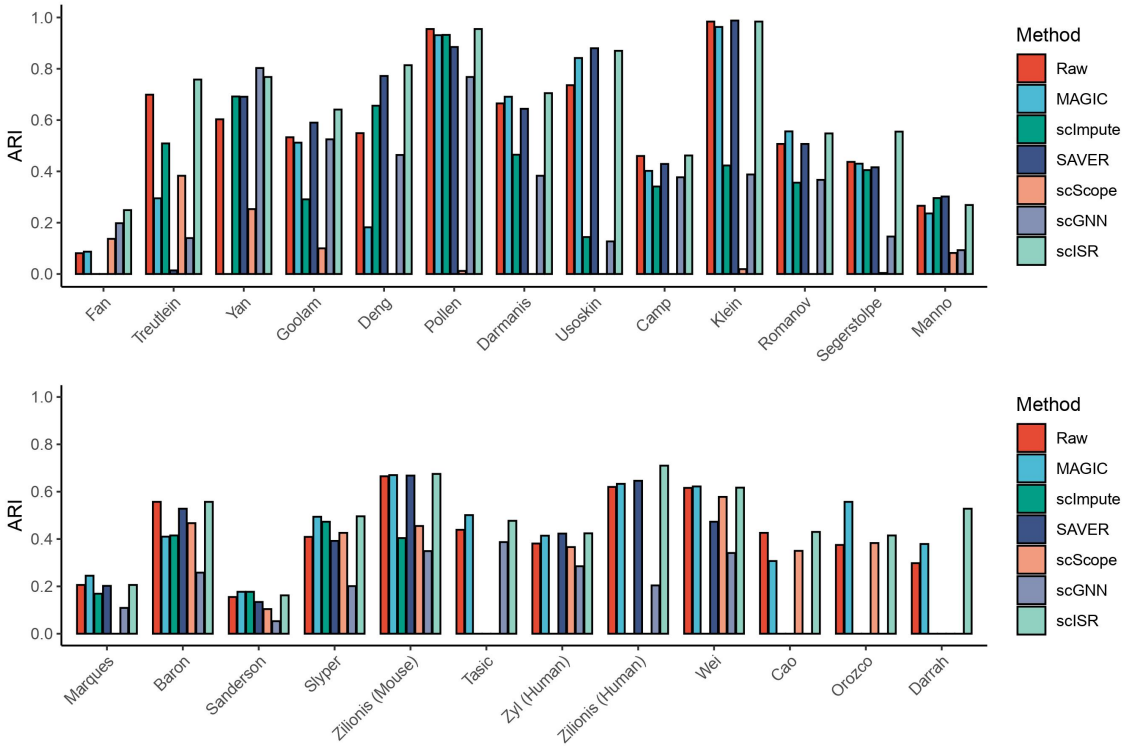

Figure 5.3: Adjusted Rand Index (ARI) obtained from raw and imputed data. The x-axis shows the names of the datasets while the y-axis shows ARI value of each method. scISR improves cluster analysis by having ARI values higher than those of the raw data in 21 out of 25 datasets.

Table 5.2: Adjusted Rand Index (ARI) obtained from raw and imputed data. In each row, a cell is highlighted in green if the ARI value is higher than that of the raw data. scISR improves cluster analysis by having ARI values higher than those of the raw data in 21 out of 25 datasets. A one-sided Wilcoxon test also confirms that the ARI values of scISR are significantly higher than those of raw data ($p = 3.2 \times 10^{-5}$) and of all other methods ($p = 9.8 \times 10^{-6}$).

Dataset	Size	Raw	MAGIC	scImpute	SAVER	scScope	scGNN	scISR
Fan	69	0.081	0.087	0.000	0.000	0.137	0.198	0.249
Treutlein	80	0.699	0.295	0.509	0.014	0.383	0.140	0.758
Yan	90	0.603	0.000	0.692	0.691	0.253	0.803	0.768
Goolam	124	0.533	0.512	0.291	0.590	0.1	0.525	0.641
Deng	268	0.549	0.182	0.656	0.772	0	0.464	0.814
Pollen	301	0.955	0.931	0.932	0.885	0.012	0.768	0.955
Darmanis	466	0.665	0.691	0.465	0.644	0	0.383	0.705
Usoskin	622	0.736	0.842	0.144	0.880	0	0.127	0.870
Camp	734	0.460	0.402	0.341	0.429	0	0.377	0.462
Klein	2,717	0.984	0.963	0.423	0.988	0.019	0.388	0.984
Romanov	2,881	0.507	0.556	0.356	0.507	0	0.367	0.548
Segerstolpe	3,514	0.437	0.430	0.405	0.576	0.004	0.146	0.555
Manno	4,029	0.266	0.236	0.296	0.302	0.082	0.093	0.269
Marques	5,053	0.206	0.245	0.169	0.202	0	0.109	0.206
Baron	8,569	0.557	0.410	0.415	0.528	0.467	0.258	0.557
Sanderson	12,648	0.155	0.177	0.177	0.134	0.104	0.053	0.162
Slyper	13,316	0.409	0.494	0.473	0.392	0.426	0.201	0.496
Zilionis (Mouse)	15,939	0.665	0.670	0.404	0.668	0.455	0.349	0.675
Tasic	23,178	0.439	0.501	N/A	N/A	0	0.387	0.477
Zyl (Human)	24,023	0.381	0.414	N/A	0.423	0.366	0.285	0.424
Zilionis (Human)	34,558	0.620	0.633	N/A	0.646	0	0.204	0.710
Wei	41,565	0.616	0.622	N/A	0.473	0.578	0.341	0.617
Cao	90,579	0.426	0.307	N/A	N/A	0.35	N/A	0.430
Orozco	100,055	0.375	0.557	N/A	N/A	0.383	N/A	0.415
Darrah	162,490	0.298	0.379	N/A	N/A	N/A	N/A	0.528
Mean ARI		0.504	0.461	0.286	0.423	0.165	0.279	0.571

[1] N/A: Out of memory or error.

Table 5.3: Jaccard Index (JI) obtained from raw and imputed data. In each row, a cell value is highlighted in green if the JI value is higher than that of the raw data. scISR improves cluster analysis by having JI values higher than those of the raw data in 21 out of 25 datasets. A Wilcoxon test also confirms that the JI values of scISR are significantly higher than those of raw data ($p = 3.2 \times 10^{-5}$) and of all other methods ($p = 4.8 \times 10^{-5}$).

Dataset	Size	Raw	MAGIC	scImpute	SAVER	scScope	scGNN	scISR
Fan	69	0.195	0.223	0.156	0.177	0.172	0.226	0.261
Treutlein	80	0.673	0.433	0.482	0.316	0.377	0.296	0.727
Yan	90	0.524	0.194	0.612	0.608	0.245	0.734	0.695
Goolam	124	0.513	0.496	0.359	0.607	0.195	0.506	0.643
Deng	268	0.524	0.333	0.629	0.739	0.293	0.446	0.780
Pollen	301	0.923	0.885	0.886	0.816	0.112	0.656	0.924
Darmanis	466	0.563	0.594	0.379	0.541	0.169	0.319	0.606
Usoskin	622	0.679	0.795	0.264	0.840	0.273	0.249	0.828
Camp	734	0.395	0.368	0.306	0.390	0.211	0.359	0.398
Klein	2,717	0.977	0.948	0.430	0.983	0.275	0.386	0.977
Romanov	2,881	0.451	0.505	0.316	0.466	0.249	0.326	0.485
Segerstolpe	3,514	0.363	0.356	0.330	0.330	0.228	0.137	0.464
Manno	4,029	0.167	0.147	0.187	0.191	0.056	0.061	0.168
Marques	5,053	0.168	0.199	0.149	0.170	0.106	0.107	0.168
Baron	8,569	0.445	0.324	0.326	0.418	0.374	0.207	0.445
Sanderson	12,648	0.243	0.277	0.273	0.225	0.2	0.120	0.256
Slyper	13,316	0.393	0.476	0.458	0.381	0.427	0.232	0.478
Zilionis (Mouse)	15,939	0.601	0.607	0.354	0.602	0.409	0.337	0.610
Tasic	23,178	0.431	0.490	N/A	N/A	0.134	0.389	0.520
Zyl	24,023	0.287	0.315	N/A	0.324	0.281	0.215	0.323
Zilionis (Human)	34,558	0.530	0.546	N/A	0.556	0.09	0.211	0.633
Wei	41,565	0.535	0.541	N/A	0.400	0.499	0.317	0.535
Cao	90,579	0.374	0.305	N/A	N/A	0.326	N/A	0.379
Orozco	100,055	0.375	0.533	N/A	N/A	0.364	N/A	0.395
Darrah	162,490	0.369	0.446	N/A	N/A	N/A	N/A	0.589
Mean		0.468	0.453	0.276	0.403	0.243	0.273	0.531

[1] N/A: Out of memory or error.

Table 5.4: Purity Index (PI) obtained from raw and imputed data. In each row, a cell value is highlighted in green if the JI value is higher than that of the raw data. scISR improves cluster analysis by having PI values higher than those of the raw data in 15 out of 25 datasets. A Wilcoxon test also confirms that the PI values of scISR are significantly higher than those of raw data ($p = 0.007$) and of all other methods ($p = 9.9 \times 10^{-5}$).

Dataset	Size	Raw	MAGIC	scImpute	SAVER	scScope	scGNN	scISR
Fan	69	0.485	0.424	0.379	0.379	0.5	0.500	0.545
Treutlein	80	0.800	0.662	0.825	0.538	0.738	0.550	0.838
Yan	90	0.811	0.356	0.811	0.833	0.567	0.867	0.844
Goolam	124	0.823	0.815	0.758	0.774	0.597	0.863	0.823
Deng	268	0.806	0.660	0.795	0.795	0.507	0.795	0.840
Pollen	301	0.963	0.920	0.924	0.870	0.236	0.857	0.963
Darmanis	466	0.841	0.820	0.702	0.830	0.283	0.655	0.848
Usoskin	622	0.830	0.879	0.524	0.929	0.378	0.505	0.929
Camp	734	0.738	0.651	0.614	0.655	0.307	0.598	0.740
Klein	2,717	0.991	0.979	0.650	0.994	0.351	0.633	0.991
Romanov	2,881	0.845	0.800	0.760	0.817	0.35	0.732	0.861
Segerstolpe	3,514	0.840	0.822	0.773	0.869	0.377	0.666	0.847
Manno	4,029	0.506	0.463	0.509	0.467	0.266	0.282	0.506
Marques	5,053	0.445	0.479	0.446	0.440	0.19	0.345	0.445
Baron	8,569	0.947	0.856	0.833	0.935	0.888	0.703	0.947
Sanderson	12,648	0.936	0.964	0.944	0.957	0.858	0.877	0.958
Slyper	13,316	0.907	0.906	0.895	0.882	0.867	0.762	0.917
Zilionis (Mouse)	15,939	0.976	0.970	0.887	0.976	0.853	0.762	0.973
Tasic	23,178	0.912	0.907	N/A	N/A	0.485	0.874	0.856
Zyl	24,023	0.861	0.878	N/A	0.863	0.787	0.780	0.875
Zilionis (Human)	34,558	0.918	0.930	N/A	0.946	0.37	0.663	0.920
Wei	41,565	0.768	0.773	N/A	0.719	0.748	0.559	0.768
Cao	90,579	0.776	0.643	N/A	N/A	0.712	N/A	0.761
Orozco	100,055	0.918	0.966	N/A	N/A	0.911	N/A	0.928
Darrah	162,490	0.924	0.921	N/A	N/A	N/A	N/A	0.942
Mean		0.823	0.778	0.521	0.659	0.525	0.593	0.835

[1] N/A: Out of memory or error.

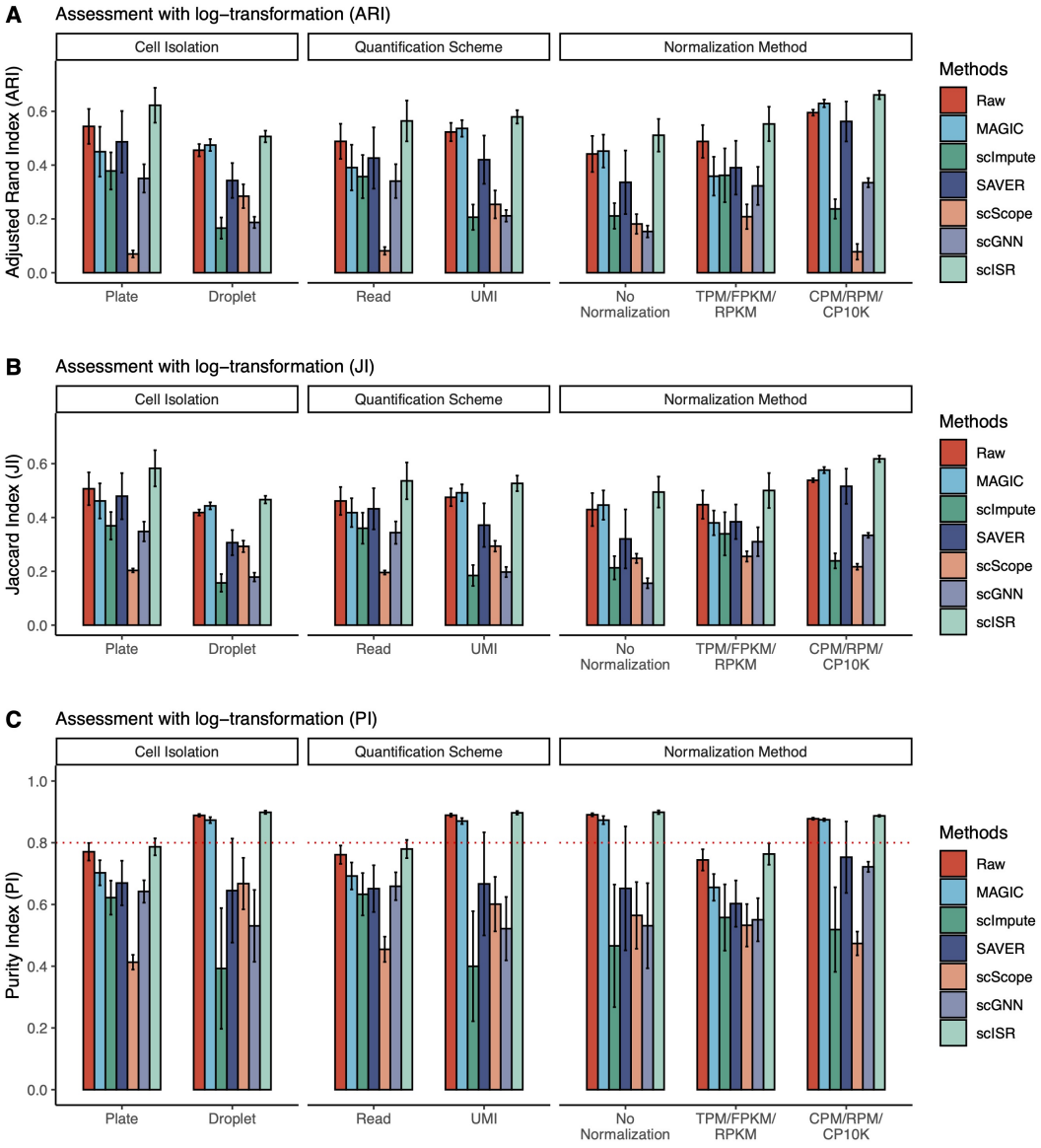

Figure 5.4: Assessment results of each imputation method with respect to cell isolation techniques, quantification schemes, or normalized units. The analysis is performed with a log transformation of the data. Panel (A) shows the results using Adjusted Rand Index (ARI), while panels (B) and (C) show the results using Jaccard Index (JI) and Purity Index (PI). scISR consistently outperforms other methods in every grouping by having the highest ARI, JI, and PI values.

Adjusted Rand Index (ARI), Jaccard Index (JI) and Purity Index (PI). With the exception of scISR, a decrease in performance is observed for all imputation methods due to the dominance of genes with large values. This leads to a wider accuracy gap between scISR and other imputation methods.

Figure 5.5A shows the ARI values obtained for data *without* log transformation. Again, the ARI values of scISR are consistently higher than those of raw data and of other methods in each grouping. Note that the ARI values of the raw data decrease (in comparison with ARI values obtained with log transformation). The reason is that the range of the data is very large. For example, the Deng dataset has a max RPKM value of 155,847 whereas the mean RPKM of the dataset is only 35. Without log transformation, genes with large values dominate the clustering analysis results, which is undesirable. A decrease in performance is observed for other imputation methods too (except scISR).

Table 5.5 shows the ARI values obtained for the raw data and the data inferred by the six imputation methods. In this analysis, scISR improves the clustering analysis in 24 out of 25 datasets by having the ARI values higher than those of the raw data. Among all methods, scISR has the highest average ARI values. Its average ARI value is 0.571, compare to 0.374, 0.356, 0.219, 0.307, 0.101 and 0.306 of the raw data, MAGIC's, scImpute's, SAVER's, scScope's, and scGNN's. A Wilcoxon test also confirms that the ARI values of scISR are significantly higher than those of raw data ($p = 6.3 \times 10^{-5}$) and of all other methods ($p = 1.9 \times 10^{-7}$).

Table 5.6 shows the JI values obtained for the raw data and the data inferred by the six imputation methods. In this analysis, scISR also improves the clustering analysis in 23 out of 25 datasets by having the JI values higher than those of the raw data. Among all methods, scISR has the highest average JI values. Its average JI value is 0.531, compare to 0.392, 0.399, 0.245, 0.308, 0.223, and 0.304 of the raw

data, MAGIC's, scImpute's, SAVER's, scScope's, and scGNN's. A Wilcoxon test also confirms that the JI values of scISR are significantly higher than those of raw data ($p = 0.0001$) and of all other methods ($p = 4.4 \times 10^{-6}$).

Table 5.7 shows the PI values obtained from raw and imputed data. The results are similar to the analysis using ARI and JI. It is the only method that has the average PI value higher than that of the raw data. All other methods have an average PI less than that of the raw data. scISR improves cluster analysis by having PI values higher than those of the raw data in most datasets (21 out of 25). A Wilcoxon test also confirms that the PI values of scISR are significantly higher than those of raw data ($p = 0.0001$) and of all other methods ($p = 2.4 \times 10^{-7}$).

5.3.1.1 Cluster analysis of 25 scRNA-seq datasets using Seurat

To further assess the performance of imputation methods, we perform an additional clustering analysis using Seurat [68]. This method can automatically determine the number of cell types from the input data. We first used Seurat to cluster the raw and imputed data of the 25 real scRNA-seq datasets. We then compared the clustering results against true cell types using Adjusted Rand Index (ARI). Figure 5.6 and Table 5.8 show the ARI values obtained from the raw data and the data obtained from the six imputation methods. scISR is able to improve the cluster analysis in 14 out of 25 datasets. MAGIC, scImpute, SAVER, scScope, and scGNN improve the cluster analysis in 5, 3, 5, 4, and 5 datasets, respectively. The mean ARI value of scISR is 0.499 which is higher than the mean ARI values of all other methods (the mean ARI values for MAGIC, scImpute, SAVER, scScope, and scGNN are 0.315, 0.283, 0.324, 0.155, and 0.186, respectively). scISR is the only method that has mean ARI higher than that of the raw data.

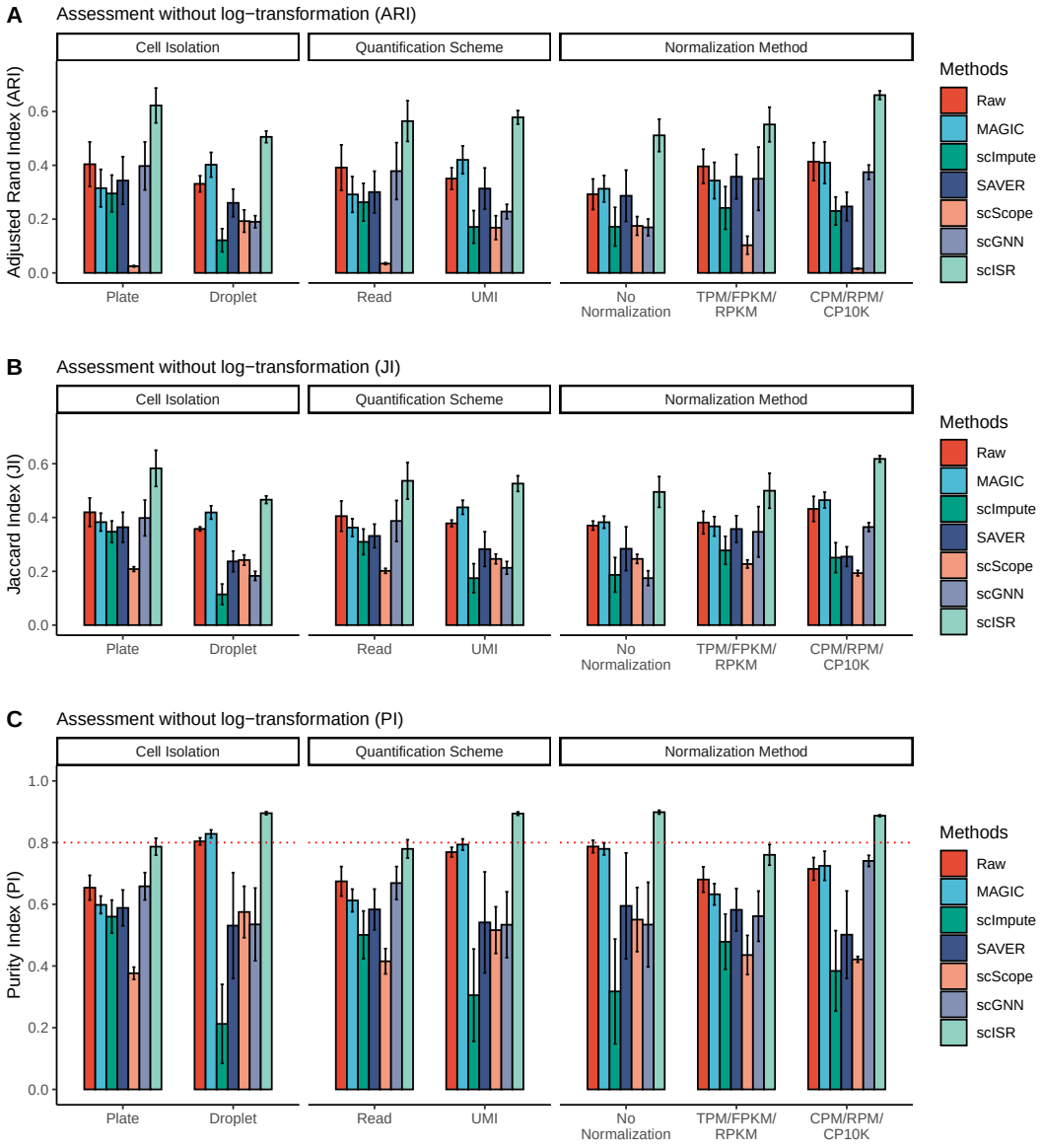

Figure 5.5: Assessment results of each imputation method with respect to cell isolation techniques, quantification schemes, or normalized units. The analysis is performed without a log transformation of the data. Panel (A) shows the results using Adjusted Rand Index (ARI) while panels (B) and (C) show the results using Jaccard Index (JI) and Purity Index (PI). scISR consistently outperforms other methods in every grouping by having the highest ARI, JI, and PI values.

Table 5.5: Adjusted Rand Index (ARI) obtained from raw and imputed data. In each row, a cell value is highlighted in green if the ARI value is higher than that of the raw data. scISR improves cluster analysis by having ARI values higher than those of the raw data in 24 out of 25 datasets. A Wilcoxon test also confirms that the ARI values of scISR are significantly higher than those of raw data ($p = 6.3 \times 10^{-5}$) and of all other methods ($p = 1.9 \times 10^{-7}$). The analysis is performed on data without log transformation.

Dataset	Size	Raw	MAGIC	scImpute	SAVER	scScope	scGNN	scISR
Fan	69	0.008	0	0	0.015	0.017	0.003	0.249
Treutlein	80	0.699	0.056	0	0	0.072	0.195	0.758
Yan	90	0.460	0.705	0.547	0.609	0.155	0.884	0.768
Goolam	124	0.629	0.17	0.281	0.379	0.112	0.657	0.641
Deng	268	0.359	0.263	0.521	0.668	0	0.865	0.814
Pollen	301	0.822	0.631	0.826	0.822	0.009	0.833	0.955
Darmanis	466	0.404	0.396	0.458	0.472	0	0.356	0.705
Usoskin	622	0.008	0.007	0.353	0.008	0.003	0.127	0.87
Camp	734	0.460	0.349	0.09	0.351	0.006	0.263	0.462
Klein	2,717	0.643	0.66	0.63	0.852	0.016	0.494	0.984
Romanov	2,881	0.193	0.29	0.519	0.45	0	0.403	0.548
Segerstolpe	3,514	0.079	0.085	0.088	0.17	0.003	0.214	0.555
Manno	4,029	0.167	0.183	0.176	0.231	0	0.107	0.269
Marques	5,053	0.100	0.179	0.181	0.231	0.001	0.124	0.206
Baron	8,569	0.276	0.271	0.331	0.471	0.008	0.284	0.557
Sanderson	12,648	0.155	0.125	N/A	0.122	0.119	0.064	0.162
Slyper	13,316	0.409	0.509	0.484	0.484	0.438	0.145	0.496
Zilionis (Mouse)	15,939	0.419	0.528	N/A	0.42	0	0.375	0.675
Tasic	24,023	0.818	0.74	N/A	N/A	0	0.442	0.477
Zyl (Human)	23,178	0.381	0.39	N/A	0.379	0.378	0.268	0.424
Zilionis (Human)	34,558	0.424	0.737	N/A	N/A	0	0.261	0.71
Wei	41,565	0.616	0.776	N/A	0.537	0.514	0.292	0.617
Cao	90,579	0.426	0.316	N/A	N/A	0.269	N/A	0.43
Orozco	100,055	0.390	0.376	N/A	N/A	0.394	N/A	0.415
Darrah	162,490	0.000	0.105	N/A	N/A	N/A	N/A	0.528
Mean ARI		0.374	0.356	0.219	0.307	0.101	0.306	0.571

[1] N/A: Out of memory or error.

Table 5.6: Jaccard Index (JI) obtained from raw and imputed data. In each row, a cell value is highlighted in green if the JI value is higher than that of the raw data. scISR improves cluster analysis by having JI values higher than those of the raw data in 23 out of 25 datasets. A Wilcoxon test also confirms that the JI values of scISR are significantly higher than those of raw data ($p = 0.0001$) and of all other methods ($p = 4.4 \times 10^{-6}$). The analysis is performed on data without log transformation.

Dataset	Size	Raw	MAGIC	scImpute	SAVER	scScope	scGNN	scISR
Fan	69	0.187	0.182	0.181	0.187	0.183	0.182	0.261
Treutlein	80	0.673	0.333	0.312	0.312	0.337	0.288	0.727
Yan	90	0.418	0.627	0.47	0.529	0.235	0.831	0.695
Goolam	124	0.634	0.403	0.434	0.401	0.355	0.621	0.643
Deng	268	0.387	0.406	0.544	0.649	0.278	0.834	0.78
Pollen	301	0.728	0.518	0.733	0.728	0.11	0.74	0.924
Darmanis	466	0.364	0.363	0.409	0.404	0.146	0.295	0.606
Usoskin	622	0.188	0.28	0.429	0.188	0.279	0.25	0.828
Camp	734	0.395	0.359	0.226	0.358	0.212	0.254	0.398
Klein	2,717	0.606	0.622	0.591	0.813	0.283	0.49	0.977
Romanov	2,881	0.268	0.346	0.484	0.418	0.246	0.356	0.485
Segerstolpe	3,514	0.243	0.245	0.247	0.192	0.227	0.185	0.464
Manno	4,029	0.108	0.116	0.113	0.144	0.03	0.069	0.168
Marques	5,053	0.134	0.172	0.174	0.19	0.107	0.116	0.168
Baron	8,569	0.259	0.254	0.303	0.379	0.199	0.223	0.445
Sanderson	12,648	0.243	0.219	N/A	0.22	0.217	0.133	0.256
Slyper	13,316	0.393	0.493	0.47	0.471	0.435	0.208	0.478
Zilionis (Mouse)	15,939	0.372	0.46	N/A	0.372	0.11	0.352	0.61
Tasic	24,023	0.809	0.735	N/A	N/A	0.134	0.421	0.52
Zyl	23,178	0.287	0.299	N/A	0.291	0.288	0.206	0.323
Zilionis (Human)	34,558	0.389	0.666	N/A	N/A	0.083	0.257	0.633
Wei	41,565	0.535	0.715	N/A	0.455	0.439	0.278	0.535
Cao	90,579	0.374	0.321	N/A	N/A	0.273	N/A	0.379
Orozco	100,055	0.370	0.355	N/A	N/A	0.37	N/A	0.395
Darrah	162,490	0.444	0.479	N/A	N/A	N/A	N/A	0.589
Mean		0.392	0.399	0.245	0.308	0.223	0.304	0.531

[1] N/A: Out of memory or error.

Table 5.7: Purity Index (PI) obtained from raw and imputed data. In each row, a cell value is highlighted in green if the PI value is higher than that of the raw data. scISR improves cluster analysis by having PI values higher than those of the raw data in 21 out of 25 datasets. A Wilcoxon test also confirms that the PI values of scISR are significantly higher than those of raw data ($p = 0.0001$) and of all other methods ($p = 2.4 \times 10^{-7}$). The analysis is performed on data without log transformation.

Dataset	Size	Raw	MAGIC	scImpute	SAVER	scScope	scGNN	scISR
Fan	69	0.394	0.364	0.364	0.409	0.364	0.364	0.545
Treutlein	80	0.800	0.55	0.538	0.538	0.562	0.638	0.838
Yan	90	0.767	0.8	0.822	0.8	0.544	0.911	0.844
Goolam	124	0.823	0.613	0.702	0.782	0.565	0.839	0.823
Deng	268	0.713	0.608	0.72	0.765	0.504	0.854	0.84
Pollen	301	0.870	0.761	0.87	0.87	0.233	0.884	0.963
Darmanis	466	0.674	0.624	0.697	0.721	0.296	0.659	0.848
Usoskin	622	0.376	0.383	0.595	0.376	0.378	0.518	0.929
Camp	734	0.738	0.542	0.396	0.54	0.313	0.55	0.74
Klein	2,717	0.803	0.81	0.81	0.883	0.363	0.688	0.991
Romanov	2,881	0.578	0.642	0.695	0.759	0.354	0.737	0.861
Segerstolpe	3,514	0.518	0.531	0.519	0.685	0.376	0.713	0.847
Manno	4,029	0.394	0.407	0.381	0.416	0.102	0.296	0.506
Marques	5,053	0.353	0.461	0.427	0.453	0.185	0.37	0.445
Baron	8,569	0.752	0.741	0.747	0.863	0.302	0.749	0.947
Sanderson	12,648	0.936	0.927	N/A	0.914	0.869	0.879	0.958
Slyper	13,316	0.907	0.903	0.894	0.899	0.85	0.706	0.917
Zilionis (Mouse)	15,939	0.873	0.971	N/A	0.873	0.503	0.797	0.973
Tasic	24,023	0.931	0.922	N/A	N/A	0.485	0.934	0.856
Zyl	23,178	0.861	0.854	N/A	0.784	0.8	0.754	0.875
Zilionis (Human)	34,558	0.749	0.918	N/A	N/A	0.37	0.701	0.92
Wei	41,565	0.768	0.772	N/A	0.75	0.743	0.561	0.768
Cao	90,579	0.776	0.669	N/A	N/A	0.595	0	0.761
Orozco	100,055	0.935	0.951	N/A	N/A	0.94	0	0.928
Darrah	162,490	0.710	0.764	N/A	N/A	N/A	0	0.942
Mean		0.720	0.700	0.407	0.563	0.464	0.604	0.835

[1] N/A: Out of memory or error.

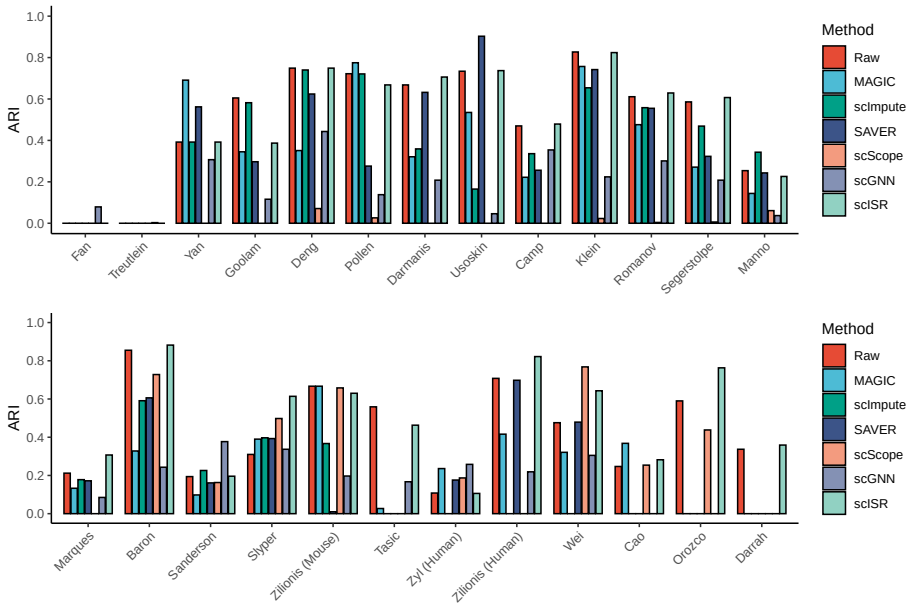

Figure 5.6: Adjusted Rand Index (ARI) obtained from raw and imputed data using Seurat as the clustering method. The x-axis shows the names of the datasets while the y-axis shows ARI value of each method.

5.3.1.2 Preservation of the transcriptome landscape

The purpose of this analysis is to assess whether the imputation alters the transcriptome landscape. Preferably, life scientists impute the data in order to improve the quality of downstream analyses. At the same time, imputation should not completely change the data because of falsely introduced signals, leading to wrong or compromised findings. In the above sections, we have demonstrated that scISR significantly improves the quality of downstream analyses (e.g., cluster analysis). In this section, we will demonstrate that scISR preserves the transcriptome landscape of the data as well. For this purpose, we will visualize the transcriptome landscape of the raw and imputed data using t-SNE [170] and UMAP [62]. We will also quantify the similarity between the imputed and original landscapes using the distance correlation index [171].

First, we use t-SNE [170] to generate the 2D transcriptome landscapes of the raw

Table 5.8: Adjusted Rand Index (ARI) obtained from raw and imputed data using Seurat as the clustering method. scISR improves cluster analysis by having ARI values higher than those of the raw data in 14 out of 25 datasets. Cells with N/A value indicate that the method failed to run due to out of memory or error.

Dataset	Size	Raw	MAGIC	scImpute	SAVER	scScope	scGNN	scISR
Fan	69	0.000	0	0	0	0	**0.079**	0
Treutlein	80	0.000	0	0	0	0	**0.003**	0
Yan	90	0.392	**0.691**	0.392	**0.562**	0	0.307	0.392
Goolam	124	0.605	0.345	0.582	0.297	0	0.116	0.387
Deng	268	0.749	0.351	0.74	0.624	0.071	0.443	0.749
Pollen	301	0.722	**0.775**	0.721	0.276	0.026	0.138	0.668
Darmanis	466	0.668	0.321	0.359	0.632	0	0.208	**0.706**
Usoskin	622	0.734	0.535	0.165	**0.903**	0	0.046	**0.737**
Camp	734	0.470	0.222	0.336	0.256	0	0.354	**0.479**
Klein	2,717	0.827	0.757	0.654	0.742	0.023	0.224	0.824
Romanov	2,881	0.611	0.476	0.558	0.555	0.004	0.301	**0.629**
Segerstolpe	3,514	0.586	0.271	0.469	0.323	0.006	0.208	**0.607**
Manno	4,029	0.254	0.144	**0.343**	0.243	0.061	0.037	0.226
Marques	5,053	0.212	0.133	0.178	0.172	0	0.085	**0.307**
Baron	8,569	0.855	0.328	0.591	0.606	0.728	0.243	**0.882**
Sanderson	12,648	0.194	0.098	**0.226**	0.161	0.163	**0.377**	**0.196**
Slyper	13,316	0.310	**0.39**	**0.397**	**0.393**	**0.498**	**0.337**	**0.614**
Zilionis (Mouse)	15,939	0.667	0.667	0.367	0.01	0.658	0.197	0.63
Tasic	23,178	0.559	0.027	N/A	N/A	0	0.167	0.463
Zyl (Human)	24,023	0.108	**0.236**	N/A	**0.176**	**0.187**	**0.258**	0.106
Zilionis (Human)	34,558	0.708	0.416	N/A	0.698	0	0.219	**0.822**
Wei	41,565	0.476	0.321	N/A	**0.479**	**0.768**	0.305	**0.643**
Cao	90,579	0.247	**0.368**	N/A	N/A	**0.254**	N/A	**0.282**
Orozco	100,055	0.590	0	N/A	N/A	0.438	N/A	**0.763**
Darrah	162,490	0.337	0	N/A	N/A	N/A	N/A	**0.359**
Mean ARI		0.475	0.315	0.283	0.324	0.155	0.186	**0.499**

and imputed data. The 2D visualizations of the 25 datasets are shown in Figures 5.7–5.11. Overall, MAGIC, SAVER, and scISR produce landscapes that are similar to those of the raw data for every single dataset analyzed. The same cannot be said about scImpute, scScope, and scGNN. For the Manno dataset (second last row in Figure 5.9), scImpute, scScope, and scGNN completely alter the landscape. scImpute tends to split cells into smaller groups while scScope and scGNN mix cells from different cell types together. This can be clearly observed in datasets such as Camp,

Segerstolpe, Manno (Human).

To perform a more comprehensive analysis, we also generate the 2D transcriptome landscapes of the 25 datasets using UMAP [62]. The visualizations are shown in Figures 5.12–5.16. Again, except for scImpute, scScope, and scGNN, other methods preserve the landscape very well. For scImpute, scScope, and scGNN, the difference between the original and imputed landscape becomes more obvious in UMAP visualization.

To quantify the similarity between the imputed and original landscapes, we calculate the distance correlation index ($dCor$) [171] for each imputed landscape generated by t-SNE and UMAP. Given X and Y as the 2D representation of the raw and imputed data, $dCor$ is calculated as $dCor = \frac{dCov(X,Y)}{\sqrt{dVar(X)dVar(Y)}}$ where $dCov(X,Y)$ is the distance covariance between X and Y while $dVar(X)$ and $dVar(Y)$ are distance variances of X and Y. Specifically, the method first calculates the pair-wise distances for X by computing the distance between each pair of cells, resulting in a square matrix. Second, it calculates the pair-wise distances for Y. Finally, it compares the two matrices using the formula described above to obtain the distance correlation. The $dCor$ coefficient takes a value between 0 and 1, with the $dCor$ is expected to be 1 for a perfect similarity. In our analysis, when we rotate the transcriptome landscape, $dCor$ does not change. In contrast to Pearson correlation, this metric measures both the linear and nonlinear associations between X and Y [171].

The $dCor$ values are displayed in each panel in Figures 5.7–5.11. We also plot the $dCor$ distributions in Figure 5.17. In this figure, the left panel shows the values obtained from t-SNE while the right panel shows the values obtained from UMAP representations. The mean correlations using t-SNE for MAGIC, scImpute, SAVER, scScope, scGNN, and scISR are 0.78, 0.46, 0.68, 0.36, 0.48, and 0.88, respectively. The bar plot shows that scISR has the highest mean correlation, as well as the small-

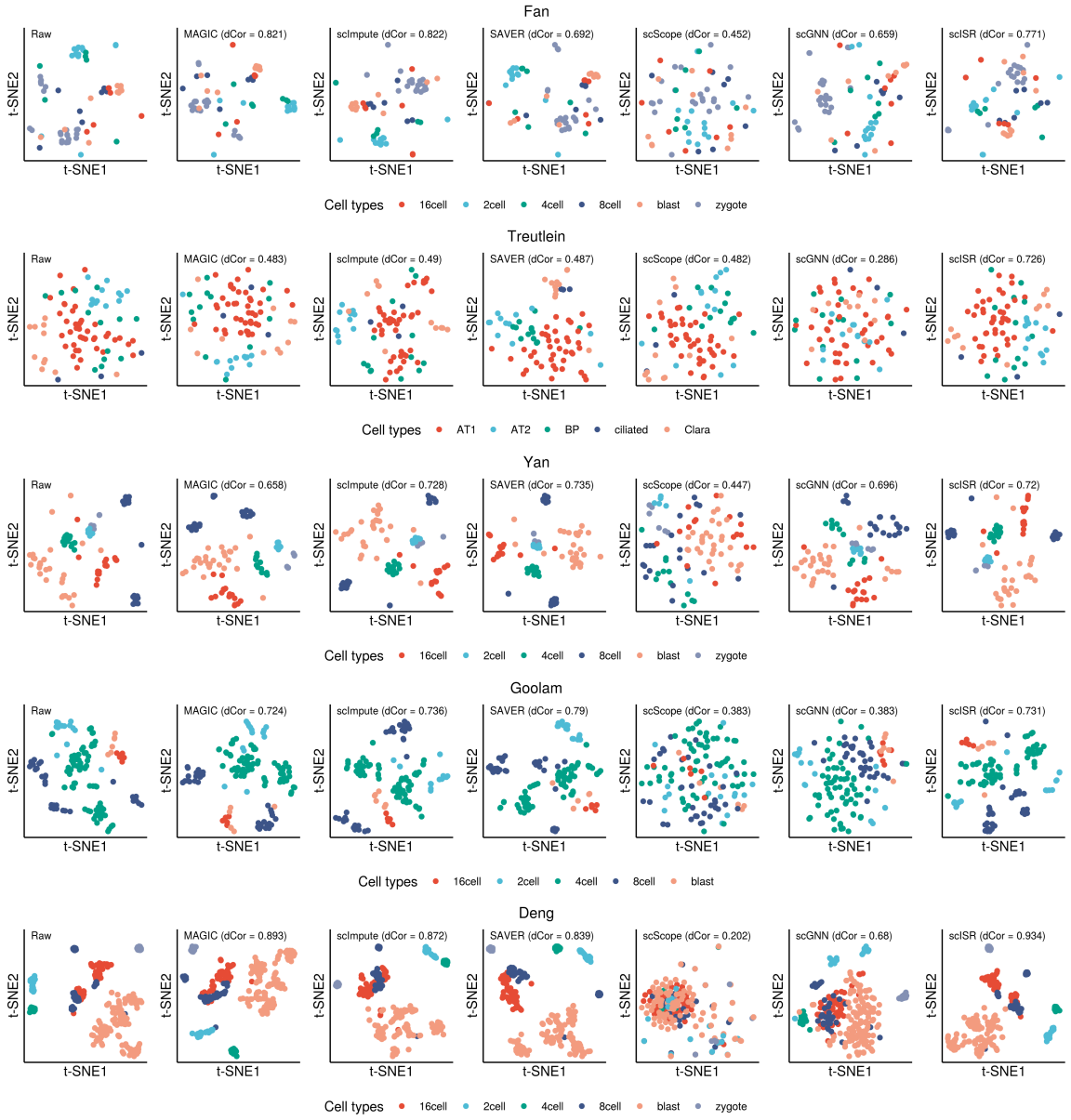

Figure 5.7: Transcriptome landscape of the Fan, Treutlein, Yan, Goolam and Deng datasets (top to bottom) using t-SNE. Different colors code for different cell types. The distance correlation calculated for each imputed dataset shows the similarity between the new landscape (from imputed data) and the original landscape (from raw data).

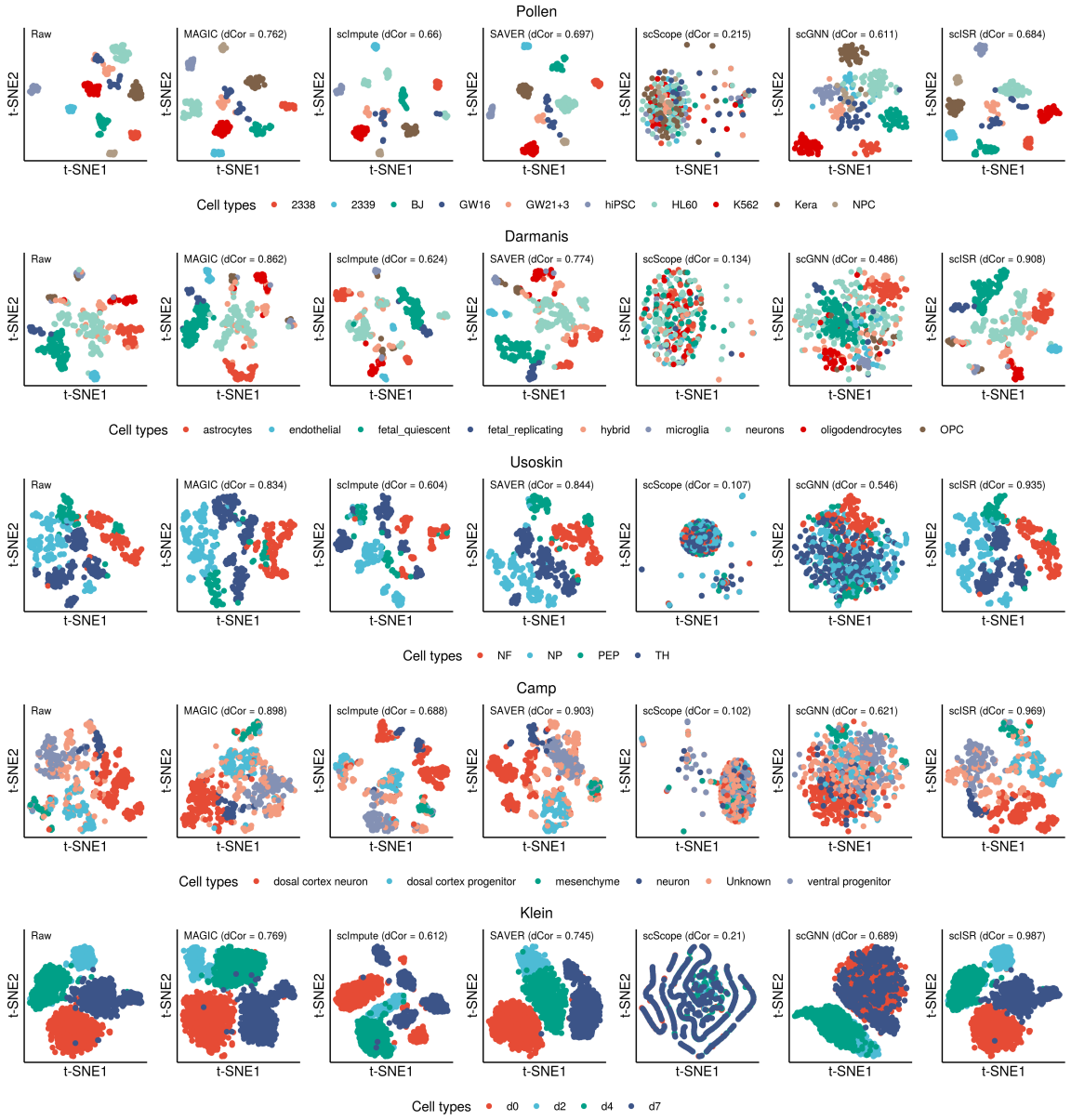

Figure 5.8: Transcriptome landscape for the Pollen, Darmanis, Usoskin, Camp and Klein datasets (top to bottom) using t-SNE. Different colors code for different cell types. The distance correlation calculated for each imputed dataset shows the similarity between the new landscape (from imputed data) and the original landscape (from raw data).

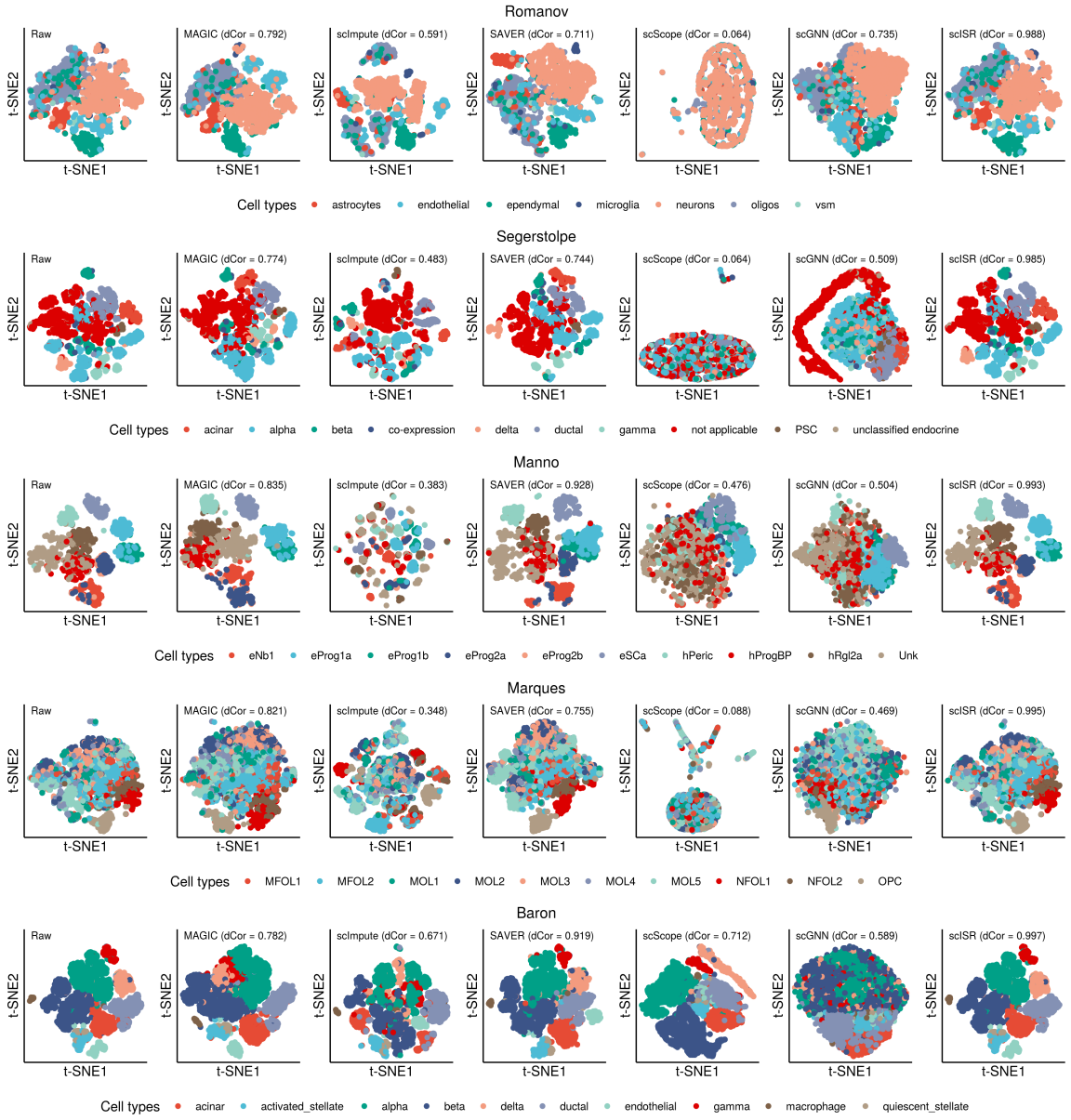

Figure 5.9: Transcriptome landscape for the Romanov, Segerstolpe, Manno (Human), Marques and Barron (Human) datasets (top to bottom) using t-SNE. Different colors code for different cell types. The distance correlation calculated for each imputed dataset shows the similarity between the new landscape (from imputed data) and the original landscape (from raw data).

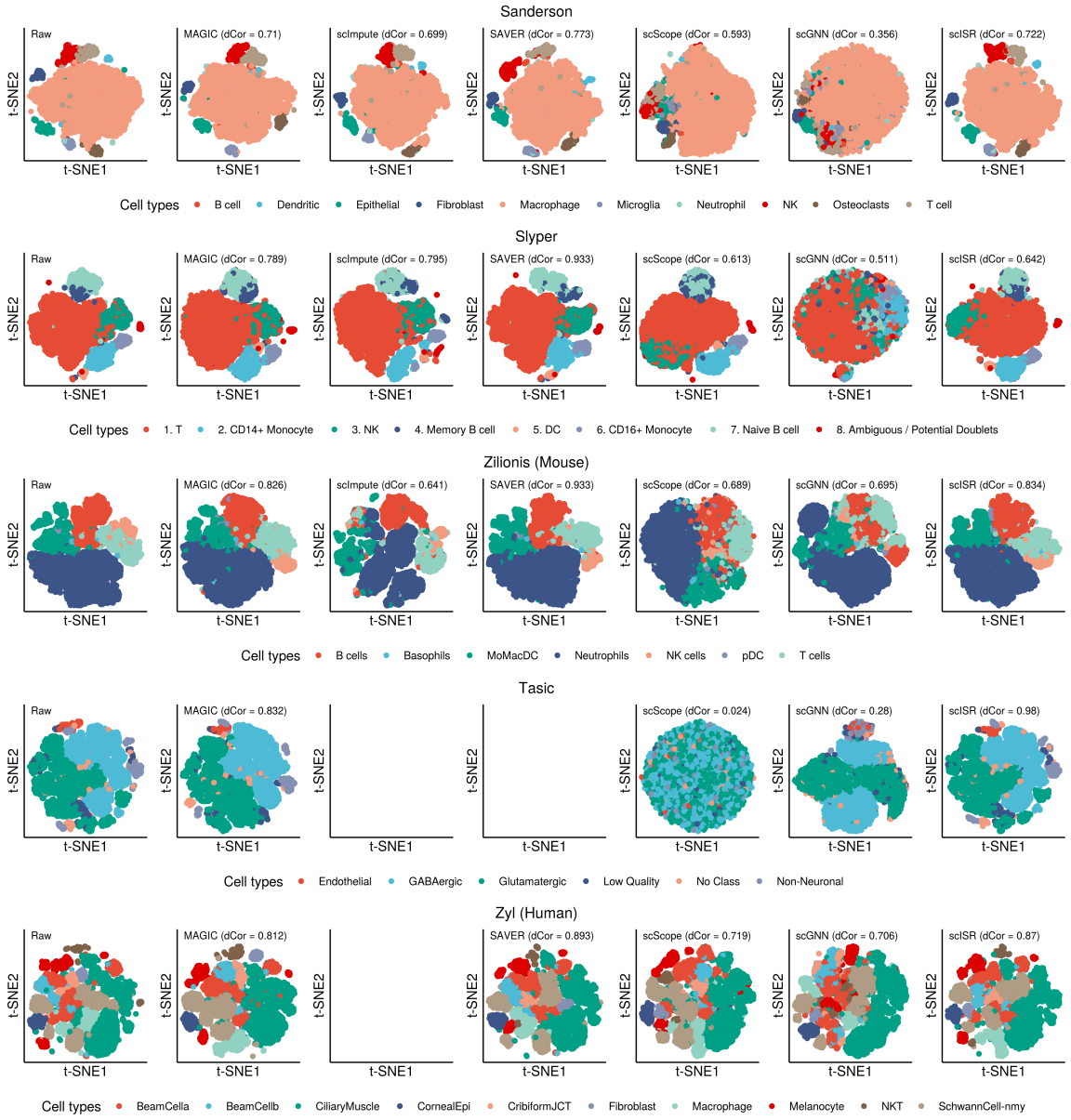

Figure 5.10: Transcriptome landscape for the Sanderson, Slyper, Zilionis (Mouse), Tasic and Zyl (Human) datasets (top to bottom) using t-SNE. Different colors code for different cell types. The distance correlation calculated for each imputed dataset shows the similarity between the new landscape (from imputed data) and the original landscape (from raw data).

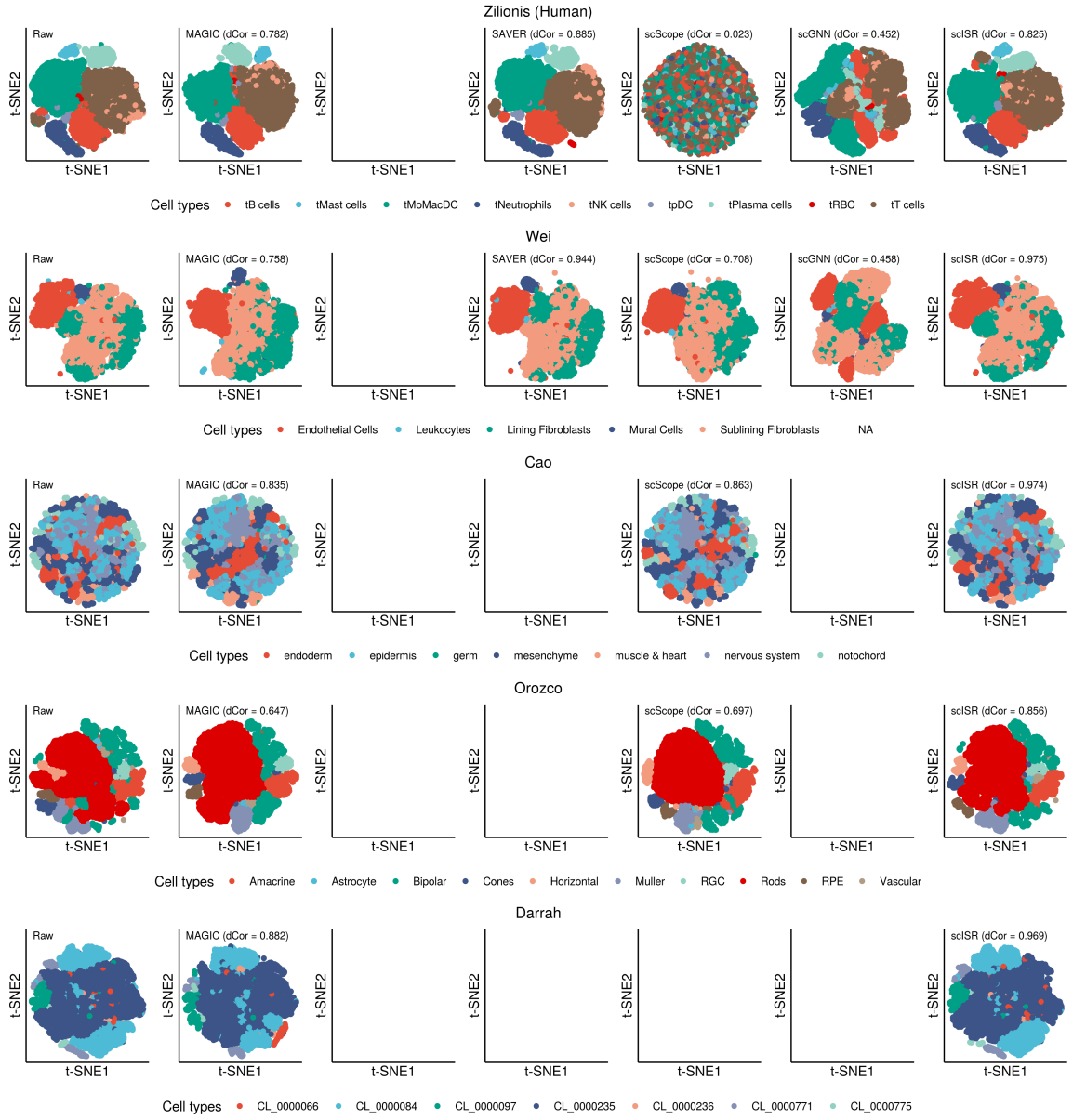

Figure 5.11: Transcriptome landscape for the Zillionis (Human), Wei (Human), Cao, Orozco and Darrah datasets (top to bottom) using t-SNE. Different colors code for different cell types. The distance correlation calculated for each imputed dataset shows the similarity between the new landscape (from imputed data) and the original landscape (from raw data).

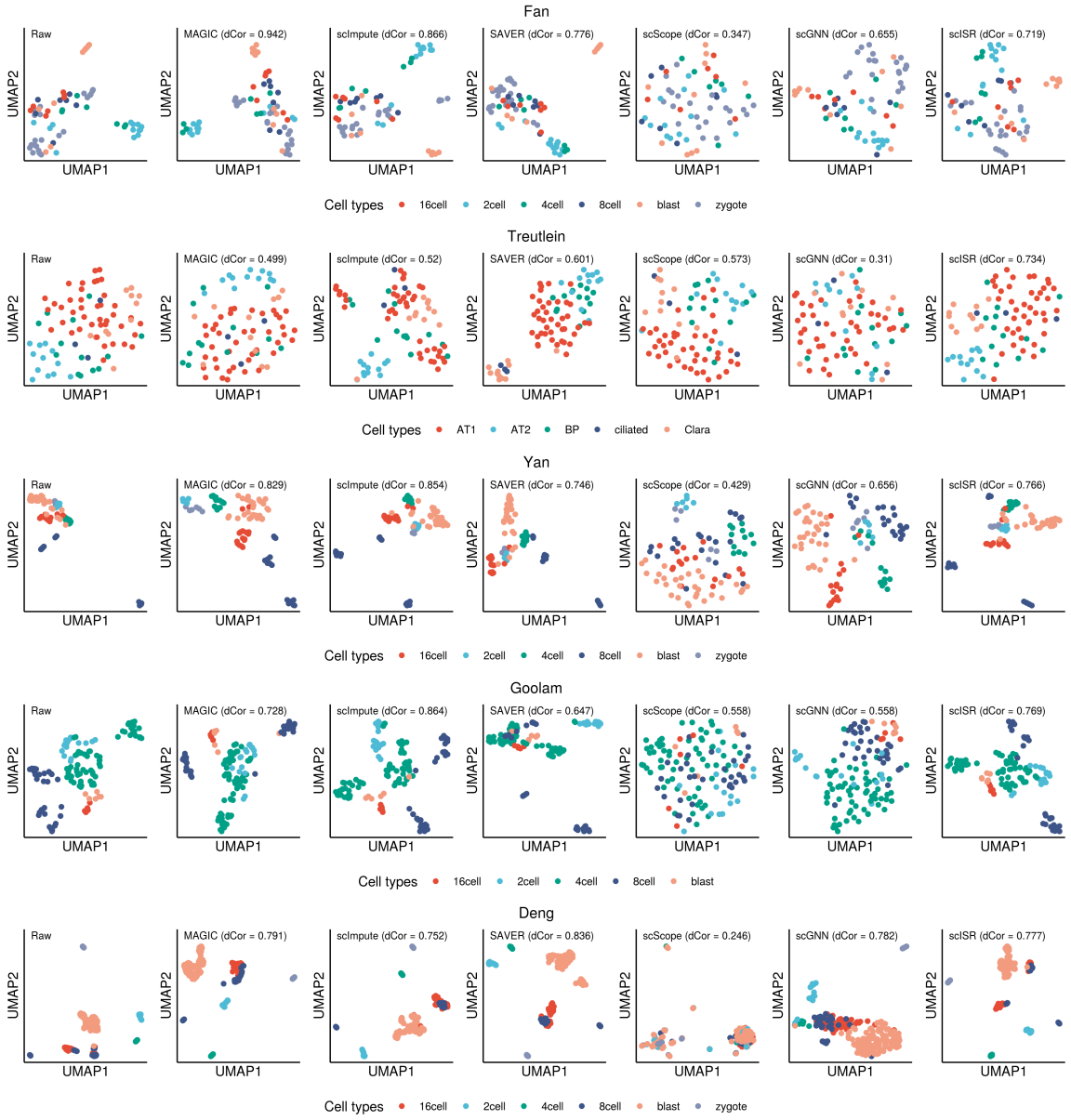

Figure 5.12: Transcriptome landscape for the Fan, Treutlein, Yan, Goolam and Deng datasets (top to bottom) using UMAP. Different colors code for different cell types. The distance correlation calculated for each imputed dataset shows the similarity between the new landscape (from imputed data) and the original landscape (from raw data).

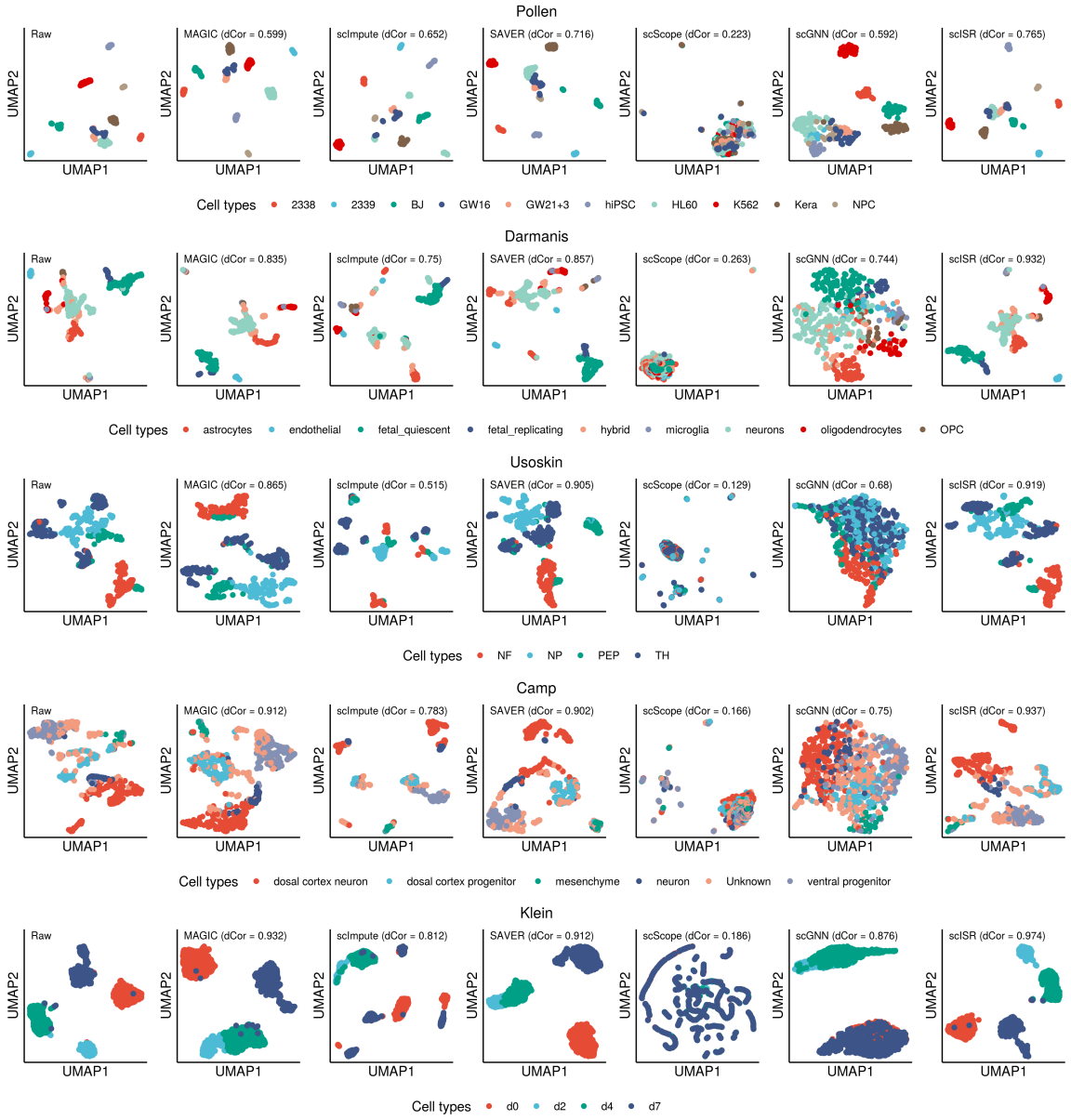

Figure 5.13: Transcriptome landscape for the Pollen, Darmanis, Usoskin, Camp and Klein datasets (top to bottom) using UMAP. Different colors code for different cell types. The distance correlation calculated for each imputed dataset shows the similarity between the new landscape (from imputed data) and the original landscape (from raw data).

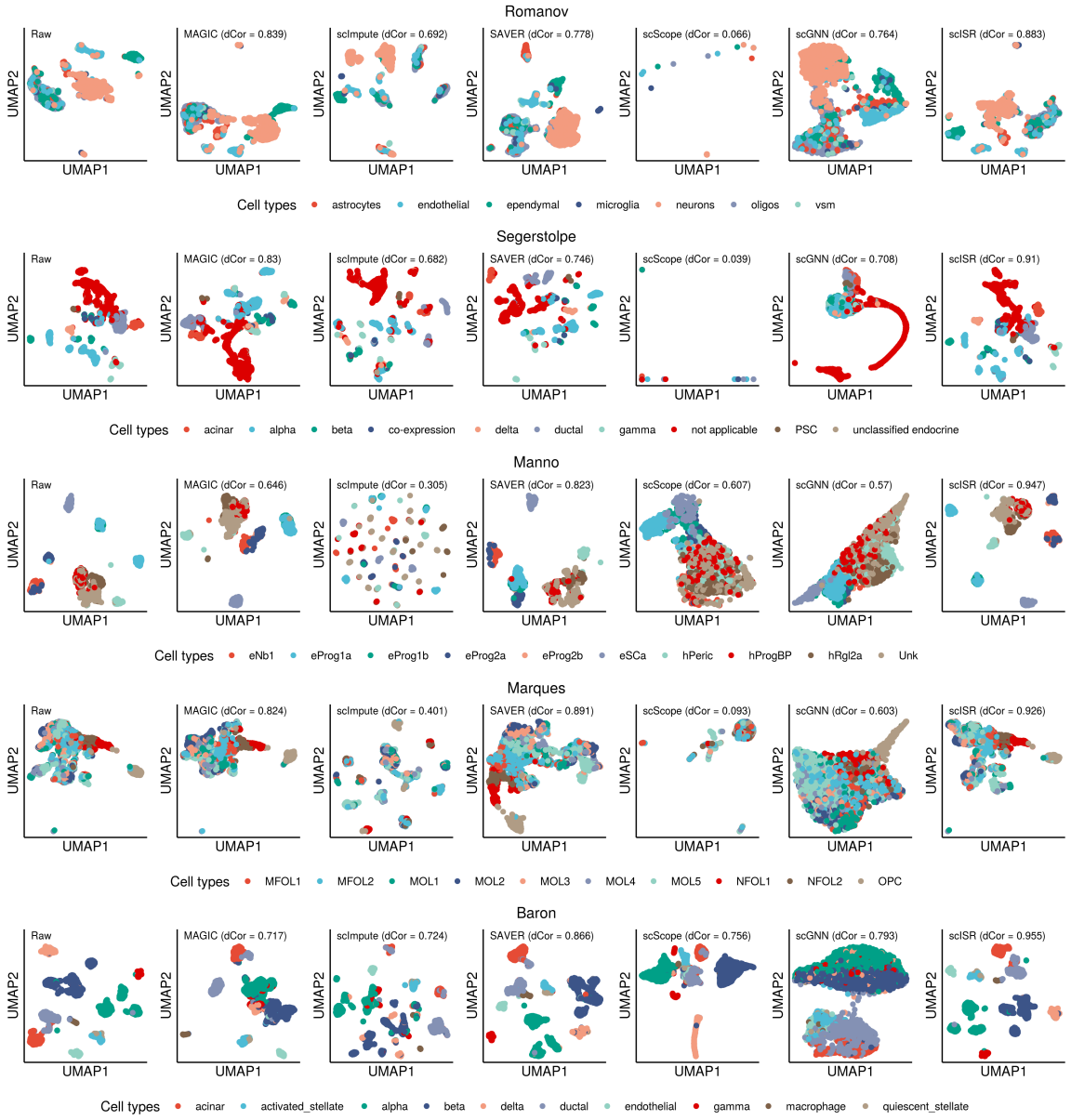

Figure 5.14: Transcriptome landscape for the Romanov, Segerstolpe, Manno (Human), Marques and Barron (Human) datasets (top to bottom) using UMAP. Different colors code for different cell types. The distance correlation calculated for each imputed dataset shows the similarity between the new landscape (from imputed data) and the original landscape (from raw data).

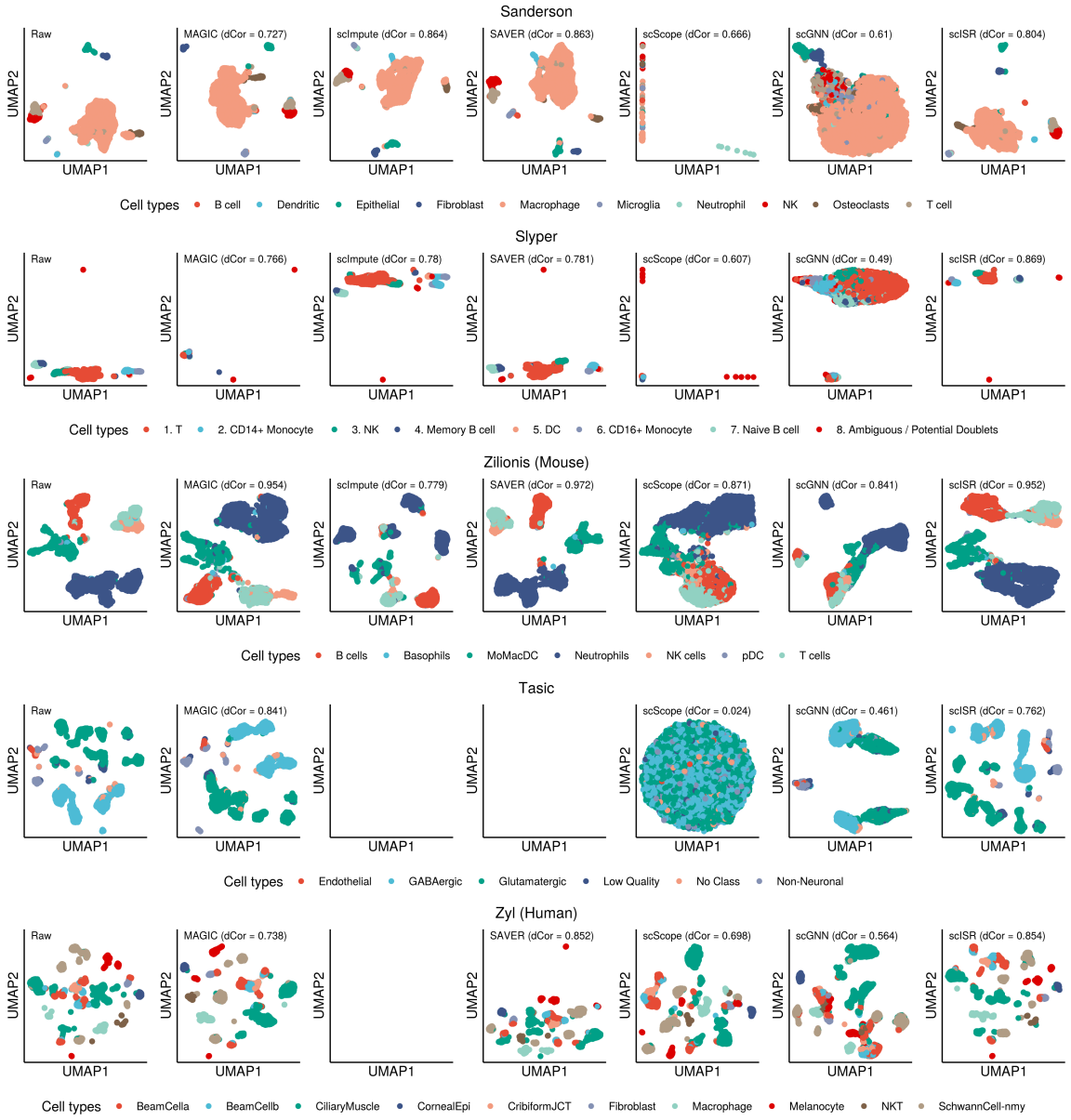

Figure 5.15: Transcriptome landscape for the Sanderson, Slyper, Zilionis (Mouse), Tasic and Zyl (Human) datasets (top to bottom) using UMAP. Different colors code for different cell types. The distance correlation calculated for each imputed dataset shows the similarity between the new landscape (from imputed data) and the original landscape (from raw data).

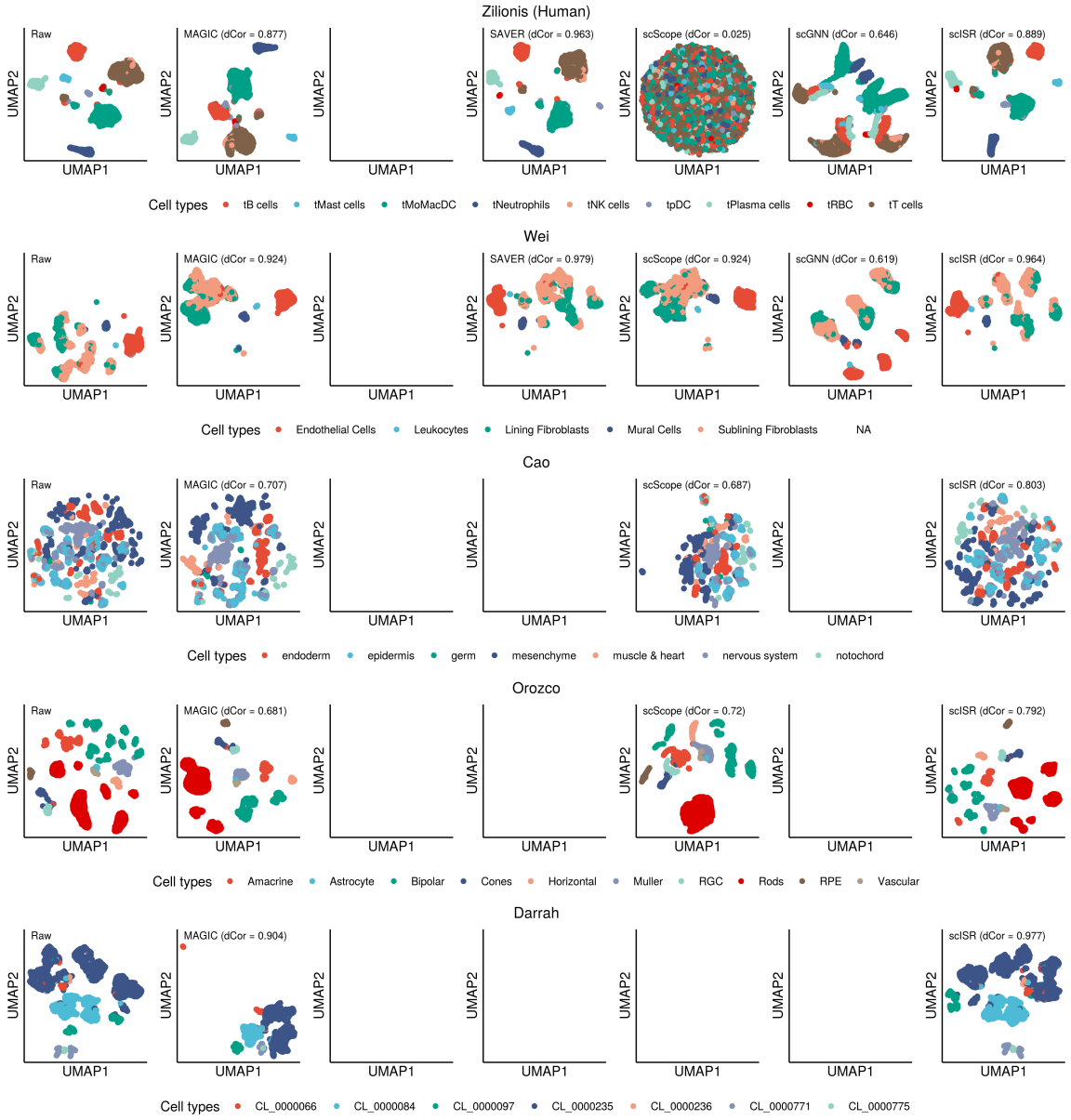

Figure 5.16: Transcriptome landscape for the Zillionis (Human), Wei (Human), Cao, Orozco and Darrah datasets (top to bottom) using UMAP. Different colors code for different cell types. The distance correlation calculated for each imputed dataset shows the similarity between the new landscape (from imputed data) and the original landscape (from raw data).

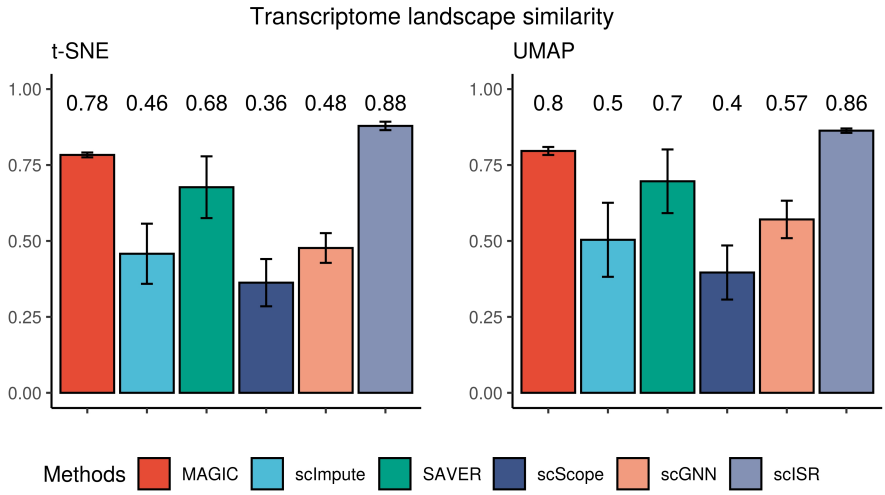

Figure 5.17: The distance correlation between raw data and imputed data using the first two components obtained from t-SNE and UMAP. Higher correlation values indicate more similarity between the imputed and original landscapes. Different colors represent different imputation methods. scISR has the highest mean correlation with the smallest variance. A one-sided Wilcoxon test indicates that the correlation values obtained from scISR are significantly higher than the rest ($p = 3 \times 10^{-9}$ and 2.8×10^{-7} for t-SNE and UMAP, respectively).

est variance. This demonstrates that scISR consistently preserves the transcriptome landscape of the datasets analyzed. MAGIC is the second-best method in this analysis. Using UMAP, scISR obtains a mean correlation of 0.86 compared to those of 0.8, 0.5, 0.7, 0.4, and 0.57, for MAGIC, scImpute, SAVER, scScope, and scGNN, respectively. A one-sided Wilcoxon test also confirms that the correlation values obtained from scISR are significantly higher than the rest ($p = 3 \times 10^{-9}$ and 2.8×10^{-7} for t-SNE and UMAP, respectively).

5.3.1.3 Normalized intra dispersion of imputed genes

For each gene, we calculated the ratio between the intra-cell-type standard deviation and the gene's standard deviation. The intra-cell-type standard deviation measures how similar the expression value of the cells for the underlying gene (cohesion). The ratio (between the intra-cell-type standard deviation and the gene's standard devi-

ation) represents the normalized intra-cell-type standard deviation. We named this as intra dispersion. In general, we expect that with an improved data quality, the expression of cells of the same type are closer to one another compared to cells of different types. Therefore, we expect that a good imputation method would have the smallest intra dispersion. For each gene, we calculate the intra dispersion for the raw and imputed data: one value for raw data and 6 values for 6 imputation methods. Figure 5.18 shows the dispersion for each dataset. scISR has the smallest dispersion compared to raw data and data imputed by 5 other methods. Indeed, the median dispersion of scISR is 3.6×10^{-3} which is much lower compared to 2×10^{-1}, 1.1×10^{2}, 2.4×10^{-1}, 1.3×10^{-1}, 2.3×10^{-2}, and 5.4×10^{1} of raw data and data imputed by MAGIC, scImpute, SAVER, scScope and scGNN, respectively.

5.3.1.4 Running time

Figure 5.19 shows the running time of imputation methods on 25 single-cell datasets. As seen in Figure 5.19, only scISR and MAGIC can analyze the Darrah dataset. scISR is the fastest method and can complete the imputation for this dataset in 50 minutes. MAGIC can analyze the Darrah dataset but it takes 170 minutes to finish the analysis. It takes scScope 350 minutes to analyze the second largest dataset (Orozco 100,000 cells). scImpute, SAVER, and scGNN cannot even analyze the three largest datasets.

5.3.1.5 Simulation studies

To present a comprehensive simulation analysis, we generate a total of 46 datasets in two different scenarios: (1) uniform dropout distribution, (2) normal dropout distribution.

In the first scenario, we generate 6 datasets by varying the number of cells from 100

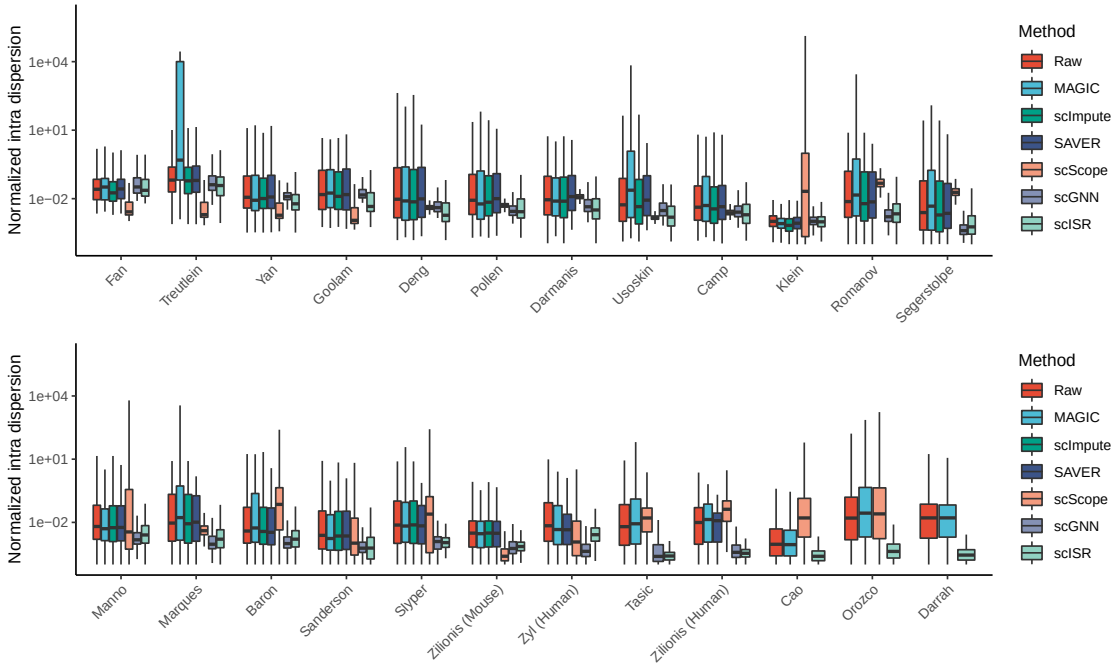

Figure 5.18: Distribution of the normalized intra dispersion for 25 real datasets. For each gene, we calculate the ratio between the intra-cell-type standard deviation and the gene's standard deviation (normalized intra dispersion). We repeat this calculation for all genes for raw and imputed data. The median dispersion of scISR is 3.6×10^{-3} which is much lower compared to 2×10^{-1}, 1.1×10^{2}, 2.4×10^{-1}, 1.3×10^{-1}, 2.3×10^{-2}, and 5.4×10^{1} of raw data and data imputed by MAGIC, scImpute, SAVER, scScope and scGNN, respectively.

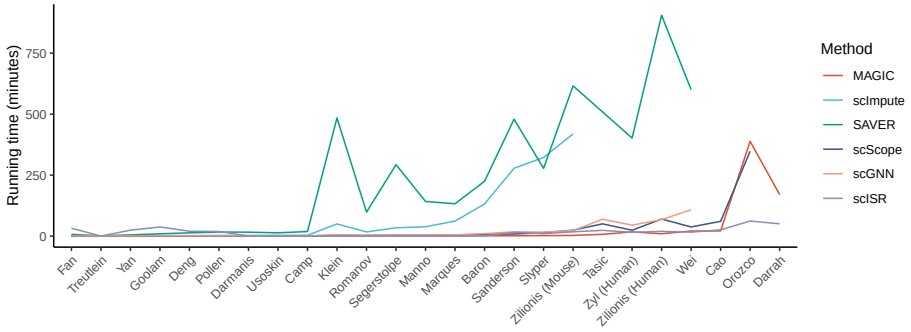

Figure 5.19: Running time of the six imputation methods on 25 real scRNA-seq datasets. scISR is the fastest and can impute the Darrah dataset in 50 minutes.

to 10,000 and the number of genes from 300 to 10,000. The cells/genes combination setups are presented as follows: 100 × 300, 1,000 × 3,000, 3,000 × 9,000, 5,000 × 10,000, 7,000 × 10,000, and 10,000 × 10,000.

In each of the 6 datasets, the expression values follow a normal distribution $\mathcal{N}(\mu, \sigma)$. We set $\mu = 1$ and $\sigma = 0.15$. We slightly shift the mean of the cells and genes by adding a certain value to each group (-1, 0, 1, 1.5 for cell groups and -1, 0, 1 for gene groups) to create 4 different cell types and 3 gene groups – each cell type has an equal number of cells. We name this data as *complete data* and use the expression values as the ground truth for benchmarking. Next, we introduce the dropout events. We randomly select 40% of the genes and consider those as genes that are impacted by dropout events. We randomly assign 30% of the values of these genes to zero. We name this data as *masked data*.

We present a detailed simulation results for 3 datasets with 100, 1,000, and 10,000 cells in Figures 5.20, 5.21 and 5.22. In each figure, panel A shows the transcriptome landscape of the complete data and panel B shows the masked data. In each dataset, the transcriptome landscape and gene-cell heatmap of the *complete data* clearly show the presence of three cell types and four gene groups. With *masked data*, dropout events clearly alter the cells' transcriptome landscape, making it difficult to separate the cell types. The ultimate goal of imputation is to infer the masked (dropout) values in order to recover the original transcriptome landscape and expression profile.

We apply the six imputation methods on the *masked data* and assess the quality of the imputed data by comparing them against the ground truth. Panels C, D, E, F, G, and H in Figures 5.20, 5.21 and 5.22 show the data imputed by MAGIC, scImpute, SAVER, scScope, scGNN, and scISR, respectively. Panels I shows Mean Absolute Error (MAE) and correlation coefficients obtained by comparing masked/imputed data with the complete data. We calculate the MAE and correlation values for each gene and then plot the distributions of each metric using boxplot.

These case studies show that MAGIC imputes the missing values by smoothing the expression values. Many expression values, including non-zero-valued entries,

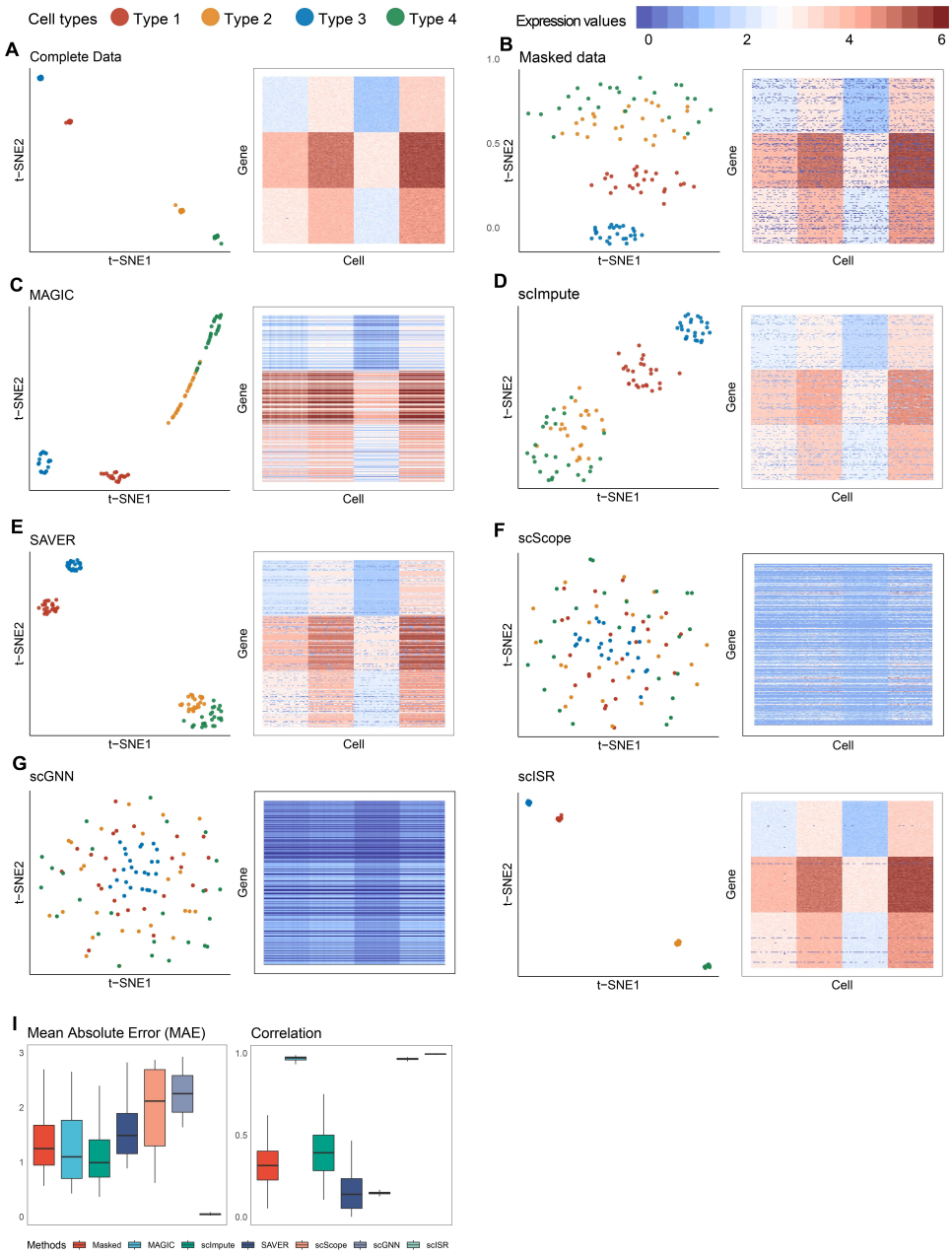

Figure 5.20: Assessment of MAGIC, scImpute, SAVER, scScope, scGNN, and scISR using simulation (100 cells and 300 genes). (A) – (H) The visualization of the *complete data*, *masked data* and *imputed data* recovered by MAGIC, scImpute, SAVER, scScope, scGNN, and scISR. In each subfigure, the left panel shows the transcriptome landscape using t-SNE while the right panel shows the gene-cell heatmap. (I) Mean Absolute Error (MAE) and correlation coefficients obtained by comparing masked/imputed data with the complete data. We calculate the MAE and correlation values for each gene and then plot the distributions of each metric using boxplot. The transcriptome landscapes and heatmaps show that scISR comes closest to recovering the complete data. scISR also has smaller MAE values as well as higher correlation coefficients than other methods.

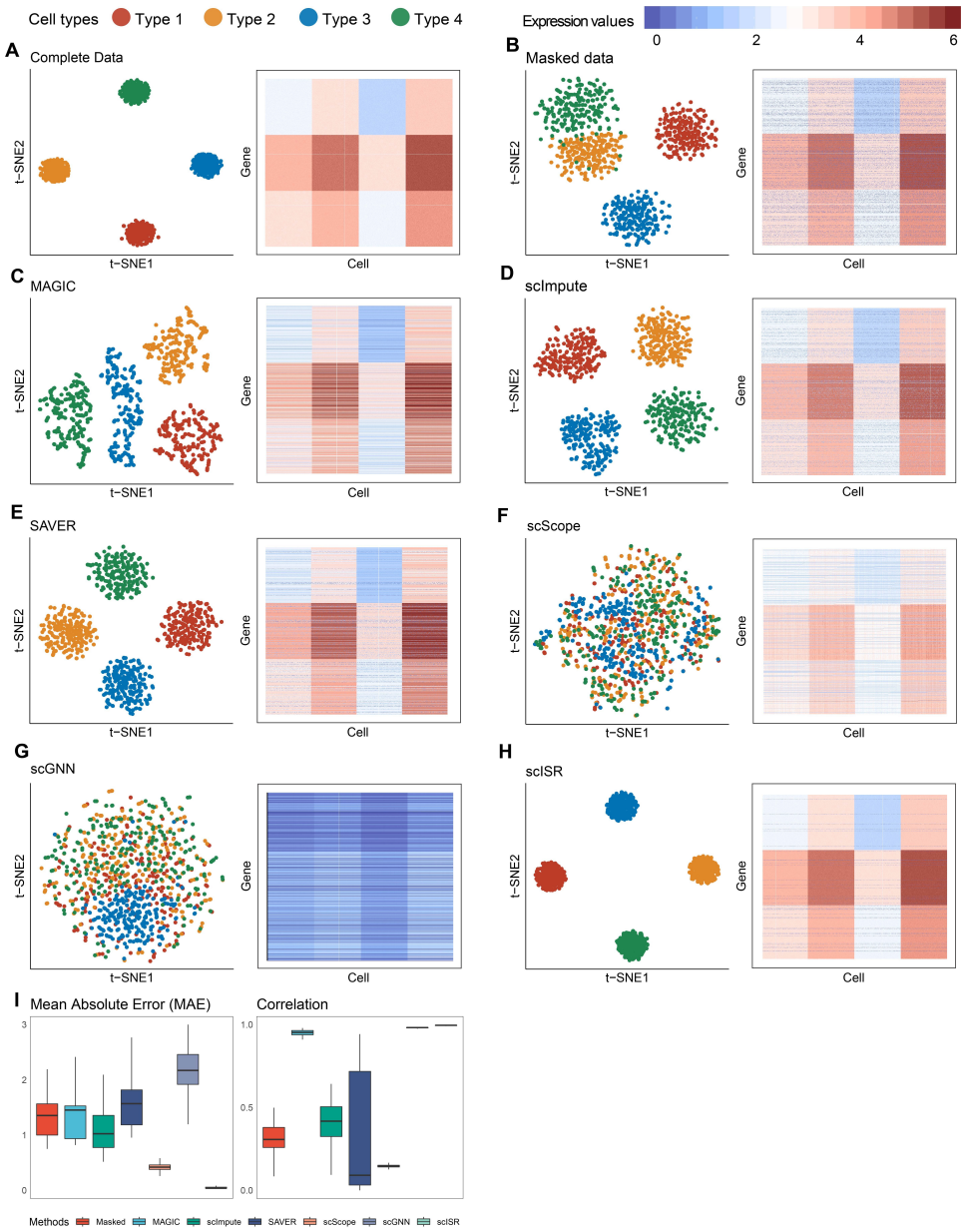

Figure 5.21: Assessment of MAGIC, scImpute, SAVER, scScope, scGNN, and scISR using simulation of 1,000 cells. (A) – (H) The visualization of the *complete data*, *masked data* and *imputed data* recovered by MAGIC, scImpute, SAVER, scScope, scGNN, and scISR. In each subfigure, the left panel shows the transcriptome landscape using t-SNE while the right panel shows the gene-cell heatmap. (I) Mean Absolute Error (MAE) and correlation coefficients obtained by comparing masked/imputed data with the complete data. We calculate the MAE and correlation values for each gene and then plot the distributions of each metric using boxplot. The transcriptome landscapes and heatmaps show that scISR comes closest to recovering the complete data. scISR also has smaller MAE values as well as higher correlation coefficients than other methods.

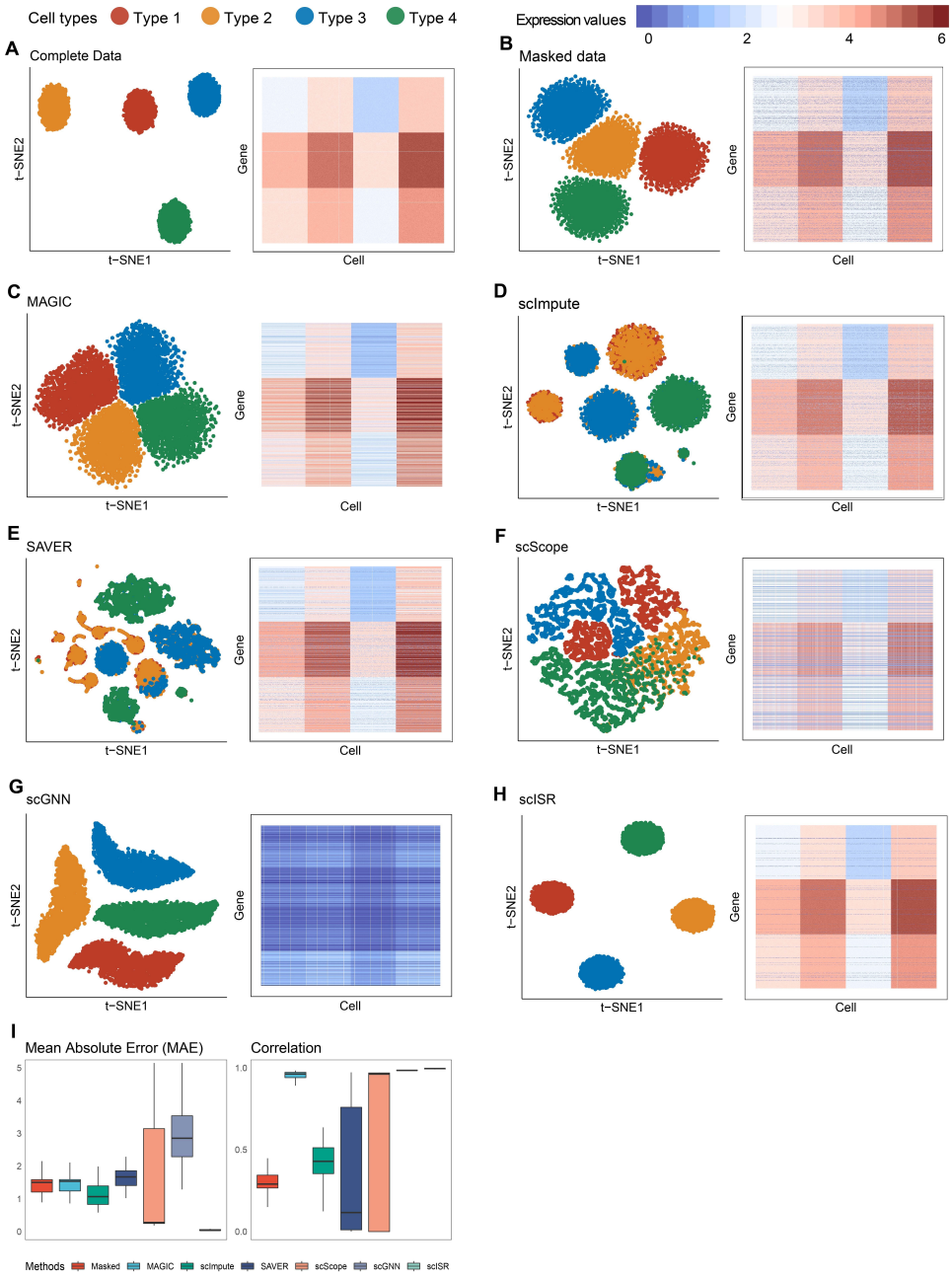

Figure 5.22: Assessment of MAGIC, scImpute, SAVER, scScope, scGNN, and scISR using simulation of 10,000 cells. (A) – (H) The visualization of the *complete data*, *masked data* and *imputed data* recovered by MAGIC, scImpute, SAVER, scScope, scGNN, and scISR. In each subfigure, the left panel shows the transcriptome landscape using t-SNE while the right panel shows the gene-cell heatmap. (I) Mean Absolute Error (MAE) and correlation coefficients obtained by comparing masked/imputed data with the complete data. We calculate the MAE and correlation values for each gene and then plot the distributions of each metric using boxplot. The transcriptome landscapes and heatmaps show that scISR comes closest to recovering the complete data. scISR also has smaller MAE values as well as higher correlation coefficients than other methods.

were altered by MAGIC, making the landscape of the imputed data very different from those of both *complete* and *masked data*. scImpute improves the quality of the data but is still not able to separate some cell types. In addition, scImpute also alters the values of non-zero entries to make the data better fit into the assumed mixture model. SAVER further improves the transcriptome landscape and separates the 4 cell types. However, data imputed by SAVER does not entirely match with the *complete data*, in which many dropout values remain uncorrected many other dropout entries imputed with wrong values. scScope and scGNN oversmooth the imputed data such that it merges all the cells in four types together. The heatmaps clearly show that many expression values, including non-zero-valued entries, were altered by scScope and scGNN. In contrast, scISR is able to recover the transcriptome landscape as well as most of the missing values. The color patterns in the imputed data's heatmap are almost identical to the patterns in the *complete data*. scISR did not alter any non-zero entry and recovered most of the dropout values. The transcriptome landscapes of scISR-imputed data (panels H) are similar to those of the complete data (panels A). scISR also has smaller MAE values as well as higher correlation coefficients than other methods (panels I).

Using the true expression values of the complete data in all 6 datasets, we calculate the mean absolute error (MAE) and correlation between the imputed data and the ground truth for the genes that were impacted by dropout events. Figure 5.23 displays the mean absolute error (MAE) (left panel) and correlation values (right panel) for each method and each cell/gene combination. scISR is the best method in recovering the gene expression values with the smallest MAE and the highest correlation values.

In the second scenario, we generate in total 40 datasets resulted from the combination of 2 different dropout distributions: uniform and normal, 4 different dropout rates: 60%, 70%, 80%, and 90%, and 5 different sizes of data with the number of

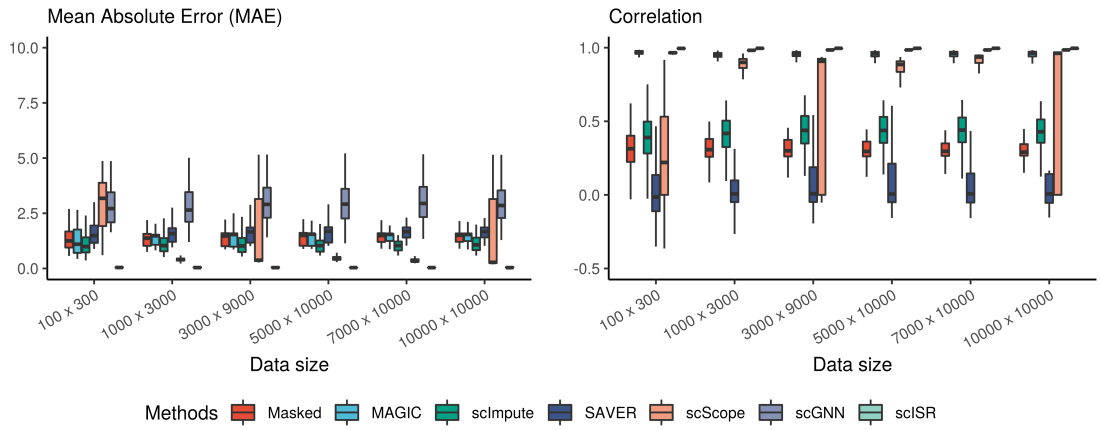

Figure 5.23: Assessment of MAGIC, scImpute, SAVER, scScope, scGNN, and scISR using simulation studies. Mean Absolute Error (MAE) and correlation coefficients were obtained by comparing imputed data with the complete data. In each analysis, scISR has smaller MAE values and higher correlation coefficients than other methods.

cells × genes are: 1,000 × 3,000, 3,000 × 9,000, 5,000 × 10,000, 7,000 × 10,000, and 10,000 × 10,000. Since scISR uses the hypergeometric test, which can be less accurate when the dropout probability does not follow a uniform distribution, we use this simulation to assess the stability of scISR when imputing data with different dropout distributions.

To generate datasets of a certain size (e.g., 1,000 × 3,000), we first generate an expression matrix whose values follow a normal distribution $N(\mu, \sigma)$ where $\mu = 1$ and $\sigma = 0.15$. We then slightly shift the mean of the cells and genes by adding a certain value to each group (-1, 0, 1, 1.5 for cell groups and -1, 0, 1 for gene groups) to create 4 different cell types. We name this as *complete data*. Next, we randomly assign dropout values to the data in two different cases. In the first case, the dropout probability is uniformly distributed. In the second case, the dropout probability follows a normal distribution. For example, at 60% dropout rate, the dropout probability follows a distribution of $N(0.6, 0.1)$. We then vary the dropout rate from 60% to 90%. We name the data with dropouts as *masked data*. Next, we impute the *masked data* using imputation methods to obtain the *imputed data*. Finally, to assess the performance of

imputation methods, we compare the imputed data against the complete data using Mean Absolute Error (MAE) and correlation coefficients.

The top left panel in Figure 5.24 shows the MAE values obtained for datasets with 1,000 cells and 3,000 genes. In this panel, the left side displays the results obtained for uniform distributions while the right side shows the results for the normal distributions. When the dropout probability is uniformly distributed, scISR is able to recover most of the dropout values, resulting in a median MAE close to zero at any dropout rate. When the dropout probability is normally distributed, scISR still performs as well at 60% to 80% dropout but it becomes less accurate at 90% rate. At 90% dropout rate, scISR recovers only a part of the data (median MAE of approximately 2.11 compared to 3.65 of masked data). Assessment results using correlation coefficient (top right panel) also confirm our finding. However, as seen in Figure 5.24, the result of scISR is still much better than other imputation methods.

The next two panels (second row) in Figure 5.24 show the results obtained for datasets with 3,000 cells and 9,000 genes. scISR is more accurate (lower MAE and higher correlation) for these datasets compared to datasets with 1,000 cells. At dropout rates of 60%, 70%, and 80%, scISR performs consistently well for uniform and normal distributions alike (median MAE value close to zero). At 90% rate, the median MAE of scISR for normal distributions is now 1.61 (compared to 2.11 for datasets with 1,000 cells and 3,000 genes). The reason for such improvement is that with the same dropout rate, larger datasets provide us with more data to learn from, leading to improved hypothesis testing (hypergeometric test) and prediction (linear regression). For datasets with 7,000 cells or more, the median MAE is close to zero for both uniform and normal distributions at any dropout rate. In summary, scISR (using hypergeometric test) performs well for large datasets with high dropout rates even when the dropout probability is not uniformly distributed. Moreover, scISR

also outperforms other methods in recovering the missing data by having the lowest median MAE and highest median correlation.

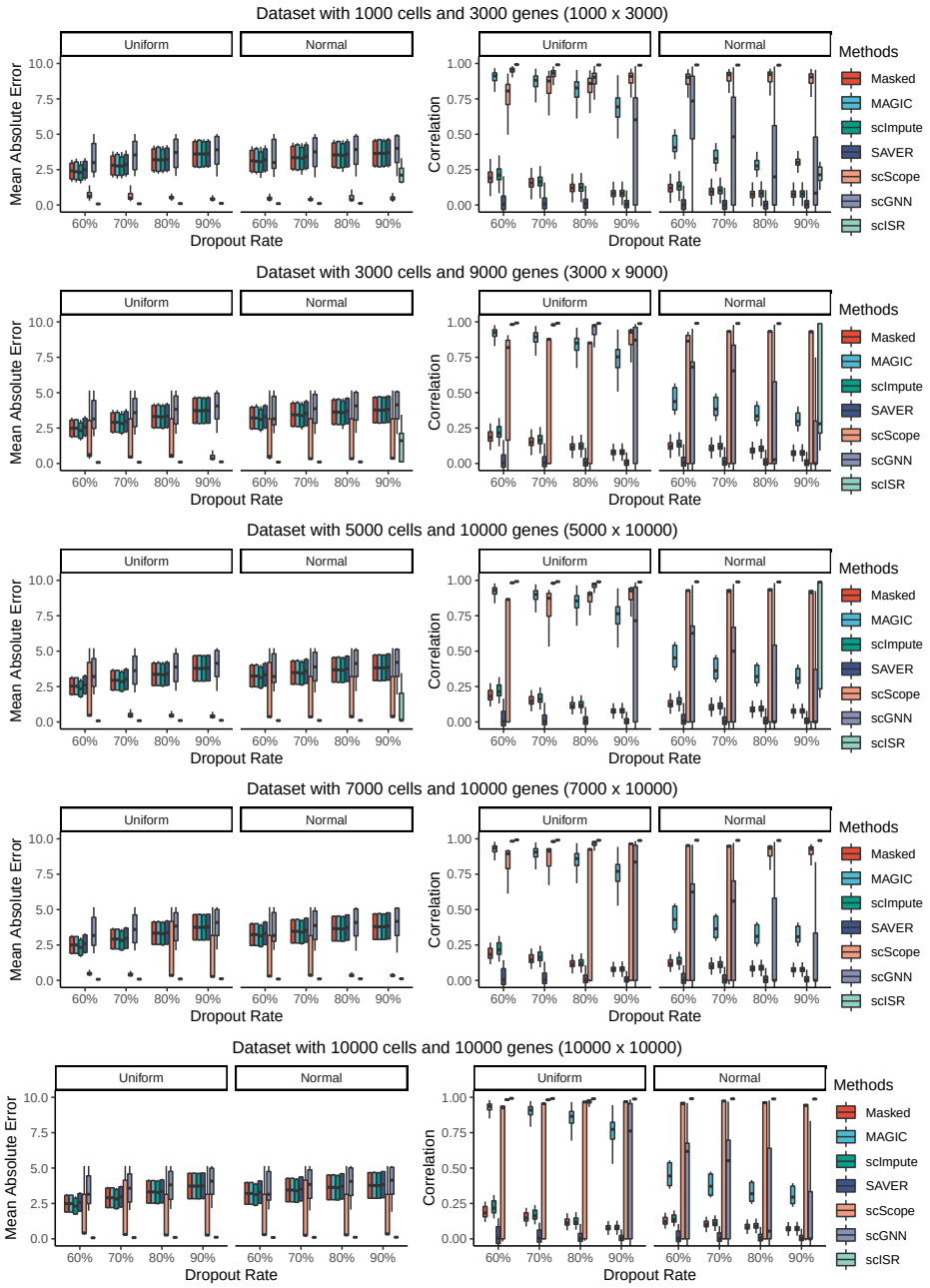

Figure 5.24: Assessment of MAGIC, scImpute, SAVER, scScope, scGNN, and scISR using simulated datasets with different dropout distributions and sample sizes. The left panels show the Mean Absolute Error (MAE) values while the right panels show the correlation coefficients. In each panel, the left side shows the results for uniform distributions while the right side shows the results for normal distributions. For small datasets (e.g., datasets with 1,000 cells) with high dropout rates, scISR is less accurate when the dropout probability is normally distributed. When the sample size increases, scISR becomes more accurate. For datasets with 7,000 cells or more, scISR performs well for both uniform and normal distributions alike across all dropout rates. For most of the dataset sizes and dropout rates, scISR have a much better median MAE and correlation compared to other methods.

5.4 Conclusion (scISR)

In this work, we introduced a new method to mitigate the effects of dropout events that frequently happen during the sequencing process of individual cells. The contribution is two-fold. First, by introducing a hypothesis testing procedure, we avoid altering true zero values. Second, the subspace regression provides a more accurate imputation by limiting the imputation to gene groups with similar expression patterns. We compared our approach with state-of-the-art methods using 25 real scRNA-seq datasets and 116 simulated datasets. We demonstrated that scISR outperforms other imputation methods in improving the quality of clustering analysis. At the same time, we also demonstrated that scISR preserves the transcriptome landscape of each dataset. Finally, we showed that scISR is robust against different dropout rates and distributions. We expect that scISR will be a very useful method that can improve the quality of single-cell data. The tool can be seamlessly incorporated into other single-cell analysis pipelines.

Chapter 6

scINN: Single-cell RNA Sequencing Data Imputation using Similarity Preserving Network

This chapter is based on the following publication: ***Duc Tran****, Hung Nguyen, Frederick C. Harris, and Tin Nguyen. Single-cell RNA sequencing data imputation using similarity preserving network. In Proceedings of the 13th International Conference on Knowledge and Systems Engineering (KSE), 2021.*

Recent advancements in single-cell RNA sequencing (scRNA-seq) technologies have allowed us to monitor the gene expression of individual cells. This level of detail in monitoring and characterization enables the research of cells in rapidly changing and heterogeneous environments such as early stage embryo or tumor tissue. However, the current scRNA-seq technologies are still facing many outstanding challenges. Due to the low amount of starting material, a large portion of expression values in scRNA-seq data is missing and reported as zeros. Moreover, scRNA-seq platforms are trending toward prioritizing high throughput over sequencing depth, which makes the

problem become more serious in large datasets. These missing values can greatly affect the accuracy of downstream analyses. Here we introduce a neural network-based approach, named single-cell Imputation using Neural Network (scINN), that can reliably recover the missing values in single-cell data and thus can effectively improve the performance of downstream analyses. To impute the dropouts in single-cell data, we build a neural network that consists of two sub-networks: imputation sub-network and quality assessment sub-network. We compare scINN with state-of-the-art imputation methods using 10 scRNA-seq datasets with a total of more than 100,000 cells. In an extensive analysis, we demonstrate that scINN outperforms existing imputation methods in improving the identification of cell sub-populations and the quality of transcriptome landscape visualization.

6.1 Introduction

The ability to monitor and characterize biological samples at single-cell resolution has opened up many novel research fields, such as studying cells in early embryonic stage or decomposition heterogeneous environment of cancer tumors [109, 120]. These promising applications have led to the generation of a massive amount of single-cell data, where each dataset consists of hundreds of thousands of cells [145, 146].

Current single-cell RNA sequencing (scRNA-seq) technologies still need to overcome significant challenges to ensure the accurate measurement of gene expression [94, 172]. One notable challenge of scRNA-seq is the dropout events, which happen when a gene that generally has high expression values but does not express in some cells [101]. The source of these errors can be attributed to the limitation of sequencing technologies. Due to the low amount of starting mRNA collected from individual cells, failed amplification can happen and causes the expression values to be inaccurately reported [102, 103, 173]. This leads to an excessive amount of zeros in the expres-

sion values of scRNA-seq data. On the other hand, the zero expression values can also be due to biological variability. Since most downstream analyses of scRNA-seq are performed on gene expression data, it is essential to have a precise expression measurement. Therefore, imputing scRNA-seq data to recover the information loss caused by dropout events would greatly improve the quality of downstream analyses.

Thus far, numerous methods have been developed to infer the missing values caused by dropout events [104–107, 158, 174–176]. Those methods can be classified into two categories: (i) statistical-based methods, and (ii) diffusion smooth-based methods. Methods in the first category include bayNorm [174], SAVER [106], scImpute [105], scRecover [177], and RIA [158]. These methods typically model the data as a mixture of distributions. For example, scImpute models the gene expression as a mixture of two different distributions: the Gaussian distribution represents the actual gene expression while the Gamma distribution accounts for the dropout events. Similarly, SAVER [106] models read counts as a mixture of Poisson-Gamma and then uses a Bayesian approach to estimate true expression values of genes by borrowing information across genes. More recent methods, RIA [158] and scIRN [176], assume that highly expressed genes follow a normal distribution and apply hypothesis testing method to identify true dropouts. Next, they impute missing values by using a linear regression model. All of these methods assume the gene expression data follows a specific distribution, which does not always hold true in reality. In addition, exiting methods involve the estimation of many parameters for genes across the whole genome. This can potentially lead to overfitting and high time complexity.

Methods in the second category include DrImpute [107], MAGIC [104], and kNN-smoothing [175]. MAGIC imputes zero expression values using a heat diffusion algorithm [157]. It constructs the affinity matrix between cells using a Gaussian kernel and then constructs a Markov transition matrix by normalizing the sc-RNA similar-

ity matrix. Next, MAGIC estimates the weights of other cells using the transition matrix. Another method is DrImpute [107] that is based on the cluster ensemble and consensus clustering. It performs clustering for a predefined number of times and imputes the data by averaging expression values of similar cells. If the number of clusters is not provided by users, DrImpute uses some default values that might not be optimal for the data. kNN-smoothing is designed to reduce noise by aggregating information from similar cells (neighbors). The method assumes that the zero counts of scRNA-seq data follows a Poisson distribution. For cells that contain zero counts, kNN-smoothing performs a smoothing step using each cell's k nearest neighbors either through the application of diffusion models or weighted sums. The major drawback of these methods is that they rely on many parameters to fine-tune their model, which often leads to over-smoothing the data.

6.2 Methodology

Here we propose a new approach, single-cell Imputation using Neural Network (scINN), that can reliably impute missing values from single-cell data. The method consists of two steps. The first step is to generate an accurate clustering result of the original data, and calculate the similarity between all pairs of samples. The second step is to estimate the missing values using a neural network and the similarity information generated in the first module. The approach is evaluated using 10 single-cell datasets in comparison with four other methods. We demonstrate that scINN outperforms existing imputation methods (DrImpute [107], MAGIC [104], scImpute [105], and SAVER [106]) in improving the identification of cell sub-populations and the quality of biological landscape.

The input of scINN is an expression matrix, in which rows represent cells and columns represent genes or transcripts. The overall workflow of scINN is described

in Figure 6.1, which consists of two modules: (i) generating an accurate clustering results of the original data, and calculating the similarity between all samples, and (ii) imputing the dropout values. The purpose of the first module is to learn the similarity information between each pair of samples. The output of the first module is the clustering assignments for samples in the dataset, and a similarity matrix with Pearson correlations for all pairs of samples. These information are used as the target for the second module. In the second module, we impute the original data using a neural network. The parameters of the neural network are repeatedly adjusted so that the clustering assignments and similarity matrix inferred from the imputed data is as similar to the outputs of the first module as possible. The details of each step are described in the following subsections.

6.2.1 Generating similarity information

To generate a compressed, low-dimensional representation of original data, we apply our previously developed method, called scDHA [178]. scDHA consists of two core modules. The first module is a non-negative kernel autoencoder that can filter out genes or components that have insignificant contributions to the representation. The second module is a Stacked Bayesian Self-learning Network that is built upon the Variational Autoencoder [112] to project the filtered data onto a much lower-dimensional space. The output of scDHA is a low-dimensional matrix that preserves the global structure of the original data. Using this representation, scDHA can cluster the samples into groups with high accuracy. We also generate the similarity matrix for all samples in the dataset. The similarity between two samples is measured by Pearson correlation. We use the similarity information between samples in the dataset to optimize our imputation module so the same information can be inferred from imputed data using a network with simpler structure.

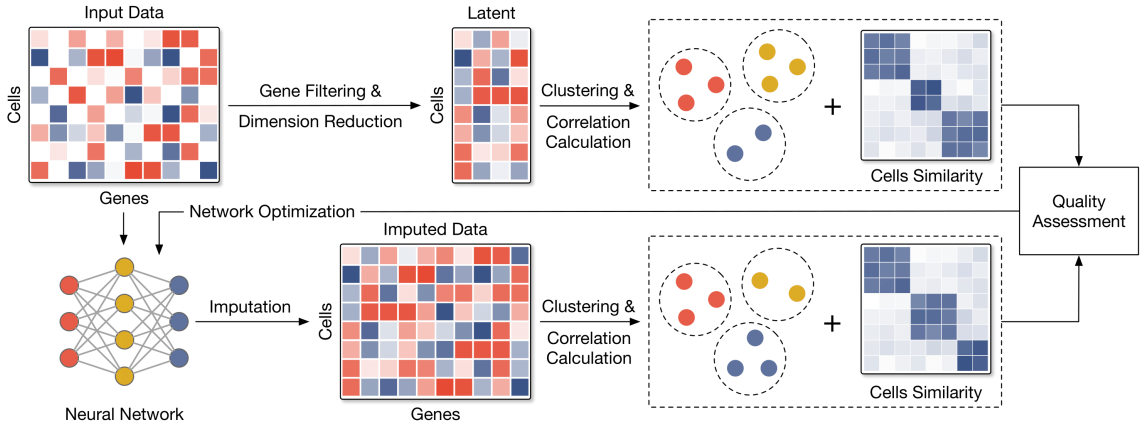

Figure 6.1: The workflow of single-cell Imputation using Residual Network (scINN). The first module (similarity module, upper part) generates an accurate clustering result of the data, and calculates the similarity between all pairs of samples. The input data is first filtered using a one-layer, non-negative kernel autoencoder to remove genes that have insignificant contribution to the global structure of the data. Next, the data is projected onto a low-dimensional space to obtain a compressed data matrix (latent data). Using this latent data, we cluster the samples into groups and compute the similarity matrix for all samples. In the second module (imputation module, lower part), zero values in input matrix are imputed using a neural network-based imputation model. These imputed values are added to original data without modifying the non-zeros values to produce the imputed data. The parameters of the neural network are repeatedly adjusted so that the clustering assignments and similarity matrix inferred from the imputed data is as similar to the output of the first module as possible.

6.2.2 Imputing dropout data using neural network

To impute the dropouts in single-cell data, we build a neural network that consists of two sub-networks. The first network aims to infer the true value of zeros in the data. The output is a matrix with the same size as the input, in which the values at zero positions are modified. The non-zero values remain the same as of the original data. The second network aims to infer the clusters of input cells and the Pearson correlations between them. By minimizing the difference between the inferred results and the results from the first module, the imputed values are ensured to have high accuracy.

The formulation of the neural network can be written as:

$$\begin{aligned} X_I &= f_I(X) \\ C + S &= f_P(X_I) \end{aligned}$$

where $X \in R_+^n$ is the input of the model (X is simply the original data), f_I and f_P represent the transformation by the two sub-networks, f_I imputes the zero values in the data, f_P predicts the clusters of the input cells and the correlations between them, C is the clustering results, and S is the similarity matrix between all input cells. The network is optimized by minimizing: (i) the binary cross entropy loss between the inferred clusters and the clustering result from the first module, and (ii) the mean square error loss between the inferred similarity matrix and the similarity matrix calculated using the representations from the first module.

6.3 Validation and Analysis Results

We compare our method with four state-of-the-art imputation methods: DrImpute [107], MAGIC [104], scImpute [105], and SAVER [106]. Each of these methods repre-

sents a distinct strategy to single-cell data imputation: DrImpute integrates clustering result from other software, MAGIC is a Markov-based technique, while scImpute and SAVER use statistical models. Table 6.1 shows the 10 datasets used in our data analysis. These scRNA-seq datasets are available on NCBI [179], and ArrayExpress [180]. The processed data of the first 7 datasets are downloaded from Hemberg lab's website (`https://hemberg-lab.github.io/scRNA.seq.datasets`). In each dataset, the cell sub-populations are known. We used this information *a posteriori* to assess how the imputation methods improve the identification of cell populations, and how they enhance the visualization of transcriptome landscapes.

For each dataset, we used the above methods to impute the data. The quality of the imputed data is assessed using two downstream analyses: clustering and visualization. For clustering, we partitioned the data using k-means and compared the obtained partitioning against the true cell types using Adjusted Rand index (ARI) [148]. For visualization, we used UMAP [62] to generate the 2D representation and then calculated the silhouette index (SI) [149] of the 2D representation. SI measures the cohesion among cells of the same type, as well as the separation between different cell types.

6.3.1 scINN improves the identification of sub-populations

Given a dataset, we used the five methods to impute the data. After imputation, we have 6 matrices: the raw data and five imputed matrices (from DrImpute, MAGIC, scImpute, SAVER, and scINN). To assess how separable the cell types in each matrix is, we reduced the number of dimensions using PCA and then clustered the data using k-means where k is the true number of cell types. The accuracy of cluster assignments is measured by ARI.

Figure 6.2 shows the ARI values for the raw and imputed data. Existing methods

Table 6.1: Description of the 10 single-cell datasets used to assess the performance of imputation methods.

Dataset	Accession ID	Tissue	Sequencing Protocol	Drop. Rate	Class	Size
1. Yan	GSE36552	Human Embryo	Tang	0.456	6	90
2. Goolam	E-MTAB-3321	Mouse Embryo	Smart-Seq2	0.685	5	124
3. Deng	GSE45719	Mouse Embryo	Smart-Seq	0.605	6	268
4. Camp	GSE75140	Human Brain	SMARTer	0.801	7	734
5. Klein	GSE65525	Mouse Embryo	inDrop	0.658	4	2,717
6. Romanov	GSE74672	Human Brain	SMARTer	0.878	7	2,881
7. Baron	GSE84133	Human Pancreas	inDrop	0.906	14	8,569
8. Tasic	GSE115746	Mouse Visual Cortex	SMART-Seq	0.798	6	23,178
9. Zilionis	GSE127465	Human Lung	inDrop	0.982	9	34,558
10. Hrvatin	GSE102827	Mouse Visual Cortex	inDrop	0.942	8	48,266

improve cluster analysis in some datasets but decreases the ARI values in some others. For example, SAVER has higher ARIs than the raw data for the Goolam, Camp, Klein, Romanov, Baron, and Zilionis but has lower ARIs in the remaining 4 datasets. scINN is the only method able to improve the clustering performance compared to raw data in every dataset. Moreover, scINN has the highest ARIs in all but Zilionis datasets. The average ARI of scINN-imputed data is 0.72, which is higher than those obtained from raw data and data imputed by DrImpute, MAGIC, scImpute, SAVER (0.52, 0.58, 0.48, 0.36, 0.53, respectively).

For a more comprehensive analysis, we also report the assessment using normalized mutual information (NMI) and Jaccard index (JI) [168] in Figures 6.3 and 6.4, respectively. Regardless of the assessment metrics, scINN outperforms other methods by having the highest NMI (9/10 datasets) and JI (9/10 datasets) values. These results demonstrate that cluster analysis using scINN-imputed data leads to a better accuracy than using the raw data or data imputed by other imputation methods.

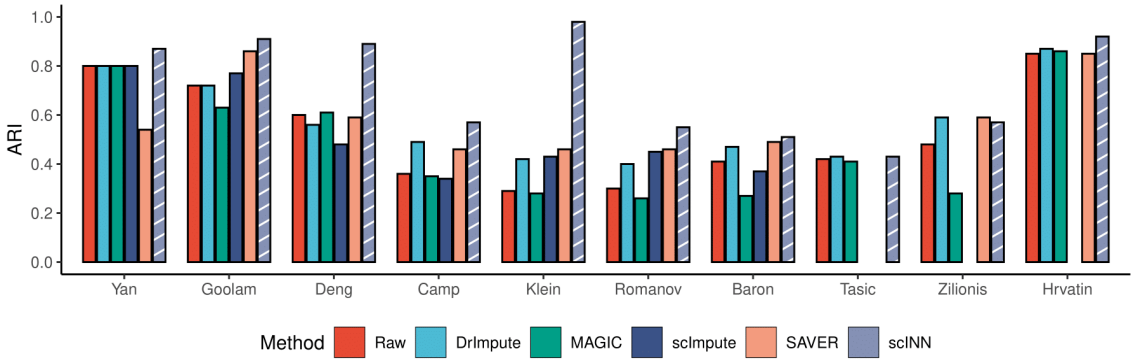

Figure 6.2: Adjusted Rand index (ARI) obtained from clustering on raw data and data imputed by DrImpute, MAGIC, scImpute, SAVER, and scINN. The x-axis shows the names of the datasets while the y-axis shows ARI value of each method. scINN outperforms other methods in all datasets except Zilionis.

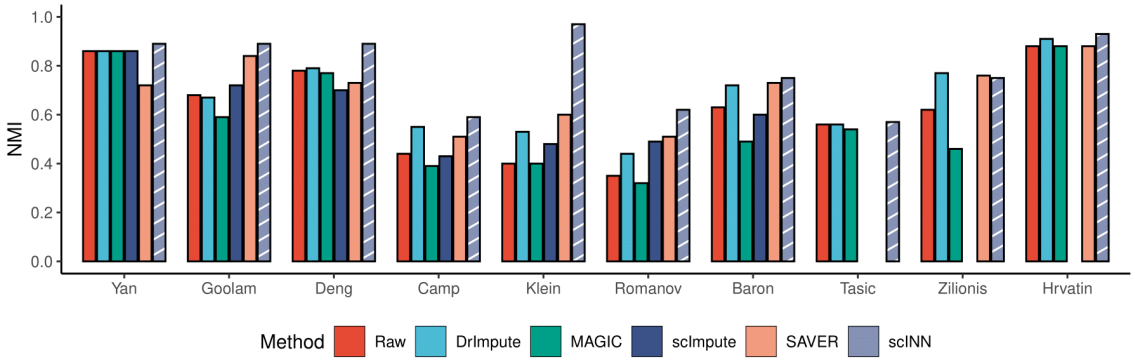

Figure 6.3: Normalized mutual information (NMI) obtained from clustering on raw data and data imputed by DrImpute, MAGIC, scImpute, SAVER, and scINN. The y-axis shows NMI value of each method. scINN outperforms other methods in all datasets except Zilionis.

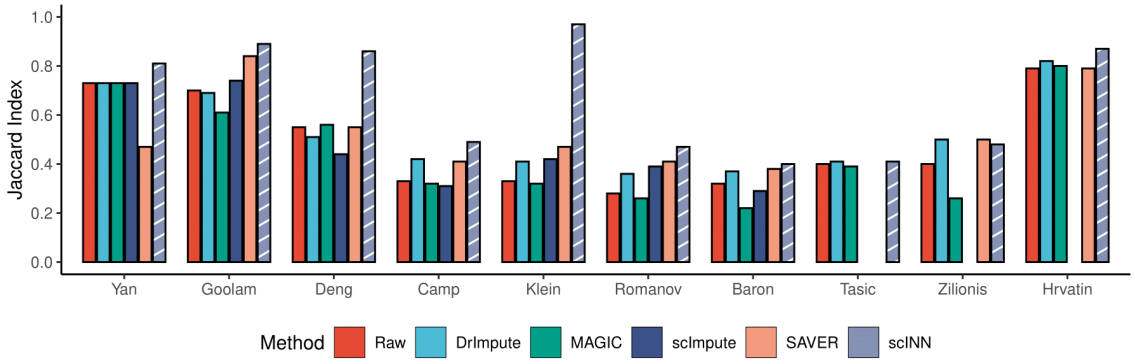

Figure 6.4: Jaccard index (JI) obtained from clustering on raw data and data imputed by DrImpute, MAGIC, scImpute, SAVER, and scINN. The y-axis shows JI value of each method. scINN outperforms other methods in all datasets except Zilionis.

6.3.2 scINN improves transcriptome landscape visualization

In this subsection, we demonstrate that scINN improves the visualization of the single-cell data. We used UMAP [62] to generate the transcriptome landscapes from raw and data imputed by DrImpute, MAGIC, scImpute, SAVER, and scINN. We performed data visualization and calculated the silhouette index for each of the 10 datasets. Figure 6.5 shows the SI values obtained for the raw data and data imputed by the five imputation methods. The figure shows that scINN can improve the quality of data visualization in most of the datasets (8/10 datasets). These results demonstrate that data imputation using scINN would lead to a much better visualization of transcriptome landscapes compared to using raw data or data imputed by other methods.

Figure 6.6 shows the transcriptome landscapes of the Klein dataset. The 2D representation of scINN-imputed data is the only one that has four separable groups, corresponding to the four real cell types. The landscapes generated using raw and data imputed by other methods have different cell types mixed together. The data imputed by scINN has the highest SI value (0.77 compared to 0.68 of the second best).

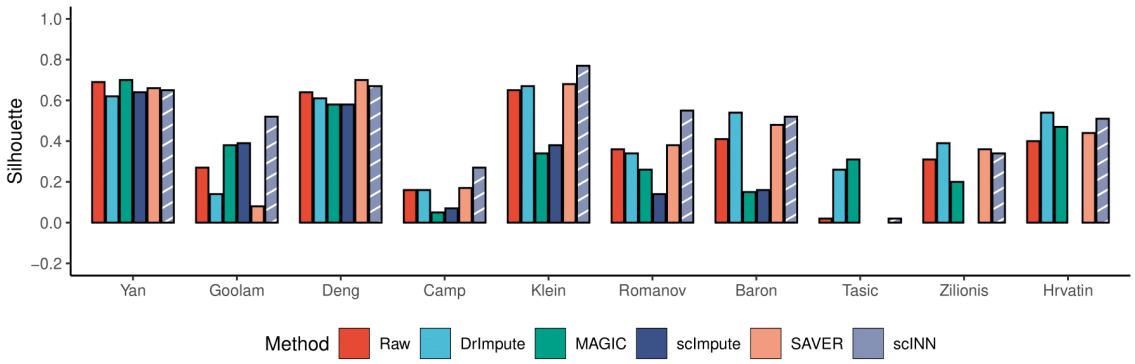

Figure 6.5: Visualization quality using raw and imputed data, measured by silhouette index (SI). The y-axis shows SI value of each method.

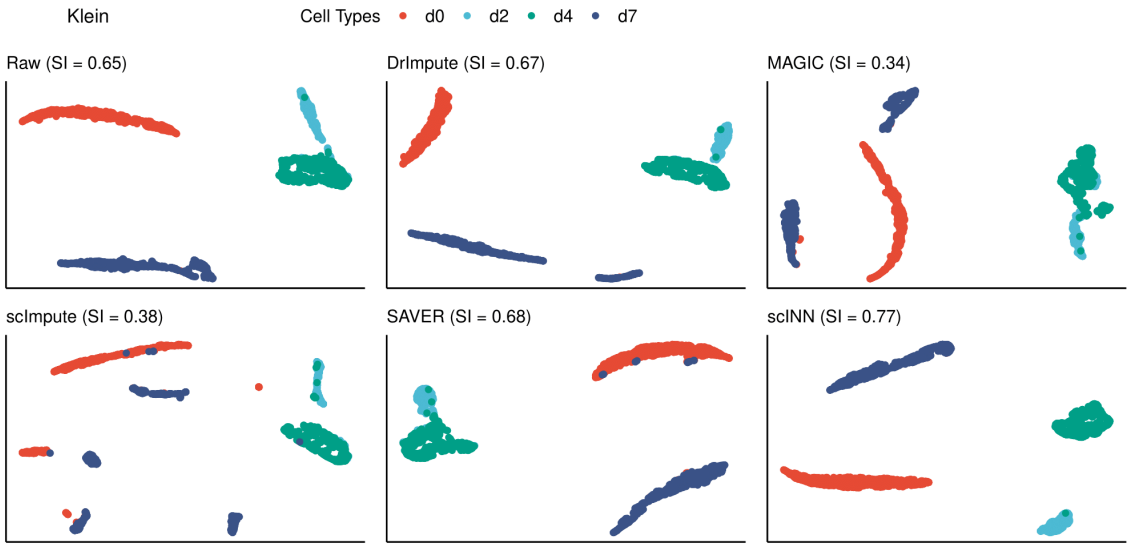

Figure 6.6: Transcriptome landscape of the Klein dataset. The scatter plot shows the first two principal components calculated by UMAP. Different colors represent different cell types. The 2D representation generated by scINN has a clear structure, where cells from different groups are separated from one other.

6.4 Conclusion (scINN)

We introduced a new method, scINN, to recover the missing data caused by dropout events in scRNA-seq data. We compared scINN with four state-of-the-art imputation methods using 10 scRNA-seq datasets. scINN outperformed existing approaches in improving the identification of cell sub-populations. scINN also improved the quality of transcriptome landscapes generated by UMAP. A potential improvement of this research is to investigate the scalability of scINN by analyzing datasets with higher number of cells. Another direction is to investigate the imputation method in other research applications, including pseudo-time trajectory inference and supervised learning.

Chapter 7

scIRN: Single-cell RNA Sequencing Data Imputation using Deep Neural Network

This chapter is based on the following publication: ***Duc Tran****, Bang Tran, Hung Nguyen, Frederick C. Harris, Nam Sy Vo, and Tin Nguyen. Single-cell RNA sequencing data imputation using deep neural network. In Proceedings of the 18th International Conference on Information Technology-New Generations (ITNG), 2021.*

Recent research in biology has shifted the focus toward single-cell data analysis. The new single-cell technologies have allowed us to monitor and characterize cells in early embryonic stage and in heterogeneous tumor tissue. However, current single-cell RNA sequencing (scRNA-seq) technologies still need to overcome significant challenges to ensure accurate measurement of gene expression. One critical challenge is to address the dropout event. Due to the low amount of starting material, a large portion of expression values in scRNA-seq data is missing and reported as zeros. These

missing values can greatly affect the accuracy of downstream analysis. Here we introduce a neural network-based approach, named single-cell Imputation using Residual Network (scIRN), that can reliably recover the missing values in single-cell data and thus can effectively improve the performance of downstream analyses. To impute the dropouts in single-cell data, we build a neural network that consists of two sub-networks: imputation sub-network and quality assessment sub-network. We compare scIRN with state-of-the-art imputation methods using 10 scRNA-seq datasets. In our extensive analysis, scIRN outperforms existing imputation methods in improving the identification of cell sub-populations and the quality of visualizing transcriptome landscape.

7.1 Methodology

Here we propose a new approach, single-cell Imputation using Residual Network (scIRN), that can reliably impute missing values from single-cell data. Our method consists of two steps. The first step is to generate a compressed and accurate low-dimensional representation of the original data. The second step is to estimate the missing values using a neural network and information from the low-dimensional representation. The approach is tested using 10 single-cell datasets in comparison with four other methods. We demonstrate that scIRN outperforms existing imputation methods (MAGIC [104], scImpute [105], SAVER [106], and DrImpute [107]) in improving the identification of cell sub-populations and the quality of biological landscape.

The input of scIRN is an expression matrix, in which rows represent cells and columns represent genes or transcripts. The overall workflow of scIRN is described in Figure 7.1, which consists of two modules: (i) generating a low-dimensional, non-redundant representation of the original data, and (ii) imputing the dropout values.

The purpose of the first module is to remove redundant signals and noise from the data. The output of the first module is a low-dimensional, non-redundant representation of the original data. This presentation is used as the target for the second module. In the second module, we impute the original data using a residual network. The parameters of the residual network are repeatedly adjusted so that the compressed representation of the imputed data is as similar to the non-redundant representation as possible. The details of each step are described in the following sections.

7.1.1 Generating low-dimensional, non-redundant representation

To generate a compressed, low-dimensional representation of original data, we apply our previously developed method, called scDHA [178]. scDHA consists of two core modules. The first module is a non-negative kernel autoencoder that can filter out genes or components that have insignificant contributions to data representation. The second module is a Stacked Bayesian Self-learning Network that is built upon the Variational Autoencoder [112] to project the filtered data onto a much lower-dimensional space. The output of scDHA is a low-dimensional matrix that preserves the global structure of the original data. This representation is used as the training target for the imputation module.

7.1.2 Imputing dropout data using residual network

To impute the dropouts in single-cell data, we build a neural network that consists of two sub-networks. The first network aims to infer the true value of zeros in the data. The output is a matrix with the same size as the input, in which the values at zero positions are modified. The non-zero values remain the same as of the original data.

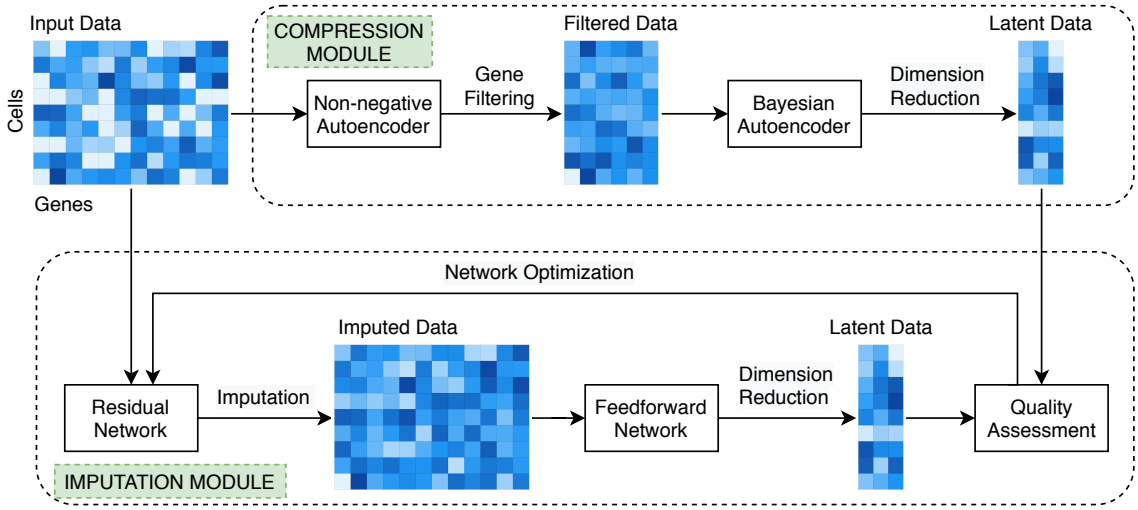

Figure 7.1: The overall workflow of single-cell Imputation using Residual Network (scIRN). The first module (compression module) generates a compressed, low-dimensional representation of original data. The input data is first filtered (using an one-layer, non-negative kernel autoencoder) to remove genes that have insignificant contribution to the global structure of the data. After that, we project the data into a low-dimensional space to obtain a compressed data matrix (latent data). This latent data is used as the training target for the imputation process. In the second module (imputation module), zero values in input matrix are imputed using a neural network-based imputation model. These imputed values are added to original data without modifying the non-zeros values to produce the imputed data matrix. The imputed data is compressed to a low-dimensional space (latent data). The parameters of the imputation module is repeatedly optimized by minimizing the difference between the two latent matrices.

The second network aims to compress the imputed data to a lower dimension. This compressed data has the same size as the representation generated in the first step. By minimizing the difference between the representation generated from imputed data and the representation from the first step, the imputed values are ensured to have high accuracy.

The formulation of the neural network can be written as:

$$\begin{aligned} X_I &= f_I(X) \\ Z' &= f_C(X_I) \end{aligned}$$

where $X \in R_+^n$ is the input of the model (X is simply the original data), f_I and f_C represent the transformation by the two sub-networks, f_I imputes the zero values in the data, f_C compresses the imputed data onto a lower-dimensional space, and $Z' \in R^m$ ($m << n$) is the compressed data. For the f_I transformation, we use residual network [113] for a more stable and accurate imputation process. The network is optimized by minimizing $\|Z' - Z\|_2^2$, where Z is the low-dimensional representation generated by scDHA.

7.2 Validation and Analysis Results

We compares our method with four state-of-the-art imputation methods: MAGIC [104], scImpute [105], SAVER [106], and DrImpute [107]. Each of these methods represents a distinct strategy to single-cell data imputation: MAGIC is a Markov-based technique, DrImpute integrates clustering result from other software, while scImpute and SAVER use statistical models. Table 7.1 shows the 10 datasets used in our data analysis. The processed datasets were downloaded from Hemberg lab's website (`https://hemberg-lab.github.io/scRNA.seq.datasets`). In each dataset, the cell sub-populations are known. We used this information *a posteriori* to assess

how the imputation methods improve the identification of cell populations, and how they enhance the visualization of transcriptome landscapes.

For each dataset, we used the above methods to impute the data. The quality of the imputed data is assessed using two downstream analyses, clustering and visualization. For clustering, we partitioned the data using k-means and compared the obtained partitioning against the true cell types using Adjusted Rand index (ARI) [148]. For visualization, we used UMAP [62] to generate the 2D representation and then calculated the silhouette index (SI) [149] of the 2D representation. SI measures the cohesion among cells of the same type, as well as the separation between different cell types.

7.2.1 scIRN improves the identification of sub-populations

Given a dataset, we used the five methods to impute the data. After imputation, we have 6 matrices: the raw data and five imputed matrices (from MAGIC, scImpute, SAVER, DrImpute, and scIRN). To assess how separable the cell types in each matrix is, we reduced the number of dimensions using PCA and then clustered the data using k-means. The accuracy of cluster assignments is measured by ARI.

Figure 7.2 shows the ARI values for the raw and imputed data. Existing methods improve cluster analysis in some datasets but decreases the ARI values in some others.

Table 7.1: Description of the 10 single-cell datasets used to assess the performance of imputation methods.

Dataset	Tissue	Size	Class	Protocol	Accession ID	Reference
1. Deng	Mouse Embryo	268	6	Smart-Seq2	GSE45719	Deng *et al.*, 2014 [120]
2. Pollen	Human Tissues	301	11	SMARTer	SRP041736	Pollen *et al.*, 2014 [121]
3. Usoskin	Mouse Brain	622	4	STRT-Seq	GSE59739	Usoskin *et al.*, 2015 [125]
4. Kolodziejczyk	Mouse Embryo Stem Cells	704	3	SMARTer	E-MTAB-2600	Kolodziejczyk *et al.*, 2015 [126]
5. Xin	Human Pancreas	1,600	8	SMARTer	GSE81608	Xin *et al.*, 2016 [128]
6. Muraro	Human Pancreas	2,126	10	CEL-Seq2	GSE85241	Muraro *et al.*, 2016 [130]
7. Klein	Mouse Embryo Stem Cells	2,717	4	inDrop	GSE65525	Klein *et al.*, 2015 [132]
8. Romanov	Mouse Brain	2,881	7	SMARTer	GSE74672	Romanov *et al.*, 2017 [133]
9. Zeisel	Mouse Brain	3,005	9	STRT-Seq	GSE60361	Zeisel *et al.*, 2015 [108]
10. Baron	Human Pancreas	8,569	14	inDrop	GSE84133	Baron *et al.*, 2016 [129]

For example, MAGIC has higher ARIs than the raw data for the Deng, Usoskin, Muraro, Klein, Romanov, and Baron but has lower ARIs in the remaining 4 datasets. scIRN is the only method able to improve the clustering performance compared to raw data in every dataset. Moreover, scIRN has the highest ARIs in all but Usoskin datasets. The average ARI of scIRN-imputed data is 0.77, which is higher than those obtained from raw data and data imputed by MAGIC, scImpute, SAVER, DrImpute (0.44, 0.41, 0.46, 0.43, 0.58, respectively).

For a more comprehensive analysis, we also report the assessment using normalized mutual information (NMI) and Jaccard index (JI) in Figures 7.3 and 7.4, respectively. Regardless of the assessment metrics, scIRN outperforms other methods by having the highest NMI (10/10 datasets) and JI (9/10 datasets) values. These results demonstrate that cluster analysis using scIRN-imputed data leads to a better accuracy than using the raw data or data imputed by other imputation methods.

7.2.2 scIRN improves transcriptome landscape visualization

In this section, we demonstrate that scIRN improves the visualization of the single-cell data. We used UMAP [62] to generate the transcriptome landscapes from raw and data imputed by MAGIC, scImpute, SAVER, DrImpute, and scIRN. We performed data visualization and calculated the silhouette index for each of the 10 datasets. Figure 7.5 shows the SI values obtained for the raw data and data imputed by the five imputation methods. The figure shows that scIRN can improve the quality of data visualization in all datasets. scIRN also has the highest SI in each of these datasets. These results demonstrate that data imputation using scIRN would lead to a much better visualization of transcriptome landscapes compared to using raw data or data imputed by other methods.

Figure 7.6 shows the transcriptome landscapes of the Usoskin dataset. Using

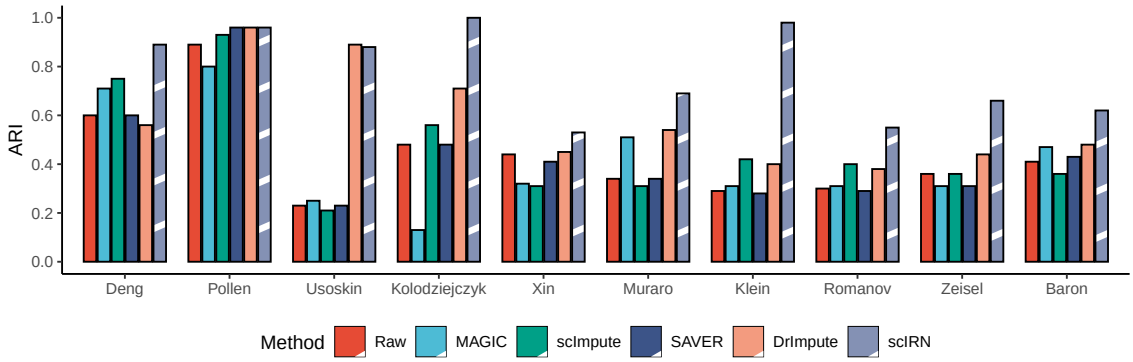

Figure 7.2: Adjusted Rand index (ARI) obtained from clustering on raw data and data imputed by MAGIC, SAVER, scImpute, DrImpute, and scIRN. The x-axis shows the names of the datasets while the y-axis shows ARI value of each method. scIRN outperforms other methods in all datasets except Usoskin.

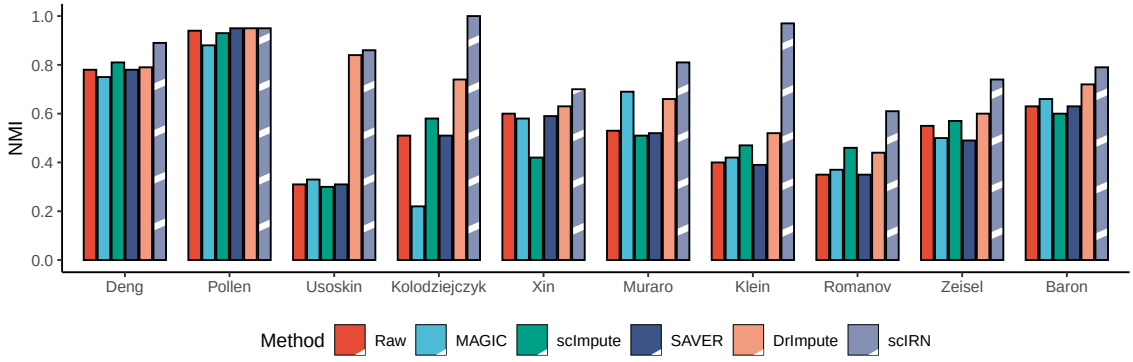

Figure 7.3: Normalized mutual information (NMI) obtained from clustering on raw data and data imputed by MAGIC, SAVER, scImpute, DrImpute, and scIRN. The x-axis shows the names of the datasets while the y-axis shows NMI value of each method. scIRN outperforms other methods in all datasets.

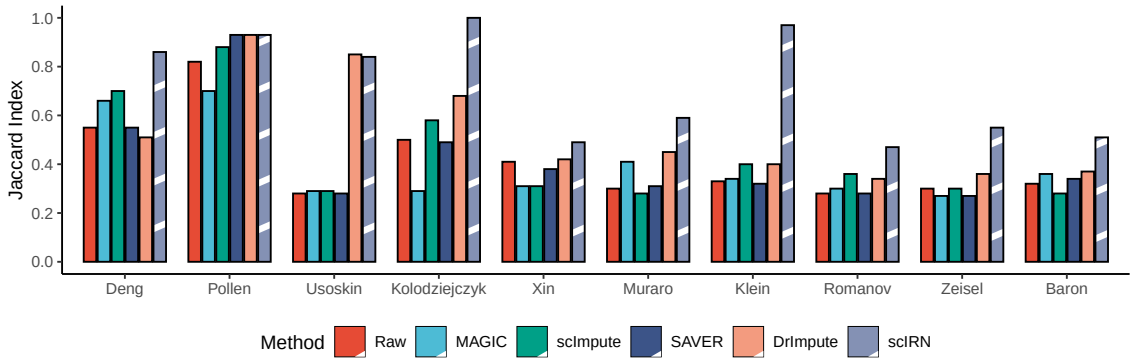

Figure 7.4: Jaccard index (JI) obtained from clustering on raw data and data imputed by MAGIC, SAVER, scImpute, DrImpute, and scIRN. The x-axis shows the names of the datasets while the y-axis shows JI value of each method. scIRN outperforms other methods in all datasets except Usoskin.

scIRN imputed data, UMAP was able to generate a clear representation, where cells from different groups are well-separated. When using data imputed by other methods, cells are usually mixed together. scIRN outperformed other imputation methods by having the highest SI value (0.67 compared to 0.28, -0.09, 0.14, 0.26, 0.5 of raw data, MAGIC, scImpute, SAVER, and DrImpute, respectively).

Figure 7.7 shows the transcriptome landscapes of the Klein dataset. The 2D representation of scIRN-imputed data is the only one that has four separable groups, corresponding to the four real cell types. The landscapes generated using raw and data imputed by other methods have different cell types mixed together. The data imputed by scIRN has the highest SI value (0.89 compared to 0.61 of the second best).

7.3 Conclusion (scIRN)

We introduce a new method, scIRN, to recover the missing data caused by dropout events in scRNA-seq. We assess the performance of our approach using 10 single-cell datasets in a comparison with four current state-of-the-art imputation methods.

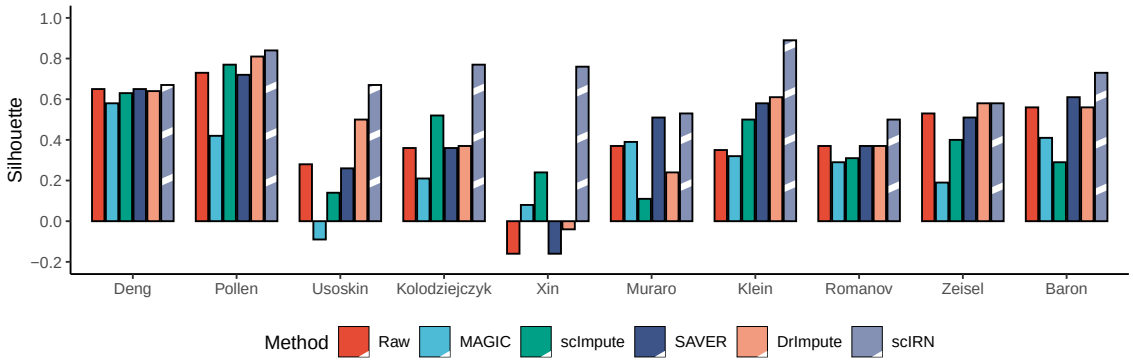

Figure 7.5: Visualization quality using raw and imputed data, measured by silhouette index (SI). The x-axis shows the names of the datasets while the y-axis shows SI value of each method. scIRN outperforms other methods in all datasets.

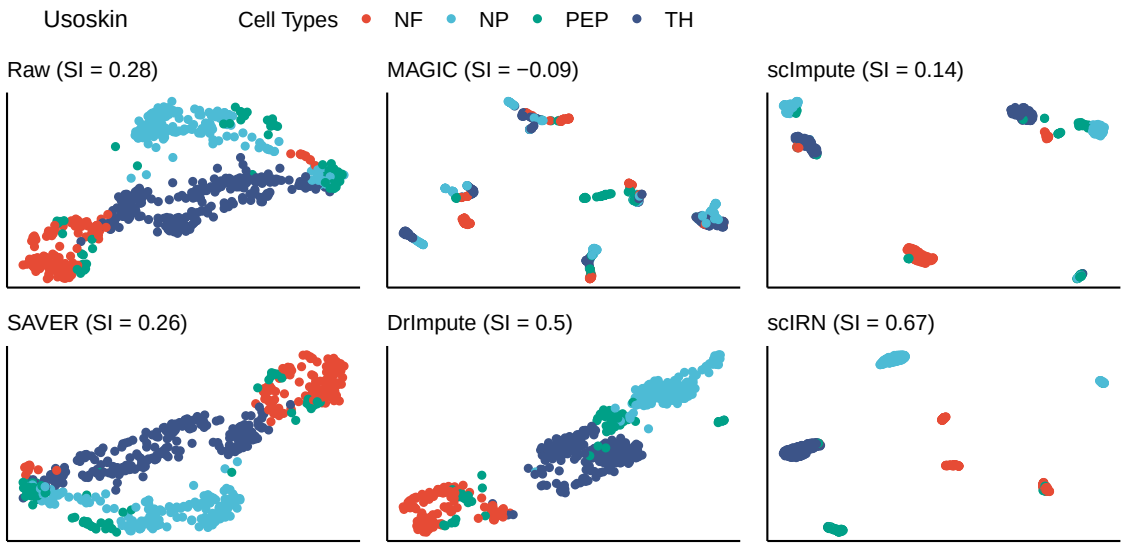

Figure 7.6: Transcriptome landscape of the Usoskin dataset. The scatter plot shows the first two principal components calculated by UMAP. Different colors represent different cell types. The 2D representation generated by scIRN has a clear structure, where cells from different groups are separated from one other.

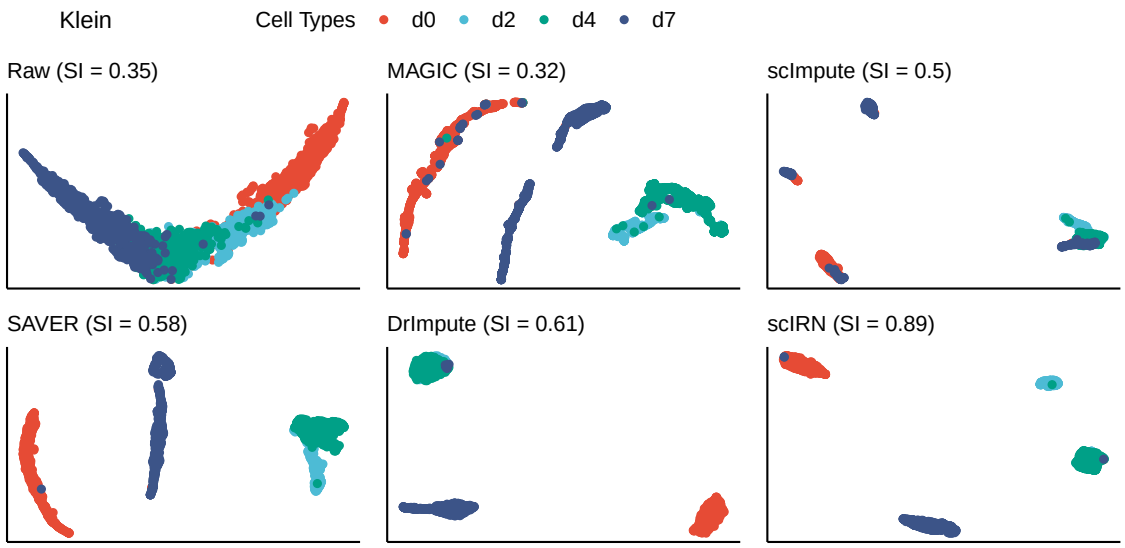

Figure 7.7: Transcriptomics landscape of the Klein dataset. The scatter plot shows the first two principal components calculated by UMAP for raw and imputed data. The 2D representation generated from scIRN has a clear structure, where cells from different groups are separate from each other.

Our analysis shows that scIRN outperforms existing approaches in improving the identification of cell sub-populations. scIRN also improves the quality of transcriptome landscapes generated by UMAP. A potential improvement of this research is to investigate the scalability of scIRN by analyzing datasets with higher number of cells. Another direction is to investigate the imputation method in other research applications, including pseudo-time trajectory inference and supervised learning.

Part III

Summary

Chapter 8

Conclusion

Advances in next generation sequencing techniques have produced a vast amount biological data from different modalities. However, studying biological systems using multiple levels of data, from tissue level with multi-omics to single-cell level with scRNA-seq, is still an ongoing challenge. To this end, we presented novel computational approaches to translate the high-dimensional, large-scale biological data to knowledge and insights of complex diseases.

First, we proposed a novel method for multi-omics data integration, disease subtyping, and risk assessment. Our method, called Subtyping via Consensus Factor Analysis (SCFA), aims to address the limitations of current integrative methods including their statistical assumption, and their sensitivity to noise. The contribution of SCFA is two-fold. First, it utilizes a robust dimension reduction procedure using autoencoder and factor analysis to retain only essential signals. Second, it allows researchers to predict risk scores of patients using multi-omics data – the attribute that is missing in current state-of-the-art subtyping methods. We validated our method by comparing it with current state-of-the-arts using data obtained from 7,973 patients related to 30 cancer diseases downloaded from The Cancer Genome Atlas (TCGA). We demonstrated that our method was able to exploiting the complementary sig-

nals available in different types of data in order to improve the accuracy of disease subtyping and risk prediction tasks.

Second, we introduced a powerful deep learning-based framework for scRNA-seq data analysis, called called single-cell Decomposition using Hierarchical Autoencoder (scDHA). The method aims to address the computation challenges of single-cell data analysis including the exponentially increasing in size of scRNA-seq dataset and technical noise. The scDHA framework includes two main modules. The first module is a non-negative kernel autoencoder that is capable of filtering out the noisy features and improving the quality of the data. The second module is a Stacked Bayesian Autoencoder that is built upon the Variational Autoencoder [112] (VAE) to project the data onto a low-dimensional space. The low-dimensional representation has much lower number of features than the original data, while still retaining most of the information in the original data. We demonstrated that using this low-dimensional representation would improve both accuracy and scalability of single-cell data analysis. For our evaluation, we compared scDHA against state-of-the-arts using 34 real scRNA-seq datasets in four different research sub-fields including *de novo* clustering of cells, visualizing the transcriptome landscape, classifying cells, and inferring pseudo-time. We showed that scDHA outperforms other methods in each sub-fields by having significantly higher accuracy and lower time complexity.

Third, we proposed a new method, name single-cell Imputation via Subspace Regression (scISR), to mitigate the effects of dropout events that frequently happen during the sequencing process of individual cells. The contribution of scISR is two-fold. First, by introducing a hypothesis testing procedure, we avoid altering true zero values. Second, the subspace regression provides a more accurate imputation by limiting the imputation to gene groups with similar expression patterns. We compared our approach with state-of-the-art methods using 25 real scRNA-seq datasets

and 46 simulated datasets. We demonstrated that scISR outperforms other imputation methods in improving the quality of clustering analysis. At the same time, we also demonstrated that scISR preserves the transcriptome landscape of each dataset. Lastly, we showed that scISR is robust against different dropout rates and distributions. Moreover, because scISR is capable of improving data quality without filtering out features from data, it can be seamlessly incorporated into other single-cell analysis pipelines.

Finally, we introduced two new methods, single-cell Imputation using Neural Network (scINN) and single-cell Imputation using Residual Network (scIRN), to recover the missing data caused by dropout events in scRNA-seq data. We compared our methods with four state-of-the-art imputation methods using 10 scRNA-seq datasets. Both methods outperformed existing approaches in improving the identification of cell sub-populations. They also improved the quality of transcriptome landscapes generated by UMAP. A potential improvement of this research is to investigate the scalability of scINN by analyzing datasets with higher number of cells. Another direction is to investigate the imputation method in other research applications, including pseudo-time trajectory inference and supervised learning.

Chapter 9

Future Research

For future work, I plan to modify the proposed methods so that they can be applied in conjunction with other analysis methods that the colleagues in my current research laboratory is developing, including gene networks [65, 181–192], meta-analysis [193–198], cancer subtyping [44, 45, 199–211], single-cell analysis [158, 176, 178, 212–218], and other important research areas [219–232]. There are also two immediate directions that can be investigated to improve the accuracy of the developed techniques:

- Early multi-omics data integration: The current approach used in SCFA includes analyzing and subtyping each data type individually, then generating consensus subtypes from these results. This approach is categorized as a late stage data integration approach. While SCFA is capable of producing significantly different subtypes, the limitation of this approach is that it does not take into consideration the direct interactions between different data types within in the cell, e.g., miRNA can prevent protein translation of mRNA, or methylation of DNA can deactivate a gene transcription. We will work on developing a method that can generate a single low-dimensional representation of multiple data types. Because this representation is not only expected to contain the

condensed information from different data types, but also their interactions, the subtyping result using the representation should be more accurate and clinically relevant.

- Improvement of autoencoder-based scRNA-seq analysis framework: The scDHA framework is built upon the Variational Autoencoder, which assumes that the latent space is normally distributed. Although the current framework is able to remove noise from the original data and generate an informative low-dimensional representation, the latent space with normal distribution might not be the best choice to represent the original data. We will investigate the performance of scDHA framework with other latent distributions that could be more suitable for single-cell data including Poisson, Negative Binomial, and Zero-Inflated Negative Binomial distributions, etc. We can also replace Variational Autoencoder with a more recent architecture such as Generative Adversarial Network [233]. Finally, we can extend the scDHA framework to be able to utilize the available single-cell data to a new dataset through transfer learning.

For the longer term, I plan to further improve the proposed techniques by improving the based computational model using more advanced techniques. By continuously refining and expanding these computational tools, the scientific community can harness the full potential of biological data and accelerate the discovery of novel biological insights, ultimately benefiting human health and well-being.

References

[1] Laura J Esserman, Ian M Thompson, Brian Reid, Peter Nelson, David F Ransohoff, H Gilbert Welch, Shelley Hwang, Donald A Berry, Kenneth W Kinzler, William C Black, Mina Bissell, Howard Parnes, and Sudhir Srivastava. Addressing overdiagnosis and overtreatment in cancer: a prescription for change. *The Lancet Oncology*, 15(6):e234–e242, 2014.

[2] Laura Esserman, Yiwey Shieh, and Ian Thompson. Rethinking screening for breast cancer and prostate cancer. *Journal of the American Medical Association*, 302(15):1685–1692, 2009.

[3] Ahmedin Jemal, Rebecca Siegel, Elizabeth Ward, Yongping Hao, Jiaquan Xu, and Michael J Thun. Cancer statistics, 2009. *CA: A Cancer Journal for Clinicians*, 59(4):225–249, 2009.

[4] Herbert Seidman, Margaret H Mushinski, Steven K Gelb, and Edwin Silverberg. Probabilities of eventually developing or dying of cancer—United States, 1985. *CA: A Cancer Journal for Clinicians*, 35(1):36–56, 1985.

[5] Hidetaka Uramoto and Fumihiro Tanaka. Recurrence after surgery in patients with NSCLC. *Translational Lung Cancer Research*, 3(4):242–249, 2014.

[6] Christopher M. Booth and Frances A. Shepherd. Adjuvant chemotherapy for

resected non-small cell lung cancer. *Journal of Thoracic Oncology*, 1(2):180–187, 2006.

[7] NSCLC Meta-analysis Collaborative Group. Preoperative chemotherapy for non-small-cell lung cancer: a systematic review and meta-analysis of individual participant data. *The Lancet*, 383(9928):1561–1571, 2014.

[8] Enriqueta Felip, Rafael Rosell, José Antonio Maestre, José Manuel Rodriguez-Paniagua, Teresa Morán, Julio Astudillo, Guillermo Alonso, José Manuel Borro, José Luis González-Larriba, Antonio Torres, Carlos Camps, Ricardo Guijarro, Dolores Isla, Rafael Aguiló, Vicente Alberola, José Padilla, Abel Sánchez-Palencia, José Javier Sánchez, Eduardo Hermosilla, and Bartomeu Massuti. Preoperative chemotherapy plus surgery versus surgery plus adjuvant chemotherapy versus surgery alone in early-stage non-small-cell lung cancer. *Journal of Clinical Oncology*, 28(19):3138–3145, 2010.

[9] Eric A Collisson, Peter Bailey, David K Chang, and Andrew V Biankin. Molecular subtypes of pancreatic cancer. *Nature Reviews Gastroenterology & Hepatology*, 16(4):207–220, 2019.

[10] Rodrigo Dienstmann, Louis Vermeulen, Justin Guinney, Scott Kopetz, Sabine Tejpar, and Josep Tabernero. Consensus molecular subtypes and the evolution of precision medicine in colorectal cancer. *Nature Reviews Cancer*, 17:79–92, 2017.

[11] Clinton Yam, Sendurai A Mani, and Stacy L Moulder. Targeting the molecular subtypes of triple negative breast cancer: understanding the diversity to progress the field. *The Oncologist*, 22(9):1086–1093, 2017.

[12] Brian D Lehmann, Joshua A Bauer, Xi Chen, Melinda E Sanders, A Bapsi

Chakravarthy, Yu Shyr, and Jennifer A Pietenpol. Identification of human triple-negative breast cancer subtypes and preclinical models for selection of targeted therapies. *The Journal of Clinical Investigation*, 121(7):2750–2767, 2011.

[13] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9): 1464–1480, 1990.

[14] Pablo Tamayo, Donna Slonim, Jill Mesirov, Qing Zhu, Sutisak Kitareewan, Ethan Dmitrovsky, Eric S Lander, and Todd R Golub. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences*, 96(6):2907–2912, 1999.

[15] Todd R Golub, Donna K Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P Mesirov, Hilary Coller, Mignon L Loh, James R Downing, Mark A Caligiuri, Clara D Bloomfield, and Eric S Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[16] Javier Herrero, Alfonso Valencia, and Joaquın Dopazo. A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics*, 17(2):126–136, 2001.

[17] Feng Luo, Latifur Khan, Farokh Bastani, I-Ling Yen, and Jizhong Zhou. A dynamically growing self-organizing tree (DGSOT) for hierarchical clustering gene expression profiles. *Bioinformatics*, 20(16):2605–2617, 2004.

[18] Geoffrey J. McLachlan, RW Bean, and David Peel. A mixture model-based

approach to the clustering of microarray expression data. *Bioinformatics*, 18 (3):413–422, 2002.

[19] Debashis Ghosh and Arul M Chinnaiyan. Mixture modelling of gene expression data from microarray experiments. *Bioinformatics*, 18(2):275–286, 2002.

[20] Daxin Jiang, Chun Tang, and Aidong Zhang. Cluster analysis for gene expression data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1370–1386, 2004.

[21] Jean-Philippe Brunet, Pablo Tamayo, Todd R. Golub, and Jill P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences*, 101(12):4164–4169, 2004.

[22] Yuan Gao and George Church. Improving molecular cancer class discovery through sparse non-negative matrix factorization. *Bioinformatics*, 21(21):3970–3975, 2005.

[23] Roded Sharan and Ron Shamir. CLICK: a clustering algorithm with applications to gene expression analysis. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, volume 8, pages 306–316, 2000.

[24] Erez Hartuv and Ron Shamir. A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76(4):175–181, 2000.

[25] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3-4):281–297, 1999.

[26] Matthew D. Wilkerson and D. Neil Hayes. ConsensusClusterPlus: a class discovery tool with confidence assessments and item tracking. *Bioinformatics*, 26 (12):1572–1573, 2010.

[27] Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1-2):91–118, 2003.

[28] Sandrine Dudoit and Jane Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, 3(7):1–21, 2002.

[29] Asa Ben-Hur, Andre Elisseeff, and Isabelle Guyon. A stability based method for discovering structure in clustered data. In *Pacific Symposium on Biocomputing*, volume 7, pages 6–17, 2001.

[30] George C Tseng and Wing H Wong. Tight clustering: a resampling-based approach for identifying stable and tight patterns in data. *Biometrics*, 61(1): 10–16, 2005.

[31] Shihua Zhang, Chun-Chi Liu, Wenyuan Li, Hui Shen, Peter W. Laird, and Xianghong Jasmine Zhou. Discovery of multi-dimensional modules by integrative analysis of cancer genomic data. *Nucleic Acids Research*, 40(19):9379–9391, 2012.

[32] Prabhakar Chalise and Brooke L. Fridley. Integrative clustering of multi-level 'omic data based on non-negative matrix factorization algorithm. *PLOS ONE*, 12(5):e0176278, 2017.

[33] Dingming Wu, Dongfang Wang, Michael Q. Zhang, and Jin Gu. Fast dimension reduction and integrative clustering of multi-omics data using low-rank approximation: application to cancer molecular classification. *BMC Genomics*, 16(1): 1022, 2015.

[34] Eric F. Lock and David B. Dunson. Bayesian consensus clustering. *Bioinformatics*, 29(20):2610–2616, 2013.

[35] Paul Kirk, Jim E. Griffin, Richard S. Savage, Zoubin Ghahramani, and David L. Wild. Bayesian correlated clustering to integrate multiple datasets. *Bioinformatics*, 28(24):3290–3297, 2012.

[36] Qianxing Mo, Ronglai Shen, Cui Guo, Marina Vannucci, Keith S. Chan, and Susan G. Hilsenbeck. A fully Bayesian latent variable model for integrative clustering analysis of multi-type omics data. *Biostatistics*, 19(1):71–86, 2018.

[37] Qianxing Mo, Sijian Wang, Venkatraman E. Seshan, Adam B. Olshen, Nikolaus Schultz, Chris Sander, R. Scott Powers, Marc Ladanyi, and Ronglai Shen. Pattern discovery and cancer gene identification in integrated cancer genomic data. *Proceedings of the National Academy of Sciences*, 110(11):4245–4250, 2013.

[38] Ronglai Shen, Adam B Olshen, and Marc Ladanyi. Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, 25(22):2906–2912, 2009.

[39] Ronglai Shen, Qianxing Mo, Nikolaus Schultz, Venkatraman E Seshan, Adam B Olshen, Jason Huse, Marc Ladanyi, and Chris Sander. Integrative subtype discovery in glioblastoma using iCluster. *PLoS ONE*, 7(4):e35236, 2012.

[40] Bo Wang, Aziz M. Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods*, 11(3): 333–337, 2014.

[41] Nora K. Speicher and Nico Pfeifer. Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery. *Bioinformatics*, 31(12):i268–i275, 2015.

[42] Nimrod Rappoport and Ron Shamir. NEMO: Cancer subtyping by integration of partial multi-omic data. *Bioinformatics*, 35(18):3348–3356, 2019.

[43] Daniele Ramazzotti, Avantika Lal, Bo Wang, Serafim Batzoglou, and Arend Sidow. Multi-omic tumor data reveal diversity of molecular mechanisms that correlate with survival. *Nature Communications*, 9:4453, 2018.

[44] Tin Nguyen, Rebecca Tagett, Diana Diaz, and Sorin Draghici. A novel approach for data integration and disease subtyping. *Genome Research*, 27:2025–2039, 2017.

[45] Hung Nguyen, Sangam Shrestha, Sorin Draghici, and Tin Nguyen. PINSPlus: A tool for tumor subtype discovery in integrated genomic data. *Bioinformatics*, 35(16):2843–2846, 2019.

[46] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[47] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1957.

[48] Harry H. Harman and Wayne H. Jones. Factor analysis by minimizing residuals (minres). *Psychometrika*, 31(3):351–368, Sep 1966. ISSN 1860-0980. doi: 10.1007/BF02289468. URL `https://doi.org/10.1007/BF02289468`.

[49] Cheng-Hsien Li. The performance of ML, DWLS, and ULS estimation with robust corrections in structural equation models with ordinal variables. *Psy-*

chological Methods, 21(3):369–387, 2016. ISSN 1939-1463(Electronic),1082-989X(Print).

[50] Shibiao Wan, Junil Kim, and Kyoung Jae Won. SHARP: hyperfast and accurate processing of single-cell RNA-seq data via ensemble random projection. *Genome Research*, 30(2):205–213, 2020.

[51] Hui Zou and Trevor Hastie. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(2):301–320, 2005. ISSN 13697412, 14679868.

[52] Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for Cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13, 2011.

[53] Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):457–481, 1958.

[54] Atul B. Shinagare, Raghu Vikram, Carl Jaffe, Oguz Akin, Justin Kirby, Erich Huang, John Freymann, Nisha I. Sainani, Cheryl A. Sadow, Tharakeswara K. Bathala, Daniel L. Rubin, Aytekin Oto, Matthew T. Heller, Venkateswar R. Surabhi, Venkat Katabathina, and Stuart G. Silverman. Radiogenomics of clear cell renal cell carcinoma: preliminary findings of The Cancer Genome Atlas–Renal Cell Carcinoma (TCGA–RCC) Imaging Research Group. *Abdominal imaging*, 40(6):1684–1692, 2015.

[55] George V Thomas, Chris Tran, Ingo K Mellinghoff, Derek S Welsbie, Emily Chan, Barbara Fueger, Johannes Czernin, and Charles L Sawyers. Hypoxia-

inducible factor determines sensitivity to inhibitors of mtor in kidney cancer. *Nature medicine*, 12(1):122–127, 2006.

[56] Ignacio Varela, Patrick Tarpey, Keiran Raine, Dachuan Huang, Choon Kiat Ong, Philip Stephens, Helen Davies, David Jones, Meng-Lay Lin, Jon Teague, Graham Bignell, Adam Butler, Juok Cho, Gillian L. Dalgliesh, Danushka Galappaththige, Chris Greenman, Claire Hardy, Mingming Jia, Calli Latimer, King Wai Lau, John Marshall, Stuart McLaren, Andrew Menzies, Laura Mudie, Lucy Stebbings, David A. Largaespada, L. F. A. Wessels, Stephane Richard, Richard J. Kahnoski, John Anema, David A.Tuveson, Pedro A. Perez-Mancera, Ville Mustonen, Andrej Fischer, David J. Adams, Alistair Rust, Waraporn Chan-on, Chutima Subimerb, Karl Dykema, Kyle Furge, Peter J. Campbell, Bin Tean Teh, Michael R. Stratton, and P. Andrew Futreal. Exome sequencing identifies frequent mutation of the swi/snf complex gene pbrm1 in renal carcinoma. *Nature*, 469(7331):539–542, 2011.

[57] Frank E. Harrell, Robert M. Califf, David B. Pryor, Kerry L. Lee, and Robert A. Rosati. Evaluating the yield of medical tests. *Journal of the American Medical Association*, 247(18):2543–2546, 1982.

[58] Antoine-Emmanuel Saliba, Alexander J Westermann, Stanislaw A Gorski, and Jörg Vogel. Single-cell RNA-seq: advances and future challenges. *Nucleic Acids Research*, 42(14):8845–8860, 2014.

[59] C Wyatt Shields IV, Catherine D Reyes, and Gabriel P López. Microfluidic cell sorting: a review of the advances in the separation of cells from debulking to rare cell isolation. *Lab on a Chip*, 15(5):1230–1249, 2015.

[60] Kristofer Davie, Jasper Janssens, Duygu Koldere, Maxime De Waegeneer, Uli Pech, Łukasz Kreft, Sara Aibar, Samira Makhzami, Valerie Christiaens, Car-

men Bravo González-Blas, Suresh Poovathingal, Gert Hulselmans, Katina I. Spanier, Thomas Moerman, Bram Vanspauwen, Sarah Geurs, Thierry Voet, Jeroen Lammertyn, Bernard Thienpont, Sha Liu, Nikos Konstantinides, Mark Fiers, Patrik Verstreken, and Stein Aerts. A Single-Cell Transcriptome Atlas of the Aging Drosophila Brain. *Cell*, 174(4):982–998, 2018.

[61] Orit Rozenblatt-Rosen, Michael JT Stubbington, Aviv Regev, and Sarah A Teichmann. The Human Cell Atlas: From vision to reality. *Nature*, 550(7677): 451–453, 2017.

[62] Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 37:38–44, 2019.

[63] Yvan Saeys, Sofie Van Gassen, and Bart N Lambrecht. Computational flow cytometry: helping to make sense of high-dimensional immunology data. *Nature Reviews Immunology*, 16:449–462, 2016.

[64] Kelly Street, Davide Risso, Russell B Fletcher, Diya Das, John Ngai, Nir Yosef, Elizabeth Purdom, and Sandrine Dudoit. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics*, 19:477, 2018.

[65] Hung Nguyen, Duc Tran, Bang Tran, Bahadir Pehlivan, and Tin Nguyen. A comprehensive survey of regulatory network inference methods using single-cell RNA sequencing data. *Briefings in Bioinformatics*, 22(3):1–15, 2021.

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

[67] Aaron M. Newman, Chloé B. Steen, Chih Long Liu, Andrew J. Gentles, Aadel A. Chaudhuri, Florian Scherer, Michael S. Khodadoust, Mohammad S. Esfahani, Bogdan A. Luca, David Steiner, Maximilian Diehn, and Ash A. Alizadeh. Determining cell type abundance and expression from bulk tissues with digital cytometry. *Nature Biotechnology*, 37(7):773–782, 2019. ISSN 1546-1696.

[68] Rahul Satija, Jeffrey A. Farrell, David Gennert, Alexander F. Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33:495–502, 2015.

[69] Lambda Moses and Lior Pachter. Museum of spatial transcriptomics. *Nature methods*, pages 1–13, 2022.

[70] Yong Wang and Nicholas E Navin. Advances and applications of single-cell sequencing technologies. *Molecular Cell*, 58(4):598–609, 2015.

[71] Amit Zeisel, Ana B Muñoz-Manchado, Simone Codeluppi, Peter Lönnerberg, Gioele La Manno, Anna Juréus, Sueli Marques, Hermany Munguba, Liqun He, Christer Betsholtz, Charlotte Rolny, Gonçalo Castelo-Branco, Jens Hjerling-Leffler, and Sten Linnarsson. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, 347(6226):1138–1142, 2015.

[72] Sophia K Longo, Margaret G Guo, Andrew L Ji, and Paul A Khavari. Integrating single-cell and spatial transcriptomics to elucidate intercellular tissue dynamics. *Nature Reviews Genetics*, 22(10):627–644, 2021.

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

[74] Vladimir Yu Kiselev, Kristina Kirschner, Michael T Schaub, Tallulah Andrews, Andrew Yiu, Tamir Chandra, Kedar N Natarajan, Wolf Reik, Mauricio Barahona, Anthony R Green, and Martin Hamberg. SC3: consensus clustering of single-cell RNA-seq data. *Nature Methods*, 14:483–486, 2017.

[75] Minzhe Guo, Hui Wang, S Steven Potter, Jeffrey A Whitsett, and Yan Xu. SINCERA: a pipeline for single-cell RNA-seq profiling analysis. *PLoS Computational Biology*, 11(11):e1004575, 2015.

[76] Peijie Lin, Michael Troup, and Joshua W. K. Ho. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biology*, 18(1):59, 2017.

[77] F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19:15, 2018.

[78] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[79] Ronald R Coifman, Stephane Lafon, Ann B Lee, Mauro Maggioni, Boaz Nadler, Frederick Warner, and Steven W Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences*, 102(21):7426–7431, 2005.

[80] El-ad David Amir, Kara L Davis, Michelle D Tadmor, Erin F Simonds, Jacob H Levine, Sean C Bendall, Daniel K Shenfeld, Smita Krishnaswamy, Garry P Nolan, and Dana Pe'er. viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nature Biotechnology*, 31(6):545, 2013.

[81] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

[82] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. URL http://citeseer.ist.psu.edu/breiman01random.html.

[83] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521: 436–444, 2015.

[84] Jerome H. Friedman. Greedy function approximation: A Gradient Boosting Machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[85] Wouter Saelens, Robrecht Cannoodt, Helena Todorov, and Yvan Saeys. A comparison of single-cell trajectory inference methods. *Nature Biotechnology*, 37:547–554, 2019.

[86] Amos Tanay and Aviv Regev. Scaling single-cell genomics from phenomenology to mechanism. *Nature*, 541(7637):331–338, 2017.

[87] Martin Etzrodt, Max Endele, and Timm Schroeder. Quantitative single-cell approaches to stem cell research. *Cell Stem Cell*, 15(5):546–558, 2014.

[88] Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J Lennon, Kenneth J Livak, Tarjei S Mikkelsen, and John L Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology*, 32:381–386, 2014.

[89] Zhicheng Ji and Hongkai Ji. TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic Acids Research*, 44(13):e117, 2016.

[90] Devon A. Lawson, Kai Kessenbrock, Ryan T. Davis, Nicholas Pervolarakis, and Zena Werb. Tumour heterogeneity and metastasis at single-cell resolution. *Nature Cell Biology*, 20(12):1349–1360, 2018.

[91] Karlynn E. Neu, Qingming Tang, Patrick C. Wilson, and Aly A. Khan. Single-cell genomics: Approaches and utility in immunology. *Trends in Immunology*, 38(2):140–149, 2017.

[92] Wu Liu, Hongzhang He, and Si-Yang Zheng. Microfluidics in single-cell virology: Technologies and applications. *Trends in Biotechnology*, 38(12):1360 – 1372, 2020.

[93] Valentine Svensson, Roser Vento-Tormo, and Sarah A Teichmann. Exponential scaling of single-cell RNA-seq in the past decade. *Nature Protocols*, 13(4):599–604, 2018. ISSN 1750-2799.

[94] Philip Brennecke, Simon Anders, Jong Kyoung Kim, Aleksandra A Kolodziejczyk, Xiuwei Zhang, Valentina Proserpio, Bianka Baying, Vladimir Benes, Sarah A Teichmann, John C Marioni, and Marcus G Heisler. Accounting for technical noise in single-cell RNA-seq experiments. *Nature Methods*, 10 (11):1093–1095, 2013.

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

[96] Justina žurauskienė and Christopher Yau. pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics*, 17(1):1–11, 2016.

[97] Bo Wang, Junjie Zhu, Emma Pierson, Daniele Ramazzotti, and Serafim Batzoglou. Visualization and analysis of single-cell rna-seq data by kernel-based similarity learning. *Nature methods*, 14(4):414–416, 2017.

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

[99] Tian Tian, Ji Wan, Qi Song, and Zhi Wei. Clustering single-cell RNA-seq data with a model-based deep learning approach. *Nature Machine Intelligence*, 1(4): 191–198, 2019.

[100] Bin Yu, Chen Chen, Ren Qi, Ruiqing Zheng, Patrick J Skillman-Lawrence, Xiaolin Wang, Anjun Ma, and Haiming Gu. scgmai: a gaussian mixture model for clustering single-cell rna-seq data based on deep autoencoder. *Briefings in Bioinformatics*, 22(4):bbaa316, 2021.

[101] Peter V Kharchenko, Lev Silberstein, and David T Scadden. Bayesian approach to single-cell differential expression analysis. *Nature Methods*, 11(7):740–742, 2014.

[102] Simone Rizzetto, Auda A Eltahla, Peijie Lin, Rowena Bull, Andrew R Lloyd, Joshua WK Ho, Vanessa Venturi, and Fabio Luciani. Impact of sequencing depth and read length on single cell RNA sequencing data of T cells. *Scientific Reports*, 7:12781, 2017.

[103] Swati Parekh, Christoph Ziegenhain, Beate Vieth, Wolfgang Enard, and Ines Hellmann. The impact of amplification on differential expression analyses by RNA-seq. *Scientific Reports*, 6:25533, 2016.

[104] David Van Dijk, Roshan Sharma, Juozas Nainys, Kristina Yim, Pooja Kathail, Ambrose J Carr, Cassandra Burdziak, Kevin R Moon, Christine L Chaffer, Diwakar Pattabiraman, Brian Bierie, Linas Mazutis, Guy Wolf, Smita Krishnaswamy, and Dana Pe'er. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729, 2018.

[105] Wei Vivian Li and Jingyi Jessica Li. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nature Communications*, 9:997, 2018.

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

[107] Wuming Gong, Il-Youp Kwak, Pruthvi Pota, Naoko Koyano-Nakagawa, and Daniel J Garry. DrImpute: imputing dropout events in single cell RNA sequencing data. *BMC Bioinformatics*, 19:220, 2018.

[108] A. Zeisel, A. B. Munoz-Manchado, S. Codeluppi, P. Lonnerberg, G. La Manno, A. Jureus, S. Marques, H. Munguba, L. He, C. Betsholtz, C. Rolny, G. Castelo-Branco, J. Hjerling-Leffler, and S. Linnarsson. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, 347(6226):1138–1142, 2015.

[109] Anoop P. Patel, Itay Tirosh, John J. Trombetta, Alex K. Shalek, Shawn M. Gillespie, Hiroaki Wakimoto, Daniel P. Cahill, Brian V. Nahed, William T.

Curry, Robert L. Martuza, David N. Louis, Orit Rozenblatt-Rosen, Mario L. Suvà, Aviv Regev, and Bradley E. Bernstein. Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, 344(6190):1396–1401, 2014. ISSN 0036-8075. doi: 10.1126/science.1254257.

[110] Nils Eling, Michael D Morgan, and John C. Marioni. Challenges in measuring and understanding biological noise. *Nature Reviews Genetics*, 20(1):536–548, 2019.

[111] Oliver Stegle, Sarah A Teichmann, and John C Marioni. Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, 16(3):133–145, 2015.

[112] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114*, 2013.

[113] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[114] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 6105–6114, Long Beach, California, USA, 2019.

[115] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[116] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. *arXiv:1602.02282 [cs, stat]*, 2016.

[117] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*, pages 971–980, 2017.

[118] Liying Yan, Mingyu Yang, Hongshan Guo, Lu Yang, Jun Wu, Rong Li, Ping Liu, Ying Lian, Xiaoying Zheng, Jie Yan, Jin Huang, Ming Li, Xinglong Wu, Lu Wen, Kaiqin Lao, Ruiqiang Li, Jie Qiao, and Fuchou Tang. Single-cell RNA-seq profiling of human preimplantation embryos and embryonic stem cells. *Nature Structural & Molecular Biology*, 20:1131–1139, 2013.

[119] Mubeen Goolam, Antonio Scialdone, Sarah JL Graham, Iain C Macaulay, Agnieszka Jedrusik, Anna Hupalowska, Thierry Voet, John C Marioni, and Magdalena Zernicka-Goetz. Heterogeneity in Oct4 and Sox2 targets biases cell fate in 4-cell mouse embryos. *Cell*, 165(1):61–74, 2016.

[120] Qiaolin Deng, Daniel Ramsköld, Björn Reinius, and Rickard Sandberg. Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science*, 343(6167):193–196, 2014.

[121] Alex A. Pollen, Tomasz J. Nowakowski, Joe Shuga, Xiaohui Wang, Anne A. Leyrat, Jan H. Lui, Nianzhen Li, Lukasz Szpankowski, Brian Fowler, Peilin Chen, Naveen Ramalingam, Gang Sun, Myo Thu, Michael Norris, Ronald Lebofsky, Dominique Toppani, Darnell W. Kemp Ii, Michael Wong, Barry Clerkson, Brittnee N. Jones, Shiquan Wu, Lawrence Knutsson, Beatriz Alvarado, Jing Wang, Lesley S. Weaver, Andrew P. May, Robert C. Jones, Marc A. Unger, Arnold R. Kriegstein, and Jay A. A. West. Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nature Biotechnology*, 32:1053–1058, 2014.

[122] Yue J. Wang, Jonathan Schug, Kyoung-Jae Won, Chengyang Liu, Ali Naji,

Dana Avrahami, Maria L. Golson, and Klaus H. Kaestner. Single-cell transcriptomics of the human endocrine pancreas. *Diabetes*, 65(10):3028–3038, 2016.

[123] Spyros Darmanis, Steven A Sloan, Ye Zhang, Martin Enge, Christine Caneda, Lawrence M Shuer, Melanie G Hayden Gephart, Ben A Barres, and Stephen R Quake. A survey of human brain transcriptome diversity at the single cell level. *Proceedings of the National Academy of Sciences of the United States of America*, 112(23):7285–7290, 2015.

[124] J Gray Camp, Farhath Badsha, Marta Florio, Sabina Kanton, Tobias Gerber, Michaela Wilsch-Bräuninger, Eric Lewitus, Alex Sykes, Wulf Hevers, Madeline Lancaster, Juergen A Knoblich, Robert Lachmann, Svante Pääbo, Wieland Huttner, and Barbara Treutlein. Human cerebral organoids recapitulate gene expression programs of fetal neocortex development. *Proceedings of the National Academy of Sciences of the United States of America*, 112(51):15672–15677, 2015.

[125] Dmitry Usoskin, Alessandro Furlan, Saiful Islam, Hind Abdo, Peter Lönnerberg, Daohua Lou, Jens Hjerling-Leffler, Jesper Haeggström, Olga Kharchenko, Peter V Kharchenko, Sten Linnarson, and Patrik Ernfors. Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing. *Nature Neuroscience*, 18:145–153, 2015.

[126] Aleksandra A. Kolodziejczyk, Jong Kyoung Kim, Jason C.H. Tsang, Tomislav Ilicic, Johan Henriksson, Kedar N. Natarajan, Alex C. Tuck, Xuefei Gao, Marc Bühler, Pentao Liu, John C. Marioni, and Sarah A. Teichmann. Single cell RNA-sequencing of pluripotent states unlocks modular transcriptional variation. *Cell Stem Cell*, 17(4):471–485, 2015.

[127] J. Gray Camp, Keisuke Sekine, Tobias Gerber, Henry Loeffler-Wirth, Hans

Binder, Malgorzata Gac, Sabina Kanton, Jorge Kageyama, Georg Damm, Daniel Seehofer, Lenka Belicova, Marc Bickle, Rico Barsacchi, Ryo Okuda, Emi Yoshizawa, Masaki Kimura, Hiroaki Ayabe, Hideki Taniguchi, Takanori Takebe, and Barbara Treutlein. Multilineage communication regulates human liver bud development from pluripotency. *Nature*, 546(7659):533–538, 2017.

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

[129] Maayan Baron, Adrian Veres, Samuel L. Wolock, Aubrey L. Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K. Wagner, Shai S. Shen-Orr, Allon M. Klein, Douglas A. Melton, and Itai Yanai. A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell Systems*, 3(4):346–360, 2016.

[130] Mauro J. Muraro, Gitanjali Dharmadhikari, Dominic Grün, Nathalie Groen, Tim Dielen, Erik Jansen, Leon van Gurp, Marten A. Engelse, Francoise Carlotti, Eelco J.P. de Koning, and Alexander van Oudenaarden. A Single-Cell Transcriptome Atlas of the Human Pancreas. *Cell Systems*, 3(4):385–394.e3, 2016. ISSN 2405-4712.

[131] Åsa Segerstolpe, Athanasia Palasantza, Pernilla Eliasson, Eva-Marie Andersson, Anne-Christine Andréasson, Xiaoyan Sun, Simone Picelli, Alan Sabirsh, Maryam Clausen, Magnus K. Bjursell, David M. Smith, Maria Kasper, Carina Ämmälä, and Rickard Sandberg. Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. *Cell Metabolism*, 24(4):593–607, 2016.

[132] Allon M Klein, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A Weitz, and Marc W Kirschner. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201, 2015.

[133] Roman A Romanov, Amit Zeisel, Joanne Bakker, Fatima Girach, Arash Hellysaz, Raju Tomer, Alán Alpár, Jan Mulder, Frédéric Clotman, Erik Keimpema, Brian Hsueh, Ailey K Crow, Henrik Martens, Christian Schwindling, Daniela Calvigioni, Jaideep S Bains, Zoltán Máté, Gábor Szabó, Yuchio Yanagawa, Ming-Dong Zhang, Andre Rendeiro, Matthias Farlik, Mathias Uhlén, Peer Wulff, Christop Bock, Christian Broberger, Karl Deisseroth, Tomas Hökfelt, Sten Linnarsson, Tamas L Horvath, and Tibor Harkany. Molecular interrogation of hypothalamic organization reveals distinct dopamine neuronal subtypes. *Nature Neuroscience*, 20(2):176–188, 2017.

[134] Blue B Lake, Rizi Ai, Gwendolyn E Kaeser, Neeraj S Salathia, Yun C Yung, Rui Liu, Andre Wildberg, Derek Gao, Ho-Lim Fung, Song Chen, et al. Neuronal subtypes and diversity revealed by single-nucleus RNA sequencing of the human brain. *Science*, 352(6293):1586–1590, 2016.

[135] Sidharth V. Puram, Itay Tirosh, Anuraag S. Parikh, Anoop P. Patel, Keren Yizhak, Shawn Gillespie, Christopher Rodman, Christina L. Luo, Edmund A. Mroz, Kevin S. Emerick, Daniel G. Deschler, Mark A. Varvares, Ravi Mylvaganam, Orit Rozenblatt-Rosen, James W. Rocco, William C. Faquin, Derrick T. Lin, Aviv Regev, and Bradley E. Bernstein. Single-cell transcriptomic analysis of primary and metastatic tumor ecosystems in head and neck cancer. *Cell*, 171 (7):1611–1624, 2017.

[136] Daniel T. Montoro, Adam L. Haber, Moshe Biton, Vladimir Vinarsky, Brian

Lin, Susan E. Birket, Feng Yuan, Sijia Chen, Hui Min Leung, Jorge Villoria, Noga Rogel, Grace Burgin, Alexander M. Tsankov, Avinash Waghray, Michal Slyper, Julia Waldman, Lan Nguyen, Danielle Dionne, Orit Rozenblatt-Rosen, Purushothama Rao Tata, Hongmei Mou, Manjunatha Shivaraju, Hermann Bihler, Martin Mense, Guillermo J. Tearney, Steven M. Rowe, John F. Engelhardt, Aviv Regev, and Jayaraj Rajagopal. A revised airway epithelial hierarchy includes CFTR-expressing ionocytes. *Nature*, 560(7718):319, 2018. ISSN 1476-4687. doi: 10.1038/s41586-018-0393-7.

[137] Renchao Chen, Xiaoji Wu, Lan Jiang, and Yi Zhang. Single-cell RNA-seq reveals hypothalamic cell diversity. *Cell Reports*, 18(13):3227–3241, 2017.

[138] Sydney M Sanderson, Zhengtao Xiao, Amy J Wisdom, Shree Bose, Maria V Liberti, Michael A Reid, Emily Hocke, Simon G Gregory, David G Kirsch, and Jason W Locasale. The Na+/K+ atpase regulates glycolysis and defines immunometabolism in tumors. *bioRxiv*, 2020. doi: 10.1101/2020.03.31.018739.

[139] John N Campbell, Evan Z Macosko, Henning Fenselau, Tune H Pers, Anna Lyubetskaya, Danielle Tenen, Melissa Goldman, Anne MJ Verstegen, Jon M Resch, Steven A McCarroll, et al. A molecular census of arcuate hypothalamus and median eminence cell types. *Nature Neuroscience*, 20(3):484–496, 2017.

[140] Rapolas Zilionis, Camilla Engblom, Christina Pfirschke, Virginia Savova, David Zemmour, Hatice D Saatcioglu, Indira Krishnan, Giorgia Maroni, Claire V Meyerovitz, Clara M Kerwin, Sun Choi, William G Richards, Assunta De Rienzo, Daniel G Tenen, Raphael Bueno, Elena Levantini, and Allon M Pittet, Mikael J amd Klein. Single-cell transcriptomics of human and mouse lung cancers reveals conserved myeloid populations across individuals and species. *Immunity*, 50(5):1317–1334, 2019.

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

[142] Sinisa Hrvatin, Daniel R. Hochbaum, M. Aurel Nagy, Marcelo Cicconet, Keiramarie Robertson, Lucas Cheadle, Rapolas Zilionis, Alex Ratner, Rebeca Borges-Monroy, Allon M. Klein, Bernardo L. Sabatini, and Michael E. Greenberg. Single-cell analysis of experience-dependent transcriptomic states in the mouse visual cortex. *Nature Neuroscience*, 21(1):120–129, 2018.

[143] Nicholas Schaum, Jim Karkanias, Norma F. Neff, Andrew P. May, Stephen R. Quake, Tony Wyss-Coray, Spyros Darmanis, Joshua Batson, Olga Botvinnik, Michelle B. Chen, Steven Chen, Foad Green, Robert C. Jones, Ashley Maynard, Lolita Penland, Angela Oliveira Pisco, Rene V. Sit, Geoffrey M. Stanley, James T. Webber, Fabio Zanini, Ankit S. Baghel, Isaac Bakerman, Ishita Bansal, Daniela Berdnik, Biter Bilen, Douglas Brownfield, Corey Cain, Michelle B. Chen, Steven Chen, Min Cho, Giana Cirolia, Stephanie D. Conley, Spyros Darmanis, Aaron Demers, Kubilay Demir, Antoine de Morree, Tessa Divita, Haley du Bois, Laughing Bear Torrez Dulgeroff, Hamid Ebadi, F. Hernán Espinoza, Matt Fish, Qiang Gan, Benson M. George, Astrid Gillich, Foad Green, Geraldine Genetiano, Xueying Gu, Gunsagar S. Gulati, Yan Hang, Shayan Hosseinzadeh, Albin Huang, Tal Iram, Taichi Isobe, Feather Ives, Robert C. Jones, Kevin S. Kao, Guruswamy Karnam, Aaron M. Kershner, Bernhard M. Kiss, William Kong, Maya E. Kumar, Jonathan Y. Lam, Davis P. Lee, Song E. Lee, Guang Li, Qingyun Li, Ling Liu, Annie Lo, Wan-Jin Lu, Anoop Manjunath, Andrew P. May, Kaia L. May, Oliver L. May, Ashley Maynard, Marina McKay, Ross J. Metzger, Marco Mignardi, Dullei Min,

Ahmad N. Nabhan, Norma F. Neff, Katharine M. Ng, Joseph Noh, Rasika Patkar, Weng Chuan Peng, Lolita Penland, Robert Puccinelli, Eric J. Rulifson, Nicholas Schaum, Shaheen S. Sikandar, Rahul Sinha, Rene V. Sit, Krzysztof Szade, Weilun Tan, Cristina Tato, Krissie Tellez, Kyle J. Travaglini, Carolina Tropini, Lucas Waldburger, Linda J. van Weele, Michael N. Wosczyna, Jinyi Xiang, Soso Xue, Justin Youngyunpipatkul, Fabio Zanini, Macy E. Zardeneta, Fan Zhang, Lu Zhou, Ishita Bansal, Steven Chen, Min Cho, Giana Cirolia, Spyros Darmanis, Aaron Demers, Tessa Divita, Hamid Ebadi, Geraldine Genetiano, Foad Green, Shayan Hosseinzadeh, Feather Ives, Annie Lo, Andrew P. May, Ashley Maynard, Marina McKay, Norma F. Neff, Lolita Penland, Rene V. Sit, Weilun Tan, Lucas Waldburger, Justin Youngyunpipatkul, Joshua Batson, Olga Botvinnik, Paola Castro, Derek Croote, Spyros Darmanis, Joseph L. DeRisi, Jim Karkanias, Angela Oliveira Pisco, Geoffrey M. Stanley, James T. Webber, Fabio Zanini, Ankit S. Baghel, Isaac Bakerman, Joshua Batson, Biter Bilen, Olga Botvinnik, Douglas Brownfield, Michelle B. Chen, Spyros Darmanis, Kubilay Demir, Antoine de Morree, Hamid Ebadi, F. Hernán Espinoza, Matt Fish, Qiang Gan, Benson M. George, Astrid Gillich, Xueying Gu, Gunsagar S. Gulati, Yan Hang, Albin Huang, Tal Iram, Taichi Isobe, Guruswamy Karnam, Aaron M. Kershner, Bernhard M. Kiss, William Kong, Christin S. Kuo, Jonathan Y. Lam, Benoit Lehallier, Guang Li, Qingyun Li, Ling Liu, Wan-Jin Lu, Dullei Min, Ahmad N. Nabhan, Katharine M. Ng, Patricia K. Nguyen, Rasika Patkar, Weng Chuan Peng, Lolita Penland, Eric J. Rulifson, Nicholas Schaum, Shaheen S. Sikandar, Rahul Sinha, Krzysztof Szade, Serena Y. Tan, Krissie Tellez, Kyle J. Travaglini, Carolina Tropini, Linda J. van Weele, Bruce M. Wang, Michael N. Wosczyna, Jinyi Xiang, Hanadie Yousef, Lu Zhou, Joshua Batson, Olga Botvinnik, Steven Chen, Spyros Darmanis, Foad

Green, Andrew P. May, Ashley Maynard, Angela Oliveira Pisco, Stephen R. Quake, Nicholas Schaum, Geoffrey M. Stanley, James T. Webber, Tony Wyss-Coray, Fabio Zanini, Philip A. Beachy, Charles K. F. Chan, Antoine de Morree, Benson M. George, Gunsagar S. Gulati, Yan Hang, Kerwyn Casey Huang, Tal Iram, Taichi Isobe, Aaron M. Kershner, Bernhard M. Kiss, William Kong, Guang Li, Qingyun Li, Ling Liu, Wan-Jin Lu, Ahmad N. Nabhan, Katharine M. Ng, Patricia K. Nguyen, Weng Chuan Peng, Eric J. Rulifson, Nicholas Schaum, Shaheen S. Sikandar, Rahul Sinha, Krzysztof Szade, Kyle J. Travaglini, Carolina Tropini, Bruce M. Wang, Kenneth Weinberg, Michael N. Wosczyna, Sean M. Wu, Hanadie Yousef, Ben A. Barres, Philip A. Beachy, Charles K. F. Chan, Michael F. Clarke, Spyros Darmanis, Kerwyn Casey Huang, Jim Karkanias, Seung K. Kim, Mark A. Krasnow, Maya E. Kumar, Christin S. Kuo, Andrew P. May, Ross J. Metzger, Norma F. Neff, Roel Nusse, Patricia K. Nguyen, Thomas A. Rando, Justin Sonnenburg, Bruce M. Wang, Kenneth Weinberg, Irving L. Weissman, Sean M. Wu, Stephen R. Quake, Tony Wyss-Coray, The Tabula Muris Consortium, Overall coordination, Logistical coordination, Organ collection and processing, Library preparation and sequencing, Computational data analysis, Cell type annotation, Writing group, Supplemental text writing group, and Principal investigators. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature*, 562(7727):367–372, 2018.

[144] Tanya T Karagiannis, John P Cleary, Busra Gok, Andrew J Henderson, Nicholas G Martin, Masanao Yajima, Elliot C Nelson, and Christine S Cheng. Single cell transcriptomics reveals opioid usage evokes widespread suppression of antiviral gene program. *Nature Communications*, 11(1):1–10, 2020.

[145] Luz D Orozco, Hsu-Hsin Chen, Christian Cox, Kenneth J Katschke Jr, Rommel Arceo, Carmina Espiritu, Patrick Caplazi, Sarajane Saturnio Nghiem, Ying-

Jiun Chen, Zora Modrusan, Amy Dressen, Leonard D Goldstein, Christine Clarke, Tushar Bhangale, Brian Yaspan, Marion Jeanne, Michael J Townsend, Menno van Lookeren Campagne, and Jason A Hackney. Integration of eQTL and a single-cell atlas in the human eye identifies causal genes for age-related macular degeneration. *Cell Reports*, 30(4):1246–1259, 2020.

[146] Patricia A. Darrah, Joseph J. Zeppa, Pauline Maiello, Joshua A. Hackney, Marc H. Wadsworth, Travis K. Hughes, Supriya Pokkali, Phillip A. Swanson, Nicole L. Grant, Mark A. Rodgers, Megha Kamath, Chelsea M. Causgrove, Dominick J. Laddy, Aurelio Bonavia, Danilo Casimiro, Philana Ling Lin, Edwin Klein, Alexander G. White, Charles A. Scanga, Alex K. Shalek, Mario Roederer, JoAnne L. Flynn, and Robert A. Seder. Prevention of tuberculosis in macaques after intravenous BCG immunization. *Nature*, 577(7788):95–102, 2020.

[147] Velina Kozareva, Caroline Martin, Tomas Osorno, Stephanie Rudolph, Chong Guo, Charles Vanderburg, Naeem Nadaf, Aviv Regev, Wade G Regehr, and Evan Macosko. A transcriptomic atlas of mouse cerebellar cortex comprehensively defines cell types. *Nature*, 598(7879):214–219, 2021.

[148] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[149] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

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

[151] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B.*, 39:1–39, 1977.

[152] Elham Azizi, Sandhya Prabhakaran, Ambrose Carr, and Dana Pe'er. Bayesian inference for single-cell clustering and imputing. *Genomics and Computational Biology*, 3(1):e46–e46, 2017.

[153] Dilan Görür and Carl Edward Rasmussen. Dirichlet process gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology*, 25(4):653–664, 2010.

[154] Yue Deng, Feng Bao, Qionghai Dai, Lani F Wu, and Steven J Altschuler. Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning. *Nature Methods*, 16(4):311–314, 2019.

[155] Gökcen Eraslan, Lukas M Simon, Maria Mircea, Nikola S Mueller, and Fabian J Theis. Single-cell rna-seq denoising using a deep count autoencoder. *Nature Communications*, 10:390, 2019.

[156] Cédric Arisdakessian, Olivier Poirion, Breck Yunits, Xun Zhu, and Lana X Garmire. DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. *Genome Biology*, 20(1):1–14, 2019.

[157] Zdravko I Botev, Joseph F Grotowski, Dirk P Kroese, et al. Kernel density estimation via diffusion. *The Annals of Statistics*, 38(5):2916–2957, 2010.

[158] Bang Tran, Duc Tran, Hung Nguyen, Nam Sy Vo, and Tin Nguyen. Ria: a novel regression-based imputation approach for single-cell rna sequencing. In *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*, pages 1–9. IEEE, 2019.

[159] Juexin Wang, Anjun Ma, Yuzhou Chang, Jianting Gong, Yuexu Jiang, Ren Qi, Cankun Wang, Hongjun Fu, Qin Ma, and Dong Xu. scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses. *Nature Communications*, 12(1):1882, 2021.

[160] Xiaoying Fan, Xiannian Zhang, Xinglong Wu, Hongshan Guo, Yuqiong Hu, Fuchou Tang, and Yanyi Huang. Single-cell RNA-seq transcriptome analysis of linear and circular RNAs in mouse preimplantation embryos. *Genome Biology*, 16(1):148, 2015.

[161] Barbara Treutlein, Doug G Brownfield, Angela R Wu, Norma F Neff, Gary L Mantalas, F Hernan Espinoza, Tushar J Desai, Mark A Krasnow, and Stephen R Quake. Reconstructing lineage hierarchies of the distal lung epithelium using single-cell RNA-seq. *Nature*, 509:371–375, 2014.

[162] Gioele La Manno, Daniel Gyllborg, Simone Codeluppi, Kaneyasu Nishimura, Carmen Salto, Amit Zeisel, Lars E Borm, Simon RW Stott, Enrique M Toledo, J Carlos Villaescusa, Peter Lönnerberg, Jesper Ryge, Roger A Barker, Ernest Arenas, and Sten Linnarsson. Molecular diversity of midbrain development in mouse, human, and stem cells. *Cell*, 167(2):566–580, 2016.

[163] Sueli Marques, Amit Zeisel, Simone Codeluppi, David van Bruggen, Ana Mendanha Falcão, Lin Xiao, Huiliang Li, Martin Häring, Hannah Hochgerner, Roman A Romanov, Hannah Hochgerner, Roman A Romanov, Daniel Gyllborg, Ana B Muñoz-Manchado, Jens Hjerling-Leffler, Tibor Harkany, William D Richardson, Sten Linnarsson, and Gonçalo Castelo-Branco. Oligodendrocyte heterogeneity in the mouse juvenile and adult central nervous system. *Science*, 352(6291):1326–1329, 2016.

[164] Bosiljka Tasic, Zizhen Yao, Lucas T. Graybuck, Kimberly A. Smith, Thuc Nghi

Nguyen, Darren Bertagnolli, Jeff Goldy, Emma Garren, Michael N. Economo, Sarada Viswanathan, Osnat Penn, Trygve Bakken, Vilas Menon, Jeremy Miller, Olivia Fong, Karla E. Hirokawa, Kanan Lathia, Christine Rimorin, Michael Tieu, Rachael Larsen, Tamara Casper, Eliza Barkan, Matthew Kroll, Sheana Parry, Nadiya V. Shapovalova, Daniel Hirschstein, Julie Pendergraft, Heather A. Sullivan, Tae Kyung Kim, Aaron Szafer, Nick Dee, Peter Groblewski, Ian Wickersham, Ali Cetin, Julie A. Harris, Susan M. Levi, Boaz P.and Sunkin, Linda Madisen, Tanya L. Daigle, Loren Looger, Amy Bernard, John Phillips, Ed Lein, Michael Hawrylycz, Karel Svoboda, Allan R. Jones, Christof Koch, and Hongkui Zeng. Shared and distinct transcriptomic cell types across neocortical areas. *Nature*, 563(7729):72–78, 2018.

[165] Tavé van Zyl, Wenjun Yan, Alexi McAdams, Yi-Rong Peng, Karthik Shekhar, Aviv Regev, Dejan Juric, and Joshua R. Sanes. Cell atlas of aqueous humor outflow pathways in eyes of humans and four model species provides insight into glaucoma pathogenesis. *Proceedings of the National Academy of Sciences*, 117(19):10339–10349, 2020.

[166] Kevin Wei, Ilya Korsunsky, Jennifer L. Marshall, Anqi Gao, Gerald FM. Watts, Triin Major, Adam P. Croft, Jordan Watts, Philip E. Blazar, Jeffrey K. Lange, Thomas S. Thornhill, Andrew Filer, Karim Raza, Laura T. Donlin, Accelerating Medicines Partnership Rheumatoid Arthritis, Systemic Lupus Erythematosus (AMP RA/SLE) Consortium, Christian W. Siebel, Christopher D. Buckley, Soumya Raychaudhuri, and Michael B. Brenner. Notch signalling drives synovial fibroblast identity and arthritis pathology. *Nature*, 582:259–264, 2020.

[167] Chen Cao, Laurence A. Lemaire, Wei Wang, Peter H. Yoon, Yoolim A. Choi, Lance R. Parsons, John C. Matese, Michael Levine, and Kai Chen. Comprehen-

sive single-cell transcriptome lineages of a proto-vertebrate. *Nature*, 571(7765): 349–354, 2019.

[168] Paul Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579, 1901.

[169] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

[170] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[171] Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007.

[172] Florian Buettner, Kedar N Natarajan, F Paolo Casale, Valentina Proserpio, Antonio Scialdone, Fabian J Theis, Sarah A Teichmann, John C Marioni, and Oliver Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature Biotechnology*, 33(2):155–160, 2015.

[173] Ashraful Haque, Jessica Engel, Sarah A Teichmann, and Tapio Lönnberg. A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. *Genome Medicine*, 9(1):75, 2017.

[174] Wenhao Tang, François Bertaux, Philipp Thomas, Claire Stefanelli, Malika Saint, Samuel Marguerat, and Vahid Shahrezaei. bayNorm: Bayesian gene expression recovery, imputation and normalization for single-cell RNA-sequencing data. *Bioinformatics*, 36(4):1174–1181, 2020.

[175] Florian Wagner, Yun Yan, and Itai Yanai. K-nearest neighbor smoothing for high-throughput single-cell rna-seq data. *BioRxiv*, page 217737, 2017.

[176] Duc Tran, Frederick C Harris, Bang Tran, Nam Sy Vo, Hung Nguyen, and Tin Nguyen. Single-cell RNA sequencing data imputation using deep neural network. In *ITNG 2021 18th International Conference on Information Technology-New Generations*, pages 403–410. Springer, 2021.

[177] Zhun Miao, Jiaqi Li, and Xuegong Zhang. scRecover: Discriminating true and false zeros in single-cell RNA-seq data for imputation. *bioRxiv*, page 665323, 2019.

[178] Duc Tran, Hung Nguyen, Bang Tran, Carlo La Vecchia, Hung N. Luu, and Tin Nguyen. Fast and precise single-cell data analysis using hierarchical autoencoder. *Nature Communications*, 12:1029, 2021.

[179] Tanya Barrett, Stephen E. Wilhite, Pierre Ledoux, Carlos Evangelista, Irene F. Kim, Maxim Tomashevsky, Kimberly A. Marshall, Katherine H. Phillippy, Patti M. Sherman, Michelle Holko, Andrey Yefanov, Hyeseung Lee, Naigong Zhang, Cynthia L. Robertson, Nadezhda Serova, Sean Davis, and Alexandra Soboleva. NCBI GEO: archive for functional genomics data sets–update. *Nucleic Acids Research*, 41(D1):D991–D995, 2013.

[180] Gabriella Rustici, Nikolay Kolesnikov, Marco Brandizi, Tony Burdett, Miroslaw Dylag, Ibrahim Emam, Anna Farne, Emma Hastings, Jon Ison, Maria Keays, Natalja Kurbatova, James Malone, Roby Mani, Annalisa Mupo, Rui Pedro Pereira, Ekaterina Pilicheva, Johan Rung, Anjan Sharma, Y. Amy Tang, Tobias Ternent, Andrew Tikhonov, Danielle Welter, Eleanor Williams, Alvis Brazma, Helen Parkinson, and Ugis Sarkans. ArrayExpress update–trends in database

growth and links to data analysis tools. *Nucleic Acids Research*, 41(D1):D987–D990, 2013.

[181] Zeynab Maghsoudi, Ha Nguyen, Alireza Tavakkoli, and Tin Nguyen. A comprehensive survey of the approaches for pathway analysis using multi-omics data integration. *Briefings in Bioinformatics*, 23(6):bbac435, 2022.

[182] Hung Nguyen, Duc Tran, Jonathan M. Galazka, Sylvain V. Costes, Afshin Beheshti, Sorin Draghici, and Tin Nguyen. CPA: A web-based platform for consensus pathway analysis and interactive visualization. *Nucleic Acids Research*, 49(W1):W114–W124, 2021.

[183] Jovan Tanevski, Thin Nguyen, Buu Truong, Nikos Karaiskos, Mehmet Eren Ahsen, Xinyu Zhang, Chang Shu, Ke Xu, Xiaoyu Liang, Ying Hu, Hoang VV Pham, Li Xiaomei, Thuc D Le, Adi L Tarca, Gaurav Bhatti, Roberto Romero, Nestoras Karathanasis, Phillipe Loher, Yang Chen, Zhengqing Ouyang, Disheng Mao, Yuping Zhang, Maryam Zand, Jianhua Ruan, Christoph Hafemeister, Peng Qiu, Duc Tran, Tin Nguyen, Attila Gabor, Thomas Yu, Justin Guinney, Enrico Glaab, Roland Krause, Peter Banda, DREAM SCTC Consortium, Gustavo Stolovitzky, Nikolaus Rajewsky, Julio Saez-Rodriguez, and Pablo Meyer. Gene selection for optimal prediction of cell position in tissues from single-cell transcriptomics data. *Life Science Alliance*, 3(11), 2020. doi: 10.26508/lsa.202000867.

[184] Tuan-Minh Nguyen, Adib Shafi, Tin Nguyen, and Sorin Draghici. Identifying significantly impacted pathways: a comprehensive review and assessment. *Genome Biology*, 20:203, 2019.

[185] Adib Shafi, Tin Nguyen, Azam Peyvandipour, Hung Nguyen, and Sorin

Draghici. A multi-cohort and multi-omics meta-analysis framework to identify network-based gene signatures. *Frontiers in Genetics*, 10:159, 2019.

[186] Adib Shafi, Tin Nguyen, Azam Peyvandipour, and Sorin Draghici. GSMA: an approach to identify robust global and test Gene Signatures using Meta-Analysis. *Bioinformatics*, 36(2):487–495, 2019.

[187] Hung Nguyen, Sangam Shrestha, Duc Tran, Adib Shafi, Sorin Draghici, and Tin Nguyen. A comprehensive survey of tools and software for active subnetwork identification. *Frontiers in Genetics*, 10:155, 2019.

[188] Tin Nguyen, Cristina Mitrea, and Sorin Draghici. Network-based approaches for pathway level analysis. *Current Protocols in Bioinformatics*, 61(1):8–25, 2018.

[189] Edward Cruz, Hung Nguyen, Tin Nguyen, and Ian Wallace. Functional analysis tools for post-translational modification: a post-translational modification database for analysis of proteins and metabolic pathways. *The Plant Journal*, 99(5):1003–1013, 2019.

[190] Diana Diaz, Tin Nguyen, and Sorin Draghici. A systems biology approach for unsupervised clustering of high-dimensional data. In *The Second International Workshop on Machine Learning, Optimization and Big Data*, pages 193–203, 2016.

[191] Diana Diaz, Michele Donato, Tin Nguyen, and Sorin Draghici. MicroRNA-augmented pathways (mirAP) and their applications to pathway analysis and disease subtyping. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, volume 22, pages 390–401, New Jersey, 2017. World Scientific.

[192] Thair Judeh, Tin Chi Nguyen, and Dongxiao Zhu. QSEA for fuzzy subgraph querying of KEGG pathways. In *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, pages 474–481, 2012.

[193] Tin Nguyen, Adib Shafi, Tuan-Minh Nguyen, A. Grant Schissler, and Sorin Draghici. NBIA: a network-based integrative analysis framework–applied to pathway analysis. *Scientific Reports*, 10:4188, 2020.

[194] Brian Marks, Nina Hees, Hung Nguyen, and Tin Nguyen. MIA: A Multi-cohort Integrated Analysis for biomarker identification. In *Proceedings of the 9th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. ACM, 2018.

[195] Tin Nguyen, Cristina Mitrea, Rebecca Tagett, and Sorin Draghici. DANUBE: Data-driven meta-ANalysis using UnBiased Empirical distributions - applied to biological pathway analysis. *Proceedings of the IEEE*, 105(3):496–515, 2017.

[196] Tin Nguyen, Diana Diaz, Rebecca Tagett, and Sorin Draghici. Overcoming the matched-sample bottleneck: an orthogonal approach to integrate omic data. *Scientific Reports*, 6:29251, 2016. doi: 10.1038/srep29251.

[197] Tin Nguyen, Diana Diaz, and Sorin Draghici. TOMAS: A novel TOpology-aware Meta-Analysis approach applied to System biology. In *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 13–22. ACM, 2016.

[198] Tin Nguyen, Rebecca Tagett, Michele Donato, Cristina Mitrea, and Sorin Draghici. A novel bi-level meta-analysis approach-applied to biological pathway analysis. *Bioinformatics*, 32(3):409–416, 2016.

[199] Rebecca J. Austin-Datta, Carlo La Vecchia, Thomas J George, Faheez Mohamed, Paolo Boffetta, Sean P. Dineen, Daniel Q. Huang, Thanh-Huyen T Vu, Tin C. Nguyen, Jennifer B Permuth, and Hung N. Luu. A call for standardized reporting of early-onset colorectal peritoneal metastases. *European Journal of Cancer Prevention: the Official Journal of the European Cancer Prevention Organisation (ECP)*, DOI: 10.1097/CEJ.0000000000000816, 2023.

[200] Quang-Huy Nguyen, Tin Nguyen, and Duc-Hau Le. Identification and validation of a novel three hub long noncoding RNAs with m6A modification signature in low-grade gliomas. *Frontiers in Molecular Biosciences*, 9:801931, 2022.

[201] Quang-Huy Nguyen, Tin Nguyen, and Duc-Hau Le. DrGA: cancer driver gene analysis in a simpler manner. *BMC Bioinformatics*, 23:86, 2022.

[202] Bashir Dabo, Claudio Pelucchi, Matteo Rota, Harshonnati Jain, Paola Bertuccio, Rossella Bonzi, Domenico Palli, Monica Ferraroni, Zuo-Feng Zhang, Aurora Sanchez-Anguiano, Yenh Thi-Hai Pham, Chi Thi-Du Tran, Anh Gia Pham, Guo-Pei Yu, Tin C. Nguyen, Joshua Muscat, Shoichiro Tsugane, Akihisa Hidaka, Gerson S. Hamada, David Zaridze, Dmitry Maximovitch, Manolis Kogevinas, Nerea Fernàndez de Larrea, Stefania Boccia, Robert C. Pastorino, Robertav; Kurtz, Areti Lagiou, Pagona Lagiou, Jesus Vioque, M. Constanza Camargo, Maria Paula Curado, Nuno Lunet, Paolo Boffetta, Eva Negri, Carlo La Vecchia, and Hung N. Luu. The association between diabetes and gastric cancer: results from the stomach cancer pooling project consortium. *European Journal of Cancer Prevention*, 31(3):260, 2022.

[203] Hung N. Luu, Pedram Paragomi, Renwei Wang, Joyce Y. Huang, Jennifer Adams-Haduch, Øivind Midttun, Arve Ulvik, Tin C. Nguyen, Randall E. Brand, Yutang Gao, Per Magne Ueland, and Jian-Min Yuan. The associa-

tion between serum serine and glycine and related-metabolites with pancreatic cancer in a prospective cohort study. *Cancers*, 14(9):2199, 2022.

[204] Hung Nguyen, Duc Tran, Bang Tran, Monikrishna Roy, Adam Cassell, Sergiu Dascalu, Sorin Draghici, and Tin Nguyen. SMRT: Randomized data transformation for cancer subtyping and big data analysis. *Frontiers in Oncology*, 11: 725133, 2021.

[205] Thi Hai Yen Nguyen, Tin Nguyen, Quang-Huy Nguyen, and Duc-Hau Le. Re-identification of patient subgroups in uveal melanoma. *Frontiers in Oncology*, 11:731548, 2021.

[206] Duc Tran, Hung Nguyen, Uyen Le, George Bebis, Hung N. Luu, and Tin Nguyen. A novel method for cancer subtyping and risk prediction using consensus factor analysis. *Frontiers in Oncology*, 10:1052, 2020.

[207] Quang-Huy Nguyen, Hung Nguyen, Tin Nguyen, and Duc-Hau Le. Multi-omics analysis detects novel prognostic subgroups of breast cancer. *Frontiers in Genetics*, 11:1265, 2020. ISSN 1664-8021. doi: 10.3389/fgene.2020.574661.

[208] Suzan Arslanturk, Sorin Draghici, and Tin Nguyen. Integrated cancer subtyping using heterogeneous genome-scale molecular datasets. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, volume 25, page 551. World Scientific, 2020.

[209] Hung Nguyen, Bang Tran, Duc Tran, Quang-Huy Nguyen, Duc-Hau Le, and Tin Nguyen. Disease subtyping using community detection from consensus networks. In *2020 12th International Conference on Knowledge and Systems Engineering (KSE)*, pages 318–323. IEEE, 2020.

[210] Hung Nguyen, Sushil J Louis, and Tin Nguyen. MGKA: A genetic algorithm-based clustering technique for genomic data. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 103–110. IEEE, 2019.

[211] Yan Yan, Tin Nguyen, Bobby Bryant, and Frederick C Harris Jr. Robust fuzzy cluster ensemble on cancer gene expression data. In *Proceedings of 11th International Conference*, volume 60, pages 120–128, 2019.

[212] Yifan Zhang, Duc Tran, Tin Nguyen, Sergiu M Dascalu, and Frederick C. Harris. A robust and accurate single-cell data trajectory inference method using ensemble pseudotime. *BMC Bioinformatics*, 24(1):1–21, 2023.

[213] Duc Tran, Ha Nguyen, Hung Nguyen, and Tin Nguyen. Dwen: A novel method for accurate estimation of cell type compositions from bulk data samples. In *2022 14th International Conference on Knowledge and Systems Engineering (KSE)*, pages 1–6. IEEE, 2022.

[214] Attila Gabor, Marco Tognetti, Alice Driessen, Jovan Tanevski, Baosen Guo, Wencai Cao, He Shen, Thomas Yu, Verena Chung, Bernd Bodenmiller, Julio Saez-Rodriguez, Augustinas Prusokas, Alidivinas Prusokas, Renata Retkute, Anand Rajasekar, Karthik Raman, Malvika Sudhakar, Raghunathan Rengaswamy, Edward S.C. Shih, Min jeong Kim, Changje Cho, Dohyang Kim, Hyeju Oh, Jinseub Hwang, Kim Jongtae, Yeongeun Nam, Sanghoo Yoon, Taeyong Kwon, Kyeongjun Lee, Sarika Chaudhary, Nehal Sharma, Shreya Bande, Gao Gao fan zhu Cankut Cubuk, Pelin Gundogdu, Joaquin Dopazo, Kinza Rian, Carlos Loucera, Matias M Falco, Martin Garrido-Rodriguez, Maria Peña-Chilet, Huiyuan Chen, Gabor Turu, Laszlo Hunyadi, Adam Misak, Baosen Guo, Wencai Cao, He Shen, Lisheng Zhou, Xiaoqing Jiang, Pieta Zhang, Aakansha Rai, Rintu Kutum, Sadhna Rana, Rajgopal Srinivasan, Swatantra Pradhan,

James Li, Vladimir Bajic, Christophe Van Neste, Didier Barradas-bautista, Somayah Abdullah Albarade, Igor Nikolskiy, Musalula Sinkala, Duc Tran, Hung Nguyen, Tin Nguyen, Alexander Wu, Benjamin DeMeo, Brian Hie, Rohit Singh, Jiwei Liu, Xueer Chen, Leonor Saiz, Jose M. G Vilar, Peng Qiu, Akash Gosain, Anjali Dhall, Dinesh Bajaj, Harpreet Kaur, Krishna Bagaria, Mayank Chauhan, Neelam Sharma, Gajendra Raghava, Sumeet Patiyal, Jianye Hao, Jiajie Peng, Shangyi Ning, Yi Ma, Zhongyu Wei, Atte Aalto, Jorge Goncalves, Laurent Mombaerts, Xinnan Dai, Jie Zheng, Piyushkumar Mundra, Fan Xu, Jie Wang, Krishna Kant Singh, and Mingyu Lee. Cell-to-cell and type-to-type heterogeneity of signaling networks: insights from the crowd. *Molecular Systems Biology*, 17:e10402, 2021. doi: 10.15252/msb.202110402. URL `https://doi.org/10.15252/msb.202110402`.

[215] Duc Tran, Bang Tran, Hung Nguyen, and Tin Nguyen. A novel method for single-cell data imputation using subspace regression. *Scientific Reports*, 12: 2697, 2022.

[216] Bang Tran, Duc Tran, Hung Nguyen, Seungil Ro, and Tin Nguyen. scCAN: single-cell clustering using autoencoder and network fusion. *Scientific Reports*, 12:10267, 2022.

[217] Bang Tran, Quyen Nguyen, Sangam Shrestha, and Tin Nguyen. scids: Single-cell imputation by combining deep autoencoder neural networks and subspace regression. In *2021 13th International Conference on Knowledge and Systems Engineering (KSE)*, pages 1–8, 2021. doi: 10.1109/KSE53942.2021.9648664.

[218] Duc Tran, Hung Nguyen, Frederick C. Harris, and Tin Nguyen. Single-cell rna sequencing data imputation using similarity preserving network. In *2021 13th*

International Conference on Knowledge and Systems Engineering (KSE), pages 1–6. IEEE, 2021.

[219] Amira Alotaibi, Tarik Alafif, Faris Alkhilaiwi, Yasser Alatawi, Hassan Althobaiti, Abdulmajeed Alrefaei, Yousef Hawsawi, and Tin Nguyen. Vit-deit: An ensemble model for breast cancer histopathological images classification. In *2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC)*, pages 1–6. IEEE, 2023.

[220] Benjamin T. Caswell, Caio C. de Carvalho, Hung Nguyen, Monikrishna Roy, Tin Nguyen, and David C. Cantu. Thioesterase enzyme families: Functions, structures, and mechanisms. *Protein Science*, 31(3):652–676, 2022.

[221] Evagelia C. Laiakis, Maisa Pinheiro, Tin Nguyen, Hung Nguyen, Afshin Beheshti, Sucharita M. Dutta, William K. Russell, Mark R. Emmett, and Richard Britten. Quantitative proteomic analytic approaches to identify metabolic changes in the medial prefrontal cortex of rats exposed to space radiation. *Frontiers in Physiology*, DOI: 10.3389/fphys.2022.971282, 2022.

[222] Amruta Kale, Tin Nguyen, Frederick C. Harris Jr, Chenhao Li, Jiyin Zhang, and Xiaogang Ma. Provenance documentation to enable explainable and trustworthy AI: A literature review. *Data Intelligence*, pages 1–41, 2022.

[223] Egle Cekanaviciute, Duc Tran, Hung Nguyen, Alejandra Lopez Macha, Eloise Pariset, Sasha Langley, Giulia Babbi, Sherina Malkani, Sébastien Penninckx, Jonathan C. Schisler, Tin Nguyen, Gary H. Karpen, and Sylvain V. Costes. Mouse genomic associations with in vitro sensitivity to simulated space radiation. *Life Sciences in Space Research*, DOI: 10.1016/j.lssr.2022.07.006, 2022.

[224] Quang Tran, Nam Sy Vo, Eric Hicks, Tin Nguyen, and Vinhthuy Phan. Anal-

ysis of short-read aligners using genome sequence complexity. In *2020 12th International Conference on Knowledge and Systems Engineering (KSE)*, pages 312–317. IEEE, 2020.

[225] Michael P. Menden, Dennis Wang, Mike J. Mason, Bence Szalai, Krishna C. Bulusu, Yuanfang Guan, Thomas Yu, Jaewoo Kang, Minji Jeon, Russ Wolfinger, Tin Nguyen, Mikhail Zaslavskiy, AstraZeneca-Sanger Drug Combination DREAM Consortium, In Sock Jang, Zara Ghazoui, Mehmet E. Ahsen, Robert Vogel, Elias C. Neto, Thea Norman, Eric K. Y. Tang, Mathew J. Garnett, Giovanni Y. Di Veroli, Christian Zwaan, Stephen Fawell, Gustavo Stolovitzky, Justin Guinney, Jonathan R. Dry, and Julio Saez-Rodriguez. Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen. *Nature Communications*, 10:2674, 2019.

[226] John C Stansfield, Duc Tran, Tin Nguyen, and Mikhail G Dozmorov. R tutorial: Detection of differentially interacting chromatin regions from multiple Hi-C datasets. *Current Protocols in Bioinformatics*, 66(1):e76–e76, 2019.

[227] Alfred G. Schissler, Hung Nguyen, Tin Nguyen, Juli Petereit, and Vincent Gardeux. *Statistical Software*, volume 10.1002/9781118445112.stat00527.pub2, pages 1–11. American Cancer Society, 2019.

[228] Adib Shafi, Cristina Mitrea, Tin Nguyen, and Sorin Draghici. A survey of the approaches for identifying differential methylation using bisulfite sequencing data. *Briefings in Bioinformatics*, 19(5):737–753, 2018.

[229] Tin Chi Nguyen and Dongxiao Zhu. Markovbin: An algorithm to cluster metagenomic reads using a mixture modeling of hierarchical distributions. In *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, pages 115–123, 2013.

[230] Zhiyu Zhao, Tin Chi Nguyen, Nan Deng, Kristen Marie Johnson, and Dongxiao Zhu. SPATA: a seeding and patching algorithm for de novo transcriptome assembly. In *2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*, pages 26–33. IEEE, 2011.

[231] Tin Chi Nguyen, Zhiyu Zhao, and Dongxiao Zhu. SPATA: A highly accurate GUI tool for de novo transcriptome assembly. In *2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*, pages 1051–1053. IEEE, 2011.

[232] Tin Chi Nguyen, Nan Deng, Guorong Xu, Zhansheng Duan, and Dongxiao Zhu. iQuant: A fast yet accurate GUI tool for transcript quantification. In *2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*, pages 1048–1050. IEEE, 2011.

[233] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Appendix A

Evaluation metrics

Adjusted Rand Index (ARI) [148] is the corrected-for-chance version of the Rand Index, which measures the agreement between a given clustering and the ground truth. RI is calculated as:

$$RI = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{N}{2}} \tag{A.1}$$

where a is the number of pairs that belong to the same true group and are clustered together, b is the number of pairs that belong to different true groups and are not clustered together, c is the number of pairs that belong to the same groups and are not clustered together, d is the number of pairs that belong to different groups and are clustered together, and $\binom{N}{2}$ is the number of possible pairs that can be formed from the N patients. The ARI takes values from -1 to 1, with the ARI expected to be 0 for a random clustering.

Jaccard index (JI) is also known as Intersection over Union. In our context, The Jaccard index is basically the number of pairs that belong to the same true group and are clustered together (a), divided by the number of pairs that are either in the

same true group (b) or are clustered together (c). JI is calculated as:

$$J = \frac{a}{a+b+c} \tag{A.2}$$

Normalized Mutual Information (NMI) is a normalized version of Mutual Information (MI). Denoting X as the true labeling of the samples and Y is the partitioning obtained from a clustering method, the NMI is calculated as:

$$NMI = \frac{1}{2} \times \frac{I(X;Y)}{H(X)+H(Y)} \tag{A.3}$$

where $I(X;Y)$ is the mutual information between X and Y. $H(X)$ is the entropy of the true partition X and $H(Y)$ is the entropy of the partition obtained from clustering. The NMI value take a range from 0 to 1 in which 1 indicates a perfect match between true labels and clusters. In contrast, 0 value means no mutual information between true labels and clusters.

Appendix B

Publication list

B.1 Journal articles

[1] **Duc Tran**, Bang Tran, Hung Nguyen, and Tin Nguyen. A novel method for single-cell data imputation using subspace regression. *Scientific Reports*, 2022. DOI: 10.1038/s41598-022-06500-4

[2] **Duc Tran**, Hung Nguyen, Bang Tran, Carlo La Vecchia, Hung N. Luu, and Tin Nguyen. Fast and precise single-cell data analysis using hierarchical autoencoder. *Nature Communications*. 2021. DOI: 10.1038/s41467-021-21312-2

[3] **Duc Tran**, Hung Nguyen, Uyen Le, Hung N. Luu, and Tin Nguyen. A novel method for cancer subtyping and risk prediction using consensus factor analysis. *Frontiers in Oncology*, 2020. DOI: 10.3389/fonc.2020.01052

[4] Yifan Zhang, **Duc Tran**, Tin Nguyen, Sergiu M Dascalu, and Frederick C Harris. A robust and accurate single-cell data trajectory inference method using ensemble pseudotime. *BMC Bioinformatics*, 2023. DOI: 10.1186/s12859-023-05179-2

[5] Egle Cekanaviciute, **Duc Tran**, Hung Nguyen, Alejandra Lopez Macha, Eloise Pariset, Sasha Langley, Giulia Babbi, Sherina Malkani, Sébastien Penninckx, Jonathan C Schisler, Tin Nguyen, Gary H Karpen, and Sylvain V Costes. Mouse Genomic Associations with In Vitro Sensitivity to Simulated Space Radiation. *Life Sciences in Space Research*, 2023. DOI: 10.1016/j.lssr.2022.07.006

[6] Bang Tran, **Duc Tran**, Hung Nguyen, Seungil Ro, and Tin Nguyen. scCAN: single-cell clustering using autoencoder and network fusion. *Scientific Reports*, 2022. DOI: 10.1038/s41598-022-06500-4

[7] Hung Nguyen, **Duc Tran**, Bang Tran, Monikrishna Roy, Adam Cassell, Sergiu M Dascalu, Sorin Draghici, Tin Nguyen. SMRT: Randomized data transformation for cancer subtyping and big data analysis. *Frontiers in Oncology*, 2021. DOI: 10.3389/fonc.2021.725133

[8] Hung Nguyen, **Duc Tran**, Afshin Beheshti, Jonathan M. Galazka, Sylvain V. Costes, Sorin Draghici, and Tin Nguyen. CPA: Consensus Pathway Analysis and Interactive Visualization. *Nucleic Acids Research*, 2021. DOI: 10.1093/nar/gkab421

[9] Hung Nguyen, **Duc Tran**, Bang Tran, Bahadir Pehlivan, and Tin Nguyen. A comprehensive survey of regulatory network inference methods using single-cell RNA sequencing data. *Briefings in Bioinformatics*, 2020. DOI: 10.1093/bib/bbaa190

[10] John C. Stansfield, **Duc Tran**, Tin Nguyen, and Mikhail G. Dozmorov. R Tutorial: Detection of Differentially Interacting Chromatin Regions From Multiple Hi-C Datasets. *Current protocols in Bioinformatics*, 2019. DOI: 10.1002/cpbi.76

[11] Dongmei Sun, Thanh M. Nguyen, Robert J. Allaway, Jelai Wang, Verena Chung, Thomas V. Yu, Michael Mason, Isaac Dimitrovsky, Lars Ericson, Hongyang Li, Yuanfang Guan, Ariel Israel, Alex Olar, Balint Armin Pataki, Gustavo

Stolovitzky, Justin Guinney, Percio S. Gulko, Mason B. Frazier, Jake Y. Chen, James C. Costello, S. Louis Bridges Jr, and [RA2-DREAM Challenge Community, including **Duc Tran**]. A Crowdsourcing Approach to Develop Machine Learning Models to Quantify Radiographic Joint Damage in Rheumatoid Arthritis. *JAMA Network Open*, 2022. DOI: 10.1001/jamanetworkopen.2022.27423

[12] Attila Gabor, Marco Tognetti, Alice Driessen, Jovan Tanevski, Baosen Guo, Wencai Cao, He Shen, Thomas Yu, Verena Chung, [Single Cell Signaling in Breast Cancer DREAM Consortium, including **Duc Tran**], Bernd Bodenmiller, and Julio Saez-Rodriguez. Cell-to-cell and type-to-type heterogeneity of signaling networks: insights from the crowd. *Molecular Systems Biology*, 2021. DOI: 10.15252/msb.202110402

[13] Jovan Tanevski, Thin Nguyen, Buu Truong, Nikos Karaiskos, Mehmet Eren Ahsen, Xinyu Zhang, Chang Shu, Ke Xu, Xiaoyu Liang, Ying Hu, Hoang VV Pham, Li Xiaomei, Thuc D Le, Adi L Tarca, Gaurav Bhatti, Roberto Romero, Nestoras Karathanasis, Phillipe Loher, Yang Chen, Zhengqing Ouyang, Disheng Mao, Yuping Zhang, Maryam Zand, Jianhua Ruan, Christoph Hafemeister, Peng Qiu, **Duc Tran**, Tin Nguyen, Attila Gabor, Thomas Yu, Justin Guinney, Enrico Glaab, Roland Krause, Peter Banda, DREAM SCTC Consortium, Gustavo Stolovitzky, Nikolaus Rajewsky, Julio Saez-Rodriguez, and Pablo Meyer. Predicting cells position from single-cell transcriptomics. *Life Science Alliance*, 2020. DOI: 10.26508/lsa.202000867

[14] Hung Nguyen, Sangam Shrestha, **Duc Tran**, and Tin Nguyen. A comprehensive survey for active subnetwork identification. *Frontiers in Genetics*, 2019. DOI: 10.3389/fgene.2019.00155

B.2 Conference proceedings

[1] **Duc Tran**, Ha Nguyen, Hung Nguyen, and Tin Nguyen. DWEN: A novel method for accurate estimation of cell type compositions from bulk data samples. In *Proceedings of the 14th International Conference on Knowledge and Systems Engineering (KSE)*, 2022.

[2] **Duc Tran**, Hung Nguyen, Frederick C. Harris, and Tin Nguyen. Single-cell RNA sequencing data imputation using similarity preserving network. In *Proceedings of the 13th International Conference on Knowledge and Systems Engineering (KSE)*, 2021.

[3] **Duc Tran**, Bang Tran, Hung Nguyen, Frederick C. Harris, Nam Sy Vo, and Tin Nguyen. Single-cell RNA sequencing data imputation using deep neural network. In *Proceedings of the 18th International Conference on Information Technology-New Generations (ITNG)*, 2021.

[4] Bang Tran, **Duc Tran**, Hung Nguyen, Nam Sy Vo, and Tin Nguyen. RIA: a novel regression-based imputation approach for single-cell RNA sequencing. In *Proceedings of the 11th International Conference on Knowledge and Systems Engineering (KSE)*, 2019.

[5] Hung Nguyen, Bang Tran, **Duc Tran**, Quang-Huy Nguyen, Duc-Hau Le, and Tin Nguyen. Disease subtyping using community detection from consensus networks. In *Proceedings of the 12th International Conference on Knowledge and Systems Engineering (KSE)*, 2020.